

QGIS Training Manual

QGIS Project

дек. 04, 2023

Оглавление

1 Введение **1**

1.1 Предисловие . . . 1
1.1.1 Почему QGIS? . . . 1
1.1.2 Предпосылки . . . 2
1.1.3 Лицензия . . . 2
1.1.4 Спонсорские главы . . . 3
1.1.5 Авторы . . . 3
1.1.6 Индивидуальные вкладчики . . . 3
1.1.7 Спонсоры . . . 3
1.1.8 Исходные файлы и отчеты о проблемах . . . 3
1.1.9 Последняя версия . . . 4
1.2 Об упражнениях . . . 4
1.2.1 Как использовать данное руководство . . . 4
1.2.2 Цели многоуровневого курса . . . 4
1.2.3 Данные . . . 5

2 Модуль: Создание и изучение базовой карты **7**

2.1 Занятие: Обзор интерфейса . . . 7
2.1.1 Попробуйте сами: Основы . . . 8
2.1.2 Попробуйте сами 1 . . . 10
2.1.3 Попробуйте сами 2 . . . 10
2.1.4 Что дальше? . . . 11
2.2 Занятие: Добавляем ваши первые слои . . . 11
2.2.1 Идем дальше: Подготовим карту . . . 11
2.2.2 Попробуйте сами . . . 14
2.2.3 Идем дальше: Загружаем векторные данные из Базы данных GeoPackage . . . 15
2.2.4 |Идем дальше|: Загружаем векторные данные из Базы данных SpatiaLite через браузер 16
2.2.5 |Попробуйте сами|, загрузите больше векторных данных . . . 18
2.2.6 Идем дальше: Меняем порядок слоев . . . 18
2.2.7 В заключении . . . 19
2.2.8 Что дальше? . . . 19
2.3 Занятие: Навигация по полотну карты . . . 19
2.3.1 Идем дальше: Базовые инструменты навигации . . . 20
2.3.2 В заключении . . . 23
2.4 Занятие: Символика . . . 23
2.4.1 Идем дальше: Меняем цвета . . . 24
2.4.2 Попробуйте сами . . . 25
2.4.3 Идем дальше: Меняем структуру символа . . . 25
2.4.4 Попробуйте сами . . . 26
2.4.5 Идем дальше: Видимость на основе масштаба . . . 28
2.4.6 Идем дальше: Добавляем слои символа . . . 29

2.4.7 Попробуйте сами . . . 32
2.4.8 Идем дальше: Упорядочиваем уровни символа . . . 33
2.4.9 Попробуйте сами . . . 37
2.4.10 Попробуйте сами . . . 39
2.4.11 Идем дальше: Типы слоя символа . . . 40
2.4.12 Попробуйте сами . . . 48
2.4.13 Идем дальше: Символика генератора геометрии . . . 50
2.4.14 Идем дальше: Создаем пользовательскую заливку формата SVG . . . 54
2.4.15 В заключении . . . 60
2.4.16 Дополнительные материалы для чтения . . . 60
2.4.17 Что дальше? . . . 60

3 Модуль: Классификация векторных данных 61
3.1 Занятие: Атрибуты векторных данных . . . 61
3.1.1 Идем дальше: Обзор атрибутов слоя . . . 61
3.1.2 Попробуйте сами: изучите атрибуты векторных данных . . . 65
3.1.3 В заключении . . . 66
3.1.4 Что дальше? . . . 66
3.2 Занятие: Надписи/метки . . . 66
3.2.1 Идем дальше: Используем надписи/метки . . . 66
3.2.2 Идем дальше: Меняем параметры надписей . . . 68
3.2.3 Идем дальше: Применяем надписи/метки вместо слоя символики . . . 72
3.2.4 Попробуйте сами: Настройте надписи . . . 75
3.2.5 Идем дальше: Надписываем линии . . . 79
3.2.6 Идем дальше: Настраиваем параметры данных . . . 83
3.2.7 Попробуйте сами: Примените настройку параметров данных . . . 85
3.2.8 Дополнительные возможности надписей . . . 87
3.2.9 В заключении . . . 88
3.2.10 Что дальше? . . . 88
3.3 Занятие: Классификация . . . 88
3.3.1 Идем дальше: Классифицируем номинальные данные . . . 88
3.3.2 Попробуйте сами: больше Классификации . . . 93
3.3.3 Идем дальше: Классификация соотношений . . . 93
3.3.4 Попробуйте сами: Точно определить Классификацию . . . 101
3.3.5 Идем дальше: Классификация на основе правил . . . 101
3.3.6 В заключении . . . 105
3.3.7 Что дальше? . . . 105

4 Модуль: Составление базовой карты 107
4.1 Занятие: Создание динамического макета печати . . . 107
4.1.1 Идем дальше: Диспетчер макетов . . . 107
4.1.2 Идем дальше: Составляем базовую карту . . . 109
4.1.3 Идем дальше: Добавляем заголовок . . . 112
4.1.4 Идем дальше: Добавляем легенды . . . 114
4.1.5 Идем дальше: Настраиваем элементы легенд . . . 115
4.1.6 Идем дальше: Экспортируем (выгружаем) карту . . . 117
4.1.7 В заключении . . . 119
4.2 Занятие: Создание динамического макета печати . . . 119
4.2.1 Идем дальше: Создаем динамическое Полотно Карты . . . 119
4.2.2 Идем дальше: Создаем динамический заголовок . . . 120
4.2.3 Идем дальше: Создаем надписи для динамического заголовка . . . 121
4.2.4 Идем дальше: Добавляем изображения к динамическому заголовку . . . 123
4.2.5 Идем дальше: Создаем шкалу динамического заголовка . . . 124
4.2.6 Что дальше? . . . 125
4.3 Задание 1 . . . 125
4.3.1 В заключении . . . 125

5 Module: Модуль: Создание векторных данных 127
5.1 Занятие: Создание нового набора векторных данных . . . 127

5.1.1 Идем дальше: Диалоговое окно для создания слоя . . . 127
5.1.2 Идем дальше: Источник данных . . . 130
5.1.3 Попробуйте сами, оцифруйте полигоны . . . 143
5.1.4 Идем дальше: Использование таблицы редактора Vertex . . . 144
5.1.5 Попробуйте сами: Оцифруйте линии . . . 147
5.1.6 В заключении . . . 150
5.1.7 Что дальше? . . . 150
5.2 Занятие: Топология функций . . . 151
5.2.1 Идем дальше: Привязка . . . 151
5.2.2 Идем дальше: Правильные топологические Функции . . . 153
5.2.3 Идем дальше: Инструмент: Упрощение функции . . . 155
5.2.4 Попробуйте сами Инструмент: Добавить кольцо . . . 157
5.2.5 Попробуйте сами Инструмент: Добавить часть . . . 158
5.2.6 Идем дальше: Инструмент: Изменить форму Функции . . . 159
5.2.7 Попробуйте сами Инструмент: Разделить Функции . . . 162
5.2.8 Попробуйте сами: Инструмент: Слить Функции . . . 163
5.2.9 В заключении . . . 164
5.2.10 Что дальше? . . . 164
5.3 Занятие: Формы . . . 164
5.3.1 Идем дальше: Используем Функциональность дизайна Формы QGIS . . . 165
5.3.2 Попробуйте сами, используйте формы для редактирования значений . . . 166
5.3.3 Идем дальше: Настройка типов полей формы . . . 166
5.3.4 Попробуйте сами . . . 168
5.3.5 Попробуйте сами, создайте тестовые данные . . . 170
5.3.6 Идем дальше: Создание новой формы . . . 171
5.3.7 Идем дальше: Связывание формы со слоем . . . 173
5.3.8 В заключении . . . 174
5.3.9 Дополнительные материалы для чтения . . . 174
5.3.10 Что дальше? . . . 174
5.4 Занятие: Действия . . . 174
5.4.1 Идем дальше: Добавление поля для изображений . . . 175
5.4.2 Идем дальше: Создание действия . . . 177
5.4.3 Идем дальше: Поиск в Интернете . . . 181
5.4.4 Идем дальше: Открытие веб-страницы непосредственно в QGIS . . . 184
5.4.5 В заключении . . . 186
5.4.6 Что дальше? . . . 186

6 Модуль: Векторный анализ **187**
6.1 Занятие: пере-проецирование и преобразование данных . . . 187
6.1.1 Идем дальше: Проецирование . . . 188
6.1.2 Идем дальше: Пере-проецирование «на лету» . . . 188
6.1.3 Идем дальше: Сохраняем набор данных в другой справочной системе координат (СКК) 189
6.1.4 Идем дальше: Создаем собственную проекцию . . . 192
6.1.5 В заключении . . . 194
6.1.6 Дополнительные материалы для чтения . . . 194
6.1.7 Что дальше? . . . 194
6.2 Занятие: Векторный анализ . . . 194
6.2.1 Процесс GIS . . . 195
6.2.2 Проблема . . . 195
6.2.3 Данные . . . 195
6.2.4 Идем дальше: Начать проект и получить данные . . . 196
6.2.5 Попробуйте сами: Сконвертировать ССК слоев . . . 197
6.2.6 Идем дальше: Анализируем проблемы: расстояние от школ и дорог . . . 199
6.2.7 Попробуйте сами: Расстояние от школ . . . 206
6.2.8 Идем дальше: Зоны перекрытия . . . 209
6.2.9 Идем дальше: Извлечение зданий . . . 212
6.2.10 Попробуйте сами: Дальше отфильтровать наши здания . . . 213
6.2.11 Идем дальше: Выбираем здания подходящего размера . . . 216

6.2.12 Попробуйте сами . . . 217
6.2.13 В заключении . . . 217
6.2.14 Что дальше? . . . 218
6.3 Занятие: Сетевой анализ . . . 218
6.3.1 Идем дальше: Инструменты и данные . . . 218
6.3.2 Рассчитать кратчайший путь (от точки к точке) . . . 220
6.3.3 Попробуйте сами: Самый быстрый путь . . . 223
6.3.4 Идем дальше: Дополнительные опции . . . 224
6.3.5 Кратчайший путь с ограничением скорости . . . 226
6.3.6 Зона обслуживания (из слоя) . . . 229
6.3.7 В заключении . . . 231
6.3.8 Что дальше? . . . 231
6.4 Занятие: Пространственная статистика . . . 231
6.4.1 Идем дальше: Создаем тестовый набор данных . . . 231
6.4.2 Идем дальше: Базовая статистика . . . 236
6.4.3 Идем дальше: Вычисляем статистические данные относительно расстояний между точками . . . 238
6.4.4 Идем дальше: Анализ методом «ближайший сосед» (внутри слоя) . . . 240
6.4.5 Идем дальше: Средние координаты . . . 241
6.4.6 Идем дальше: Гистограммы изображений . . . 242
6.4.7 Идем дальше: Пространственная интерполяция . . . 245
6.4.8 Попробуйте сами: Различные методы интерполяции . . . 245
6.4.9 В заключении . . . 246
6.4.10 Что дальше? . . . 246

7 Модуль: Растры **247**
7.1 Занятие: Работа с растровыми данными . . . 247
7.1.1 Идем дальше: Загружаем растровые данные . . . 247
7.1.2 Идем дальше: Создаем виртуальный растр . . . 248
7.1.3 Преобразование растровых данных . . . 250
7.1.4 В заключении . . . 253
7.1.5 Что дальше? . . . 253
7.2 Идем дальше: Измененяем символику растрового слоя . . . 253
7.2.1 Попробуйте сами . . . 253
7.2.2 Идем дальше: Меняем символику растрового слоя . . . 254
7.2.3 Идем дальше: Однополосная серая . . . 255
7.2.4 Идем дальше: Однополосная псевдо-цветная . . . 256
7.2.5 Идем дальше: Меняем прозрачность . . . 258
7.2.6 В заключении . . . 261
7.2.7 Ссылки . . . 261
7.2.8 Что дальше? . . . 261
7.3 Занятие: Анализ рельефа местности . . . 261
7.3.1 Идем дальше: Вычисляем отмывки . . . 261
7.3.2 Идем дальше: Используем отмывки в качестве наложения (оверлей) . . . 263
7.3.3 Идем дальше: Поиск наилучших мест . . . 264
7.3.4 Идем дальше: Вычисляем уклон . . . 264
7.3.5 Попробуйте сами вычислить аспект . . . 265
7.3.6 Идем дальше: Находим северную сторону . . . 267
7.3.7 Попробуйте сами: Больше критериев . . . 269
7.3.8 Идем дальше: Объединяем результаты анализа растра . . . 272
7.3.9 Идем дальше: Упрощаем растр . . . 274
7.3.10 Идем дальше: Пере-классификация растра . . . 276
7.3.11 Идем дальше: Запрос к растру . . . 280
7.3.12 В заключении . . . 283
7.3.13 Что дальше? . . . 283

8 Модуль: Завершение анализа **285**
8.1 Занятие: Преобразование растрового в векторное . . . 285

8.1.1 Идем дальше: Инструмент Raster to Vector . . . 285
8.1.2 Попробуйте сами . . . 286
8.1.3 Идем дальше: Инструмент Vector to Raster . . . 287
8.1.4 В заключении . . . 289
8.1.5 Что дальше? . . . 289
8.2 Занятие: Объединение анализов . . . 289
8.2.1 Попробуйте сами . . . 289
8.2.2 Попробуйте сами: Проверить результаты . . . 290
8.2.3 Попробуйте сами: Уточнить анализ . . . 290
8.2.4 В заключении . . . 295
8.2.5 Что дальше? . . . 295
8.3 Задание . . . 295
8.4 Занятие: Дополнительное упражнение . . . 296
8.4.1 Постановка задачи . . . 296
8.4.2 Описание решения . . . 296
8.4.3 Идем дальше: Настройка карты . . . 296
8.4.4 Загрузка данных на карту . . . 297
8.4.5 Изменение порядка слоев . . . 298
8.4.6 Найти правильные районы . . . 298
8.4.7 Обрезка растров . . . 299
8.4.8 Изменение условных обозначений векторных слоев . . . 300
8.4.9 Изменение условных обозначений растровых слоев . . . 300
8.4.10 Вычистка карты . . . 301
8.4.11 Создание отмывки . . . 301
8.4.12 Уклон . . . 301
8.4.13 Попробуйте сами: Поработать с аспектом . . . 302
8.4.14 Пере-классификация растров . . . 302
8.4.15 Объединение растров . . . 303
8.4.16 Поиск сельских населенных пунктов . . . 303
8.4.17 Создание отрицательного буфера . . . 304
8.4.18 Векторизация растра . . . 304
8.4.19 Исправление геометрии . . . 305
8.4.20 Определение пересечения векторов . . . 305
8.4.21 Вычисление площади для каждого полигона (многоугольника) . . . 306
8.4.22 Выбор участка заданного размера . . . 306
8.4.23 Оцифровка Кейптаунского университета . . . 306
8.4.24 Поиск ближайших мест к Университету Кейптауна . . . 307

9 Модуль: Плагины **309**
9.1 Занятие: Установка и управление плагинами . . . 309
9.1.1 Идем дальше: Управление плагинами . . . 309
9.1.2 Идем дальше: Установка новых плагинов . . . 310
9.1.3 Идем дальше: Настройка дополнительных репозиториев плагинов . . . 312
9.1.4 В заключении . . . 314
9.1.5 Что дальше? . . . 314
9.2 Занятие: Полезные плагины QGIS . . . 314
9.2.1 Идем дальше: Плагин QuickMapServices . . . 314
9.2.2 Идем дальше: Плагин QuickOSM . . . 315
9.2.3 Идем дальше: Механизм запросов QuickOSM . . . 316
9.2.4 Идем дальше: Плагин DataPlotly . . . 318
9.2.5 В заключении . . . 323
9.2.6 Что дальше? . . . 323

10 Модуль: Онлайн ресурсы **325**
10.1 Занятие: Сервисы веб-карт . . . 325
10.1.1 Идем дальше: Загружаем слои WMS . . . 325
10.1.2 Попробуйте сами . . . 334
10.1.3 Попробуйте сами . . . 336

10.1.4 Попробуйте сами . . . 337
10.1.5 В заключении . . . 338
10.1.6 Дополнительные материалы для чтения . . . 338
10.1.7 Что дальше? . . . 338
10.2 Занятие: Веб-сервисы объектов . . . 338
10.2.1 Идем дальше: Загружаем слои WFS . . . 338
10.2.2 Идем дальше: Запрашиваем слои WFS . . . 342
10.2.3 В заключении . . . 345
10.2.4 Что дальше? . . . 345

11 Module: QGIS Server **347**
11.1 Lesson: Install QGIS Server . . . 347
11.1.1 Follow Along: Install from packages . . . 347
11.1.2 Follow Along: QGIS Server Executable . . . 348
11.1.3 HTTP Server Configuration . . . 348
11.1.4 Follow Along: Create another virtual host . . . 348
11.1.5 In Conclusion . . . 349
11.1.6 What's Next? . . . 349
11.2 Lesson: Serving WMS . . . 349
11.2.1 Логгирование . . . 352
11.2.2 GetMap requests . . . 353
11.2.3 Try Yourself Change the Image and Layers parameters . . . 354
11.2.4 Follow Along: Use Filter, Opacities and Styles parameters . . . 355
11.2.5 Follow Along: Use Redlining . . . 356
11.2.6 GetPrint requests . . . 357
11.2.7 In Conclusion . . . 359
11.2.8 What's Next? . . . 359

12 Module: GRASS **361**
12.1 Lesson: GRASS Setup . . . 361
12.1.1 Follow Along: Start a New GRASS Session . . . 361
12.1.2 Follow Along: Start a New GRASS Project . . . 364
12.1.3 Follow Along: Loading Vector Data into GRASS . . . 371
12.1.4 Follow Along: Loading Raster Data into GRASS . . . 375
12.1.5 Try Yourself Add Layers to Mapset . . . 377
12.1.6 Open an existing GRASS Mapset . . . 377
12.1.7 In Conclusion . . . 380
12.1.8 What's Next? . . . 380
12.2 Lesson: GRASS Tools . . . 380
12.2.1 Follow Along: Create an aspect map . . . 380
12.2.2 Follow Along: Get basic statistic of raster layer . . . 382
12.2.3 Follow Along: The Reclass Tool . . . 384
12.2.4 Try Yourself Reclassify with your rules . . . 386
12.2.5 Follow Along: The Mapcalc Tool . . . 387
12.2.6 In Conclusion . . . 389

13 Module: Assessment **391**
13.1 Create a base map . . . 391
13.1.1 Add the point layer . . . 391
13.1.2 Add the line layer . . . 392
13.1.3 Add the polygon layer . . . 393
13.1.4 Create the raster backdrop . . . 393
13.1.5 Finalize the base map . . . 393
13.2 Analyze the data . . . 393
13.2.1 / . . . 394
13.3 Final Map . . . 394

14 Module: Forestry Application **395**
14.1 Lesson: Forestry Module Presentation . . . 395

14.1.1 Forestry Sample Data . . . 396
14.2 Lesson: Georeferencing a Map . . . 396
14.2.1 Scan the map . . . 396
14.2.2 Follow Along: Georeferencing the scanned map . . . 396
14.2.3 In Conclusion . . . 402
14.2.4 What's Next? . . . 402
14.3 Lesson: Digitizing Forest Stands . . . 402
14.3.1 Follow Along: Extracting the Forest Stands Borders . . . 402
14.3.2 Try Yourself Georeference the Green Pixels Image . . . 405
14.3.3 Follow Along: Creating Supporting Points for Digitizing . . . 405
14.3.4 Follow Along: Digitize the Forest Stands . . . 407
14.3.5 Try Yourself Finish Digitizing the Forest Stands . . . 412
14.3.6 Follow Along: Joining the Forest Stand Data . . . 413
14.3.7 Try Yourself Renaming Attribute Names and Adding Area and Perimeter . . . 414
14.3.8 In Conclusion . . . 416
14.3.9 What's Next? . . . 416
14.4 Lesson: Updating Forest Stands . . . 416
14.4.1 Comparing the Old Forest Stands to Current Aerial Photographs . . . 416
14.4.2 Interpreting the CIR Image . . . 417
14.4.3 Try Yourself Digitizing Forest Stands from CIR Imagery . . . 419
14.4.4 Follow Along: Updating Forest Stands with Conservation Information . . . 420
14.4.5 Try Yourself Updating Forest Stands with Distance to the Stream . . . 425
14.4.6 In Conclusion . . . 426
14.4.7 What's Next? . . . 426
14.5 Lesson: Systematic Sampling Design . . . 426
14.5.1 Inventorying the Forest . . . 426
14.5.2 Follow Along: Implementing a Systematic Sampling Plot Design . . . 426
14.5.3 Follow Along: Exporting Sample Plots as GPX format . . . 430
14.5.4 In Conclusion . . . 431
14.5.5 What's Next? . . . 431
14.6 Lesson: Creating Detailed Maps with the Atlas Tool . . . 431
14.6.1 Follow Along: Preparing the Print Layout . . . 431
14.6.2 Follow Along: Adding Background Map . . . 433
14.6.3 Try Yourself Changing the Symbology of the Layers . . . 433
14.6.4 Try Yourself Create a Basic Map Template . . . 435
14.6.5 Follow Along: Adding More Elements to the Print Layout . . . 436
14.6.6 Follow Along: Creating an Atlas Coverage . . . 438
14.6.7 Follow Along: Setting Up the Atlas Tool . . . 440
14.6.8 Follow Along: Editing the Coverage Layer . . . 442
14.6.9 Follow Along: Printing the Maps . . . 445
14.6.10 In Conclusion . . . 445
14.6.11 What's Next? . . . 446
14.7 Lesson: Calculating the Forest Parameters . . . 446
14.7.1 Follow Along: Adding the Inventory Results . . . 446
14.7.2 Follow Along: Whole Forest Parameters Estimation . . . 447
14.7.3 Follow Along: Estimating Stand Parameters . . . 447
14.7.4 In Conclusion . . . 451
14.7.5 What's Next? . . . 452
14.8 Lesson: DEM from LiDAR Data . . . 452
14.8.1 Follow Along: Installing Lastools . . . 452
14.8.2 Follow Along: Calculating a DEM with LAStools . . . 454
14.8.3 Follow Along: Creating a Terrain Hillshade . . . 459
14.8.4 In Conclusion . . . 461
14.8.5 What's Next? . . . 462
14.9 Lesson: Map Presentation . . . 462
14.9.1 Follow Along: Preparing the Map Data . . . 462
14.9.2 Try Yourself Try Different Blending Modes . . . 464
14.9.3 Try Yourself Using a Layout Template to Create the Map result . . . 465

14.9.4 In Conclusion . . . 468

15 Module: Database Concepts with PostgreSQL 469
15.1 Lesson: Introduction to Databases . . . 469
15.1.1 What is a Database? . . . 469
15.1.2 Tables . . . 469
15.1.3 Columns / Fields . . . 470
15.1.4 Records . . . 470
15.1.5 Datatypes . . . 470
15.1.6 Modelling an Address Database . . . 471
15.1.7 Database Theory . . . 471
15.1.8 Normalisation . . . 471
15.1.9 Try Yourself . . . 472
15.1.10 Indexes . . . 473
15.1.11 Sequences . . . 473
15.1.12 Entity Relationship Diagramming . . . 473
15.1.13 Constraints, Primary Keys and Foreign Keys . . . 475
15.1.14 Transactions . . . 476
15.1.15 In Conclusion . . . 476
15.1.16 What's Next? . . . 476
15.2 Lesson: Implementing the Data Model . . . 476
15.2.1 Install PostgreSQL . . . 476
15.2.2 Справка . . . 477
15.2.3 Create a database user . . . 477
15.2.4 Verify the new account . . . 477
15.2.5 Create a database . . . 478
15.2.6 Starting a database shell session . . . 478
15.2.7 Make Tables in SQL . . . 478
15.2.8 Create Keys in SQL . . . 480
15.2.9 Create Indexes in SQL . . . 481
15.2.10 Dropping Tables in SQL . . . 481
15.2.11 A word on pgAdmin III . . . 482
15.2.12 In Conclusion . . . 482
15.2.13 What's Next? . . . 482
15.3 Lesson: Adding Data to the Model . . . 482
15.3.1 Insert statement . . . 482
15.3.2 Sequencing Data Addition According to Constraints . . . 483
15.3.3 Try Yourself . . . 483
15.3.4 Select data . . . 484
15.3.5 Update data . . . 484
15.3.6 Delete Data . . . 484
15.3.7 Try Yourself . . . 485
15.3.8 In Conclusion . . . 485
15.3.9 What's Next? . . . 485
15.4 Lesson: Queries . . . 485
15.4.1 Ordering Results . . . 485
15.4.2 Filtering . . . 486
15.4.3 Joins . . . 487
15.4.4 Sub-Select . . . 487
15.4.5 Aggregate Queries . . . 488
15.4.6 In Conclusion . . . 489
15.4.7 What's Next? . . . 489
15.5 Lesson: Представления . . . 489
15.5.1 Создание представления . . . 490
15.5.2 Изменение представления . . . 490
15.5.3 Удаление представления . . . 491
15.5.4 In Conclusion . . . 491
15.5.5 What's Next? . . . 491

15.6 Lesson: Rules . . . 491
15.6.1 Creating a logging rule . . . 491
15.6.2 In Conclusion . . . 492
15.6.3 What's Next? . . . 492

16 Module: Spatial Database Concepts with PostGIS **493**
16.1 Lesson: PostGIS Setup . . . 493
16.1.1 Installing under Ubuntu . . . 494
16.1.2 Installing under Windows . . . 494
16.1.3 Installing on Other Platforms . . . 494
16.1.4 Configuring Databases to use PostGIS . . . 494
16.1.5 Looking at the installed PostGIS functions . . . 495
16.1.6 Spatial Reference Systems . . . 496
16.1.7 In Conclusion . . . 496
16.1.8 What's Next? . . . 497
16.2 Lesson: Simple Feature Model . . . 497
16.2.1 What is OGC . . . 497
16.2.2 What is the SFS Model . . . 497
16.2.3 Add a geometry field to table . . . 498
16.2.4 Add a constraint based on geometry type . . . 498
16.2.5 Try Yourself . . . 498
16.2.6 Populate geometry_columns table . . . 498
16.2.7 Add geometry record to table using SQL . . . 499
16.2.8 In Conclusion . . . 502
16.2.9 What's Next? . . . 502
16.3 Lesson: Import and Export . . . 502
16.3.1 shp2pgsql . . . 502
16.3.2 pgsql2shp . . . 503
16.3.3 ogr2ogr . . . 503
16.3.4 DB Manager . . . 503
16.3.5 In Conclusion . . . 503
16.3.6 What's Next? . . . 503
16.4 Lesson: Spatial Queries . . . 504
16.4.1 Spatial Operators . . . 504
16.4.2 Spatial Indexes . . . 504
16.4.3 Try Yourself . . . 505
16.4.4 PostGIS Spatial Functions Demo . . . 505
16.4.5 In Conclusion . . . 511
16.4.6 What's Next? . . . 511
16.5 Lesson: Geometry Construction . . . 511
16.5.1 Creating Linestrings . . . 511
16.5.2 Try Yourself . . . 511
16.5.3 Creating Polygons . . . 512
16.5.4 Exercise: Linking Cities to People . . . 512
16.5.5 Looking at Our Schema . . . 514
16.5.6 Try Yourself . . . 514
16.5.7 Access Sub-Objects . . . 514
16.5.8 Data Processing . . . 514
16.5.9 Clipping . . . 515
16.5.10 Building Geometries from Other Geometries . . . 516
16.5.11 Geometry Cleaning . . . 518
16.5.12 Differences between tables . . . 518
16.5.13 Tablespaces . . . 519
16.5.14 In Conclusion . . . 519

17 The QGIS processing guide **521**
17.1 Введение . . . 521
17.2 An important warning before starting . . . 522

17.3 Setting-up the processing framework . . . 523
17.4 Running our first algorithm. The toolbox . . . 525
17.5 More algorithms and data types . . . 529
17.6 CRSs. Reprojecting . . . 536
17.7 Selection . . . 539
17.8 Running an external algorithm . . . 541
17.9 The processing log . . . 546
17.9.1 Advanced . . . 548
17.10 The raster calculator. No-data values . . . 548
17.11 Калькулятор для векторов . . . 552
17.12 Defining extents . . . 557
17.13 HTML outputs . . . 561
17.14 First analysis example . . . 563
17.15 Clipping and merging raster layers . . . 572
17.16 Hydrological analysis . . . 581
17.17 Starting with the graphical modeler . . . 593
17.18 More complex models . . . 604
17.19 Numeric calculations in the modeler . . . 609
17.20 A model within a model . . . 614
17.21 Using modeler-only tools for creating a model . . . 615
17.22 Interpolation . . . 620
17.23 More interpolation . . . 630
17.24 Iterative execution of algorithms . . . 637
17.25 More iterative execution of algorithms . . . 641
17.26 Интерфейс пакетной обработки . . . 643
17.27 Models in the batch processing interface . . . 646
17.28 Хуки пред- и постобработки . . . 647
17.29 Other programs . . . 648
17.29.1 GRASS . . . 648
17.29.2 R . . . 648
17.29.3 Others . . . 649
17.29.4 Comparison among backends . . . 649
17.30 Interpolation and contouring . . . 650
17.30.1 Interpolation . . . 650
17.30.2 Contour . . . 650
17.31 Vector simplification and smoothing . . . 650
17.32 Planning a solar farm . . . 651
17.33 Use R scripts in Processing . . . 651
17.33.1 Adding scripts . . . 652
17.33.2 Creating plots . . . 653
17.33.3 Create a vector . . . 656
17.33.4 Text and graph output from R - syntax . . . 659
17.34 Predicting landslides . . . 659

18 Module: Using Spatial Databases in QGIS **661**
18.1 Lesson: Working with Databases in the QGIS Browser . . . 661
18.1.1 Follow Along: Adding Database Tables to QGIS using the Browser . . . 661
18.1.2 Follow Along: Adding a filtered set of records as a Layer . . . 663
18.1.3 In Conclusion . . . 664
18.1.4 What's Next? . . . 664
18.2 Lesson: Using DB Manager to work with Spatial Databases in QGIS . . . 665
18.2.1 Follow Along: Managing PostGIS Databases with DB Manager . . . 665
18.2.2 Follow Along: Creating a New Table . . . 671
18.2.3 Follow Along: Basic Database Administration . . . 672
18.2.4 Follow Along: Executing SQL Queries with DB Manager . . . 673
18.2.5 Importing Data into a Database with DB Manager . . . 674
18.2.6 Exporting Data from a Database with DB Manager . . . 676
18.2.7 In Conclusion . . . 678

18.2.8 What's Next? . . . 678
18.3 Lesson: Working with SpatiaLite databases in QGIS . . . 678
18.3.1 Follow Along: Creating a SpatiaLite database with the Browser . . . 678
18.3.2 In Conclusion . . . 680

19 Appendix: Contributing To This Manual **681**
19.1 Downloading Resources . . . 681
19.2 Manual Format . . . 681
19.3 Adding a Module . . . 681
19.4 Adding a Lesson . . . 682
19.5 Adding a Section . . . 683
19.5.1 Adding a «follow along» section . . . 683
19.5.2 Adding a «try yourself» section . . . 684
19.6 Add a Conclusion . . . 684
19.7 Add a Further Reading Section . . . 684
19.8 Add a What's Next Section . . . 685
19.9 Using Markup . . . 685
19.9.1 New concepts . . . 685
19.9.2 Emphasis . . . 685
19.9.3 Images . . . 685
19.9.4 Internal links . . . 685
19.9.5 External links . . . 686
19.9.6 Using monospaced text . . . 686
19.9.7 Labeling GUI items . . . 686
19.9.8 Menu selections . . . 686
19.9.9 Adding notes . . . 686
19.9.10 Adding a sponsorship/authorship note . . . 687
19.10 Thank You! . . . 687

20 Подготовка данных для занятий **689**
20.1 Try Yourself Create OSM based vector Files . . . 689
20.2 Try Yourself Создание файлов цифровой модели местности . . . 696
20.3 Try Yourself Create imagery tiff Files . . . 696
20.4 Try Yourself Замените обозначения . . . 697

ГЛАВА 1

Введение

1.1 Предисловие

Добро пожаловать на наш курс! Мы покажем вам, как легко и эффективно использовать QGIS. Если вы новичок в GIS, мы расскажем, что вам нужно для начала. Если вы опытный пользователь, вы увидите, как QGIS выполняет все функции, которые вы ожидаете от программы GIS, и многое другое!

1.1.1 Почему QGIS?

Поскольку информация становится все более осознаваемой пространственно, то здесь нет недостатка в инструментах, которые могут выполнять некоторые или все функции, широко используемые в GIS. Почему использовать именно QGIS и какова эта программа по сравнению с другими программными пакетами GIS?

Вот лишь некоторые причины:

- *Она бесплатная, как при бесплатном обеде*. Установка и использование программы QGIS не потребует от вас никаких денежных затрат. Никаких первоначальных взносов, никаких периодических платежей, ничего.
- *Она бесплатная, так как является открытой программой*. Если вам нужны дополнительные функции в QGIS, вы можете не просто надеяться, что эти функции будут включены в выход следующей версии. Вы можете спонсировать разработку функции или добавить ее самостоятельно, если вы знакомы с программированием.
- *Она постоянно развивается*. Поскольку каждый может добавлять новые функции и улучшать существующие, QGIS никогда не стоит на месте. Разработка нового инструмента может происходить настолько быстро, насколько вам это нужно.
- *Доступна обширная документация и помощь*. Если вы на чем-то застряли, вы можете обратиться к обширной документации, к другим пользователям QGIS или даже к разработчикам.
- *Кросс-платформенность*. QGIS можно установить на MacOS, Windows и Linux.

Теперь, когда вы знаете, почему вы хотите использовать QGIS, данные упражнения вас научат тому, как это сделать.

1.1.2 Предпосылки

В 2008 году мы запустили Краткое введение в GIS, абсолютно бесплатный ресурс с открытым контентом для людей, которые хотят получить информацию о GIS, не перегруженную профессиональным жаргоном и новой терминологией. Проект был спонсирован правительством Южной Африки и имел феноменальный успех: люди со всего мира писали нам, чтобы рассказать о том? как они используют эти материалы для проведения обучающих курсов в университетах, самообучения GIS и так далее. Краткое введение - это не учебник по программному обеспечению, а скорее составлялся как общий текст для тех, кто изучает GIS (хотя мы использовали QGIS во всех примерах). Также имеется руководство QGIS, в котором представлен подробный функциональный обзор приложения QGIS. Однако он структурирован не как учебное пособие, а как справочное руководство. В компании Линфинити Консалтинг СиСи/ Linfiniti Consulting CC, так как мы часто проводим учебные курсы, мы поняли, что необходим третий ресурс - тот, который ведет читателя последовательно через изучение ключевых аспектов QGIS в формате «тренер-ученик», что и побудило нас выполнить эту работу.

Это учебное пособие предназначено для предоставления всех материалов, необходимых для проведения 5-дневного курса по QGIS, PostgreSQL и PostGIS. Курс структурирован таким образом, чтобы удовлетворить потребности как новичков, так и продвинутых пользователей, и содержит множество упражнений с аннотированными ответами по всему тексту.

1.1.3 Лицензия



Бесплатное учебное пособие Quantum GIS компании Linfiniti Consulting CC основано на ранней версии от компании Linfiniti и имеет лицензию от `Creative Commons Attribution 4.0 International `_. Разрешения, выходящие за рамки данной лицензии, доступны по ссылке ниже.

Мы опубликовали это учебное пособие по QGIS с лицензированием на свободных условиях, которая позволяет вам свободно копировать, изменять и распространять эту работу. Полная версия лицензии доступна в конце этого документа. Проще говоря, руководство по использованию заключается в следующем:

- Вы не можете представлять данную работу как свою собственную, а также удалять какие-либо авторские тексты или ссылки на эту работу.
- Вы не можете перераспределять эту работу с более ограниченными разрешениями, чем те, в соответствии с которыми она была предоставлена вам.
- Если вы проделали существенную часть работы, а затем вносите это в проект (по крайней мере, один полный модуль), то вы можете добавить свое имя в конец списка авторов данного документа (который появится на первой странице).
- Если вы вносите незначительные изменения и исправления, вы можете добавить себя в список участников ниже.
- Если вы переводите этот документ полностью, вы можете добавить свое имя в список авторов в форме «Переведено Джо Блоггс».
- Если вы спонсируете модуль или урок, вы можете попросить автора включить благодарность в начале каждого урока, например:

Примечание: Спонсором данного урока является компания MegaCorp.

- Если вы не уверены в том, что именно вы можете делать в рамках этой лицензии, свяжитесь с нами по адресу office@linfiniti.com, мы подскажем вам, приемлемо ли то, что вы собираетесь сделать.
- Если вы размещаете данную работу на вебсайте самостоятельного размещения, таком как https://www.lulu.com, мы просим вас пожертвовать прибыль проекту QGIS.

- Вы не можете использовать данную работу в коммерческих целях, за исключением случаев, когда разрешения авторов были ими предоставлены. Для ясности, под коммерциализацией мы подразумеваем, что вы не можете продавать продукт с целью получения прибыли, создавать производные коммерческие работы (например, продавать контент для использования в качестве статей в журнале). Исключением являются случаи, когда вся прибыль направляется проекту QGIS. Вы можете (и мы рекомендуем вам это делать) использовать эту работу в качестве учебника при проведении учебных курсов, даже если сам курс носит коммерческий характер. Другими словами, вы можете зарабатывать деньги, проводя учебный курс, в котором эта работа используется в качестве учебника, но вы не можете получать прибыль от продажи самой книги - вся такая прибыль должна быть возвращена в QGIS.

1.1.4 Спонсорские главы

Данная работа ни в коем случае не является полным трактатом обо всем, что вы можете делать в QGIS, и мы призываем вас добавлять новые материалы, чтобы заполнить любые пробелы. Компания Linfiniti Consulting CC может также создавать для вас дополнительные материалы в качестве коммерческой услуги, с пониманием того, что все подобные работы должны стать частью основного контента и размещаться под той же лицензией.

1.1.5 Авторы

- Rüdiger Thiede - Rudi has written the QGIS instructional materials and parts of the PostGIS materials.
- Tim Sutton (tim@kartoza.com) - Tim has overseen and guided the project and co-authored the PostgreSQL and PostGIS parts. Tim also authored the custom sphinx theme used for this manual.
- Хорст Дюстер (horst.duester@kappasys.ch) - Хорст является соавтором в части PostgreSQL и PostGIS.
- Marcelle Sutton - Marcelle provided proof-reading and editorial advice during the creation of this work.

1.1.6 Индивидуальные вкладчики

Здесь должно быть ваше имя!

1.1.7 Спонсоры

- Технологический университет Кейп Пенинсула Cape Peninsula University of Technology.

1.1.8 Исходные файлы и отчеты о проблемах

Источник данного документа доступен на GitHub `QGIS Documentation repository `_. Проконсультируйтесь с `GitHub.com `_ для получения инструкций о том, как использовать систему контроля версии git.

Несмотря на все наши усилия, вы можете найти кое-какие ошибки или пропустить какую-нибудь информацию в ходе данного курса. Пожалуйста, сообщите о них на https://github.com/qgis/QGIS-Documentation/issues.

1.1.9 Последняя версия

Вы всегда можете получить последнюю версию данного документа, онлайн версию, которая является частью QGIS на веб-сайте документации, посетив (https://docs.qgis.org).

Примечание: На веб-сайте документации имеются ссылки как на онлайн-версию, так и на PDF-версию Учебного пособия и другую документации QGIS.

1.2 Об упражнениях

Теперь, когда вы знаете почему вы хотите использовать QGIS, мы можем показать вам, как это делать.

Предупреждение: Данный курс включает в себя инструкции по тому как добавить, удалить и изменить наборы данных GIS. В этих целях мы предоставили учебные наборы данных. Прежде чем использовать описанные здесь методы для ваших собственных данных, всегда проверяйте наличие надлежащих резервных копий!

1.2.1 Как использовать данное руководство

Любой текст, который *looks like this* относится к чему-то, что вы видите в пользовательском интерфейсе QGIS.

Текст, который *looks* ► *like* ► *this* ведет вас через все меню.

Такого рода текст `This kind of text` относится к чему-то, что вы можете напечатать/набрать, например, команду.

Такого/рода/текст.text `This/kind/of.text` относится к пути или имени файла.

Такой+Такой `This+That` относится к комбинации клавиш, состоящей из двух кнопок.

1.2.2 Цели многоуровневого курса

Данный курс ориентирован на разные пользовательские уровни. В зависимости от того, к какой категории вы относитесь, вы можете ожидать разные наборы результатов курса. Каждая категория содержит информацию, которая важна для последующей, поэтому важно выполнять все упражнения, которые соответствуют вашему пользовательскому уровню или ниже вашего пользовательского уровня.



Базовый

В данной категории курс предполагает, что у вас имеется небольшой или совсем отсутствует опыт в теоретических знаниях в области GIS или работы с программным обеспечением GIS.

Набольшая теоретическая база все же будет предоставлена для объяснения цели действия, которое вы будете выполнять в программе, но упор делается на обучение на собственном опыте.

По завершении курса вы получите лучшее представление о возможностях GIS и о том, как использовать ее возможности с помощью QGIS.



Средний

В данной категории предполагается, что у вас есть рабочие знания и опыт повседневного использования программного обеспечения GIS.

Применение инструкций для начального уровня даст вам знакомую базу, а также расскажет о случаях, когда QGIS работает несколько иначе, чем другое программное обеспечение, к которому вы, возможно, привыкли. Вы также узнаете, как применять функции анализа в QGIS.

По окончании курса вы научитесь использовать все функции QGIS, которые обычно необходимы для повседневного использования.



Продвинутый

В данной категории предполагается, что у вас есть опыт работы с программным обеспечением GIS, знания и опыт работы с пространственными базами данных, применение данных на удаленном сервере, возможно, написание сценариев для анализа и т.д.

Применение инструкций для двух других уровней познакомит вас с подходом, которому следует интерфейс QGIS, и обеспечит вам знание того, как получить доступ к основным функциям, которые вам нужны. Вам также будет показано, как использовать систему плагинов QGIS, доступ к базе данных и так далее.

По окончании курса вы должны быть хорошо осведомлены о повседневной работе с QGIS, а также с ее более сложными функциями.

1.2.3 Данные

Образцы данных, прилагаемые к этому ресурсу, находятся в свободном доступе и поступают из следующих источников:

- Наборы данных Улицы и места / Streets and Places из OpenStreetMap (https://www.openstreetmap.org/)
- Property boundaries (urban and rural), water bodies from NGI (https://ngi.dalrrd.gov.za/)
- Радиолокационная топографическая миссия шаттла / SRTM DEM от CGIAR-CGI (http://srtm.csi.cgiar.org/)

Загрузите подготовленный набор данных из Учебного Репозитория / Training data repository и разархивируйте файл (Zip). Все необходимые данные находятся в папке `exercise_data` (упражнения_данные).

Если вы инструктор и хотите использовать больше соответствующих данных, вы найдете инструкции по созданию локальных данных в приложении Подготовка данных для упражнений / *Подготовка данных для занятий*.

ГЛАВА 2

Модуль: Создание и изучение базовой карты

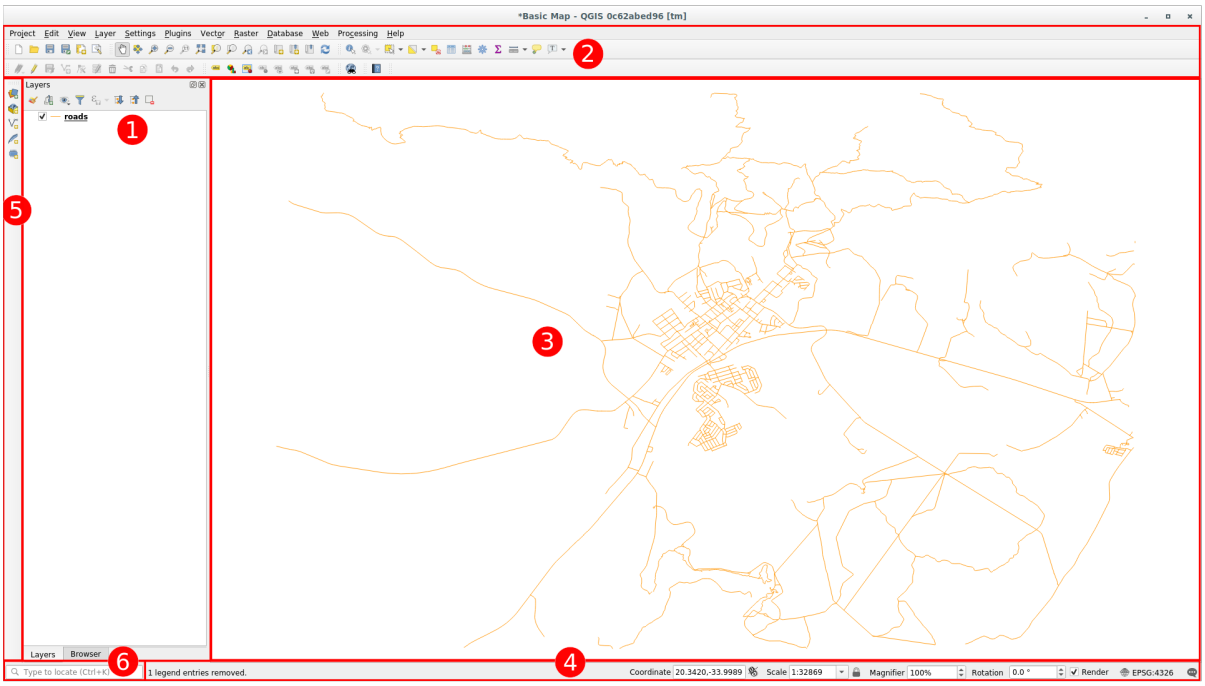
В данном модуле вы создадите базовую карту, которая будет использоваться позже в качестве основы для дальнейших демонстраций функциональности QGIS.

2.1 Занятие: Обзор интерфейса

Мы изучим пользовательский интерфейс QGIS, чтобы вы ознакомились с меню, панелью инструментов, полотном карты и списком слоев, которые образуют базовую структуру интерфейса.

Цель этого занятия: понять основы пользовательского интерфейса QGIS..

2.1.1 Попробуйте сами: Основы



Элементами, установленными на картинке выше, являются:

1. Список слоев / панель браузера
2. Панели инструментов
3. Полотно карты
4. Строка состояния
5. Боковая панель инструментов
6. Панель локатора

Список слоёв

В Списке слоев вы можете увидеть список всех доступных вам слоев в любое время.

Если раскрыть свернутые элементы (кликнув стрелку или плюсик рядом с ним), вы получите дополнительную информацию о текущем внешнем виде слоя.

При наведении курсора на слои вы получите некоторую основную информацию: имя слоя, тип геометрии, систему координат и полный путь местоположения на вашем устройстве.

Клик правой кнопкой мыши по слою откроет вам меню со множеством дополнительных опций. Вы скоро будете использовать некоторые из них, так что присмотритесь!

Примечание: Векторный слой - это набор данных, обычно состоящий из объектов определенного типа, таких как дороги, деревья и т.д. Векторный слой может состоять из точек, линий или полигонов.

Панель браузера

Браузер QGIS - это панель QGIS, которая позволяет легко перемещаться по базе данных. У вас может быть доступ к общим векторным файлам (например, файлам ESRI Shapefile или MapInfo), базам данных (например, PostGIS, Oracle, SpatiaLite, GeoPackage или MSSQL Spatial) и связям WMS / WFS. Вы также можете просмотреть свои данные GRASS.

Если вы сохранили проект, панель браузера также предоставит вам быстрый доступ ко всем слоям, хранящимся по тому же пути к файлу проекта  *Project Home*.

Кроме того, вы можете создать одну или несколько папок в качестве **Избранных**: выполните поиск по своему пути и, как только вы найдете папку, кликните ее правой кнопкой мыши и выберите Добавить в избранное `Add as a Favorite`. После этого вы сможете увидеть свою папку в Избранных  *Favorites*.

Совет: Может случиться так, что папки, добавленные в Избранное, имеют очень длинные имена: не беспокойтесь, кликните правой кнопкой мыши путь и выберите Переименовать избранное `Rename Favorite...`, чтобы задать другое имя.



Панель инструментов

Наиболее часто используемые наборы инструментов можно преобразовать в панель инструментов для основного доступа. Например, панель инструментов Файл позволяет сохранять, загружать, распечатывать и запускать новый проект. Вы можете легко настроить интерфейс так, чтобы отображались только наиболее часто используемые инструменты, добавляя или удаляя панели инструментов по мере необходимости с помощью меню *View* ► *Toolbars*.

Даже если они не отображаются на панели инструментов, все ваши инструменты останутся доступными через меню. Например, если вы удалите панель инструментов *File* (в которой есть кнопка Сохранить *Save*), вы все равно сможете сохранить карту, кликнув меню Проект *Project* а затем нажав *Save*.



Полотно карты

Здесь отображается сама карта и загружаются слои. На полотне карты вы можете взаимодействовать с видимыми слоями: приближать / отдалять, перемещать карту, выбирать функции и многие другие операции, которые мы подробно рассмотрим в следующих разделах.



Строка состояния

Показывает вам информацию о текущей карте. Также позволяет настраивать масштаб карты, поворот карты и видеть координаты курсора мыши на карте.

 Панель инструментов

По умолчанию Боковая панель инструментов содержит кнопки для загрузки слоя и все кнопки для создания нового слоя. Но помните, что вы можете переместить все панели инструментов в любое удобное для вас место.

 Панель локатора

На данной панели вы можете получить доступ почти ко всем объектам QGIS: слоям, функциям слоя, алгоритмам, пространственным закладкам и т.д. Проверьте все различные параметры в разделе Настройки локатора locator_options Руководства пользователя QGIS.

Совет: С помощью сочетания клавиш `Ctrl+K` увы можете легко получить доступ к панели.

2.1.2  Попробуйте сами 1

Попытайтесь определить четыре элемента, перечисленных выше, на собственном экране, не обращаясь к диаграмме выше. Посмотрите, сможете ли вы определить их наименования и функции. Вы лучше познакомитесь с этими элементами по мере их использования в ближайшие дни.

Answer

Refer back to the image showing the interface layout and check that you remember the names and functions of the screen elements.

2.1.3  Попробуйте сами 2

Попытайтесь найти каждый инструмент у себя на экране. Какое у них предназначение?

1. 
2. 
3. 
4.  *Render*
5. 

Примечание: Если какой-либо из этих инструментов не отображается на экране, попробуйте запустить некоторые панели инструментов, которые в настоящее время скрыты. Также имейте в виду, что если на экране недостаточно места, панель инструментов может быть укорочена и скрывать некоторые инструменты. Вы можете увидеть скрытые инструменты, кликнув на кнопку с двойной стрелкой вправо на любой такой свернутой панели инструментов. Вы можете увидеть всплывающую подсказку с названием любого инструмента, удерживая курсор мыши над инструментом некоторое время.

Answer

1. *Save as*
2. *Zoom to layer(s)*
3. *Invert selection*
4. *Rendering on/off*
5. *Measure line*

2.1.4 Что дальше?

Теперь, когда вы знакомы с основами интерфейса QGIS, в следующем занятии вы увидите, как загрузить некоторые распространенные типы данных.

2.2 Занятие: Добавляем ваши первые слои

Мы запустим приложение и создадим базовую карту для использования в примерах и упражнениях.

Цель этого занятия: начать работу с образцом карты.

Примечание: Прежде чем приступить к этому упражнению на вашем компьютере, необходимо установить QGIS. Кроме того, вы должны были загрузить *sample data* для использования.

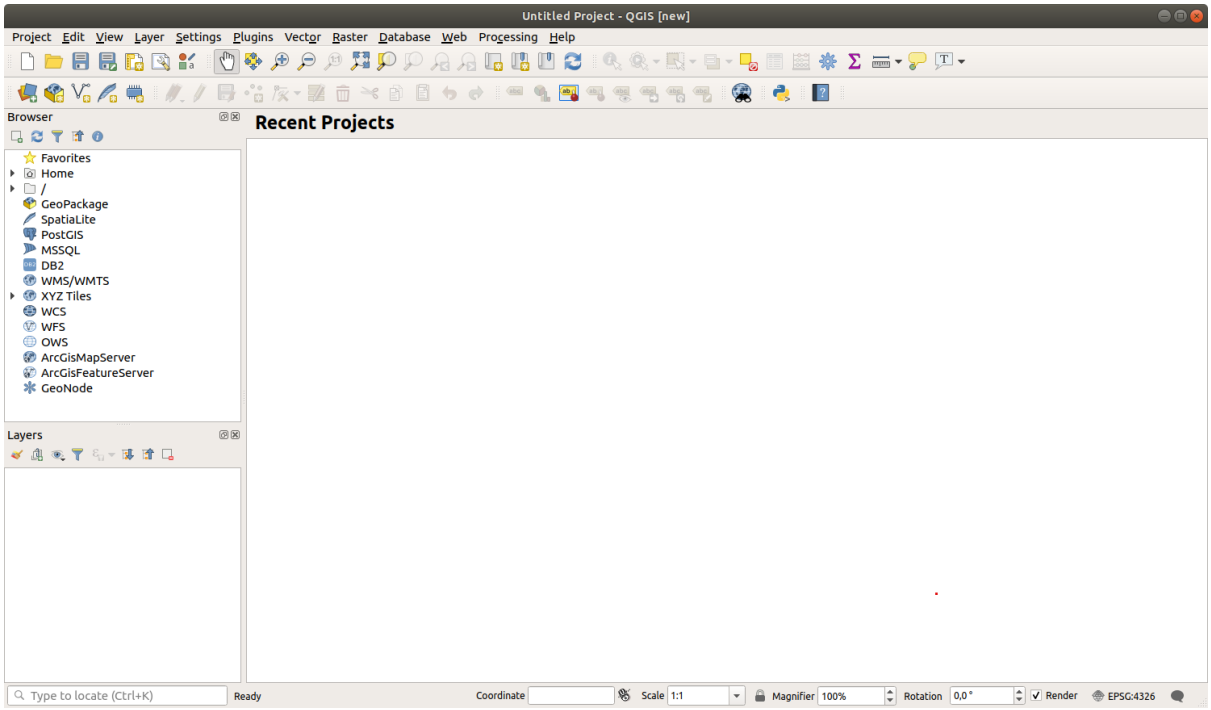
Запустите QGIS с его ярлыка на рабочем столе, пункта меню и т.д., в зависимости от того, как вы настроили его установку.

Примечание: Скриншоты для данного курса были сделаны в QGIS 3.4 на Linux. В зависимости от ваших настроек, скрины, с которыми вы сталкиваетесь, могут выглядеть несколько иначе. Однако все те же кнопки по-прежнему будут доступны, а инструкция будет работать на любой OS. Вам необходима QGIS 3.4 (последняя версия на момент написания) для участия в данном курсе.

Давайте начнем прямо сейчас!

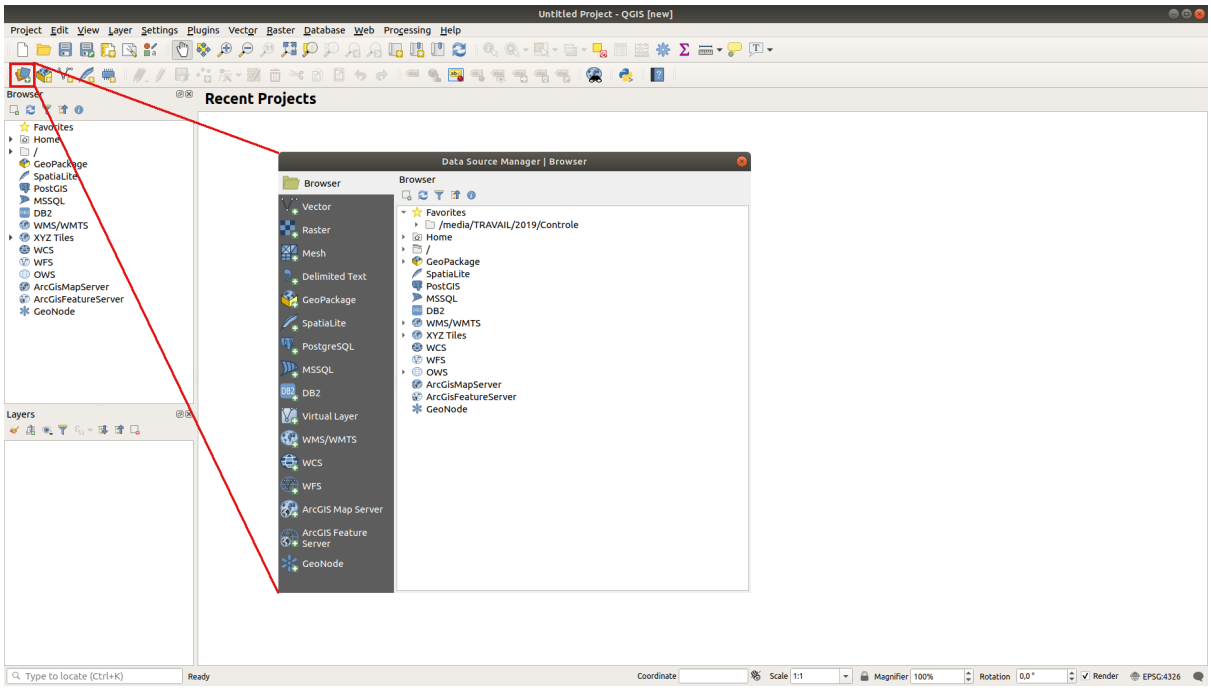
2.2.1  Идем дальше: Подготовим карту

1. Откройте QGIS. У вас появится новая, чистая карта.



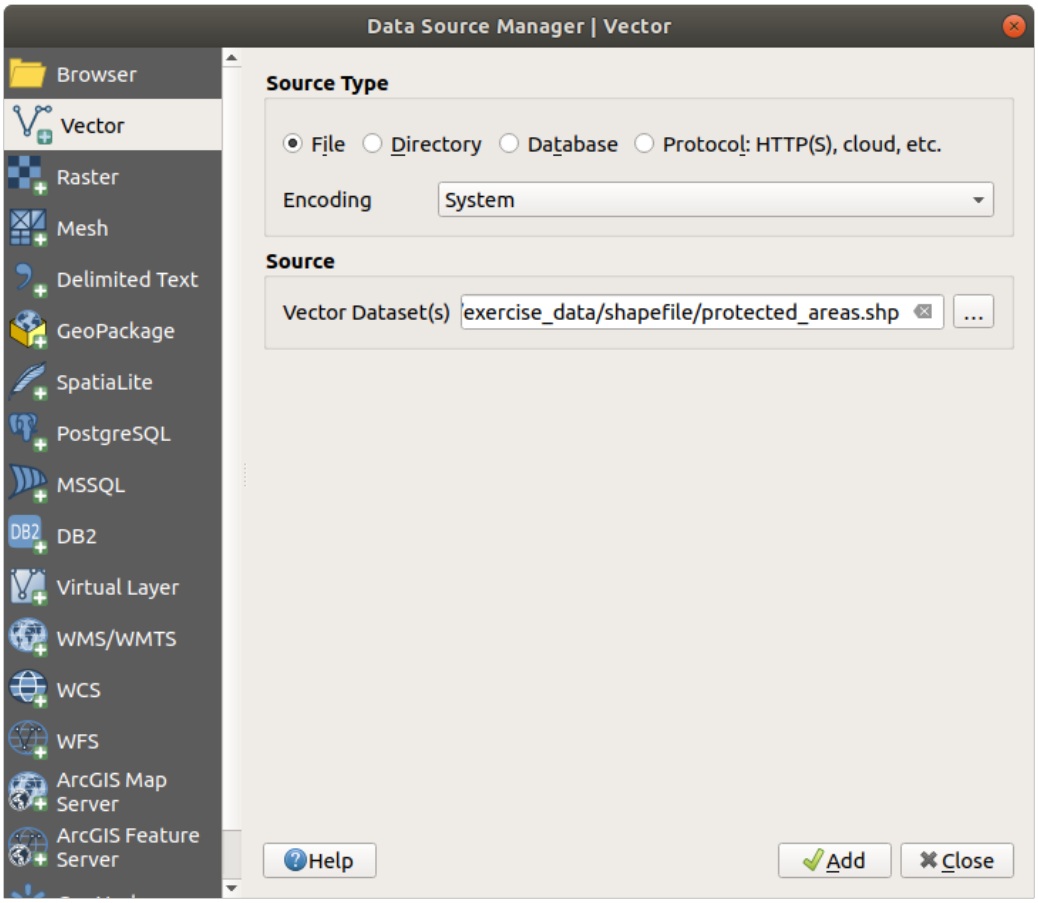
2. Диалоговое окно Диспетчер источников данных *Data Source Manager* позволяет вам выбрать данные для загрузки в зависимости от типа данных. Мы будем использовать его для загрузки нашего набора данных: кликните кнопку Диспетчер открытых источников данных  Open Data Source Manager.

   Если вы не можете найти иконку, убедитесь, что панель инструментов Диспетчер источников данных *Data Source Manager* запущен в меню *View* ► *Toolbars*.

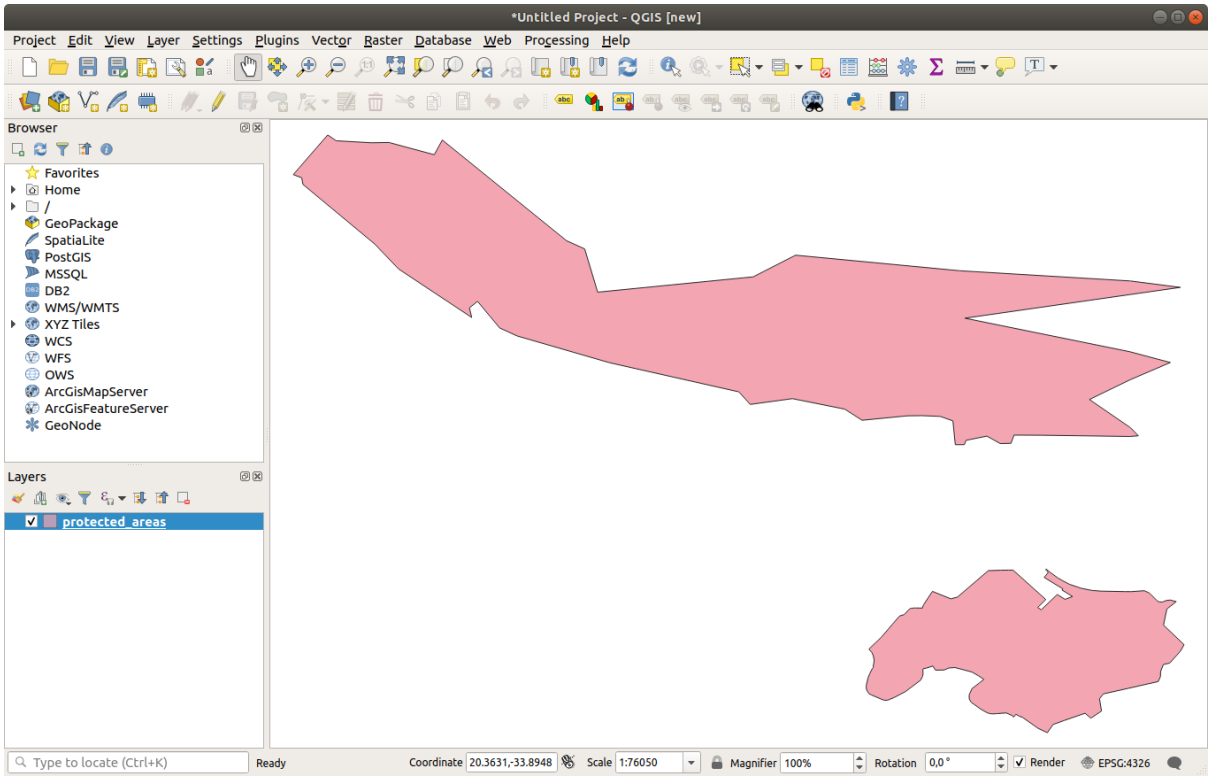


3. Загрузите набор векторных данных Охраняемые территории `protected_areas.shp`:
   1. Кликните по вкладке *Vector*.
   2. Запустите тип источника  *File*.

3. Нажмите на кнопку многоточие … рядом с Набор(ы) векторных данных *Vector Dataset(s)*.
4. Выберите файл `exercise_data/shapefile/protected_areas.shp` в вашем учебном каталоге.
5. Кликните Открыть *Open*. Появится исходное диалоговое окно с заполненным путем к файлу.



6. Кликните также Добавить *Add* сюда. Указанные вами данные теперь загрузятся: вы увидите элемент `protected_areas` на панели Слои *Layers* (внизу, слева) с его функциями, показанными на основном полотне карты.



Поздравляем! Теперь у вас есть базовая карта. Сейчас необходимо сохранить вашу работу.

1. Кликните на кнопку Сохранить как *Save As*: 
2. Сохраните карту в папке Решение `solution` folder рядом с `exercise_data` и назовите ее `basic_map.qgz`.

2.2.2  Попробуйте сами

Повторите шаги, описанные выше для добавления слоев Места `places.shp` и Реки `rivers.shp` из той же папки (`exercise_data/shapefile`) в карту.

Answer

In the main area of the dialog you should see many shapes with different colors. Each shape belongs to a layer you can identify by its color in the left panel (your colors may be different from the ones in Рис. 2.1 below):

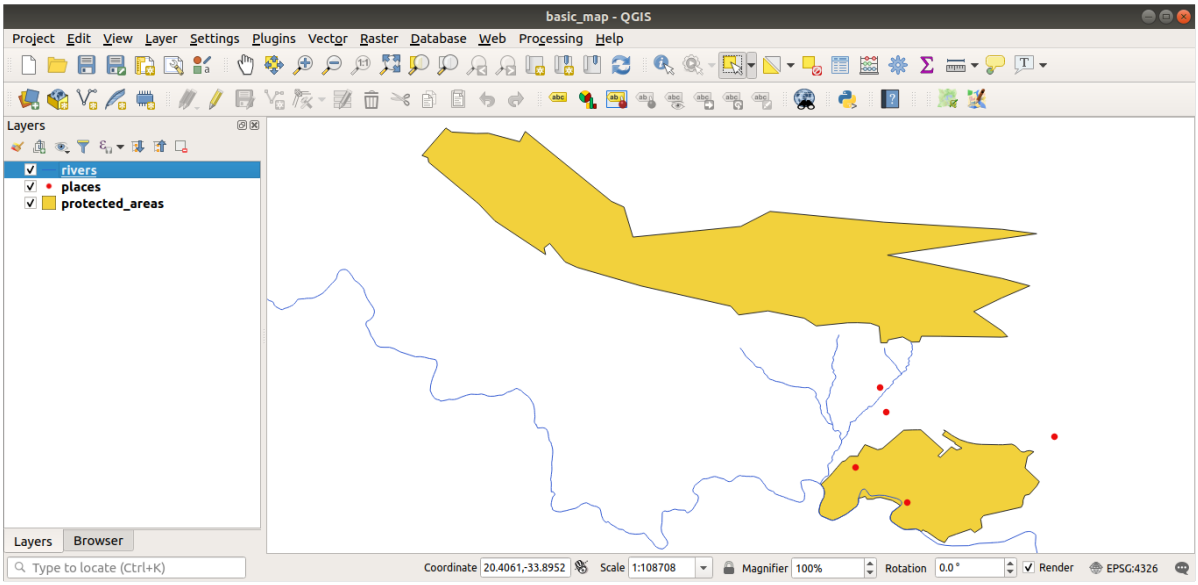


Рис. 2.1: A basic map

2.2.3 Идем дальше: Загружаем векторные данные из Базы данных GeoPackage

Базы данных позволяют хранить большой объем связанных данных в одном файле. Возможно, вы уже знакомы с Системой управления базами данных (СУБД / DBMS), такой как Libreoffice Base или MS Access. Приложения GIS также могут использовать базы данных. Системы СУБД, ориентированные на GIS (например, PostGIS) имеют дополнительные функции, поскольку им необходимо обрабатывать пространственные данные.

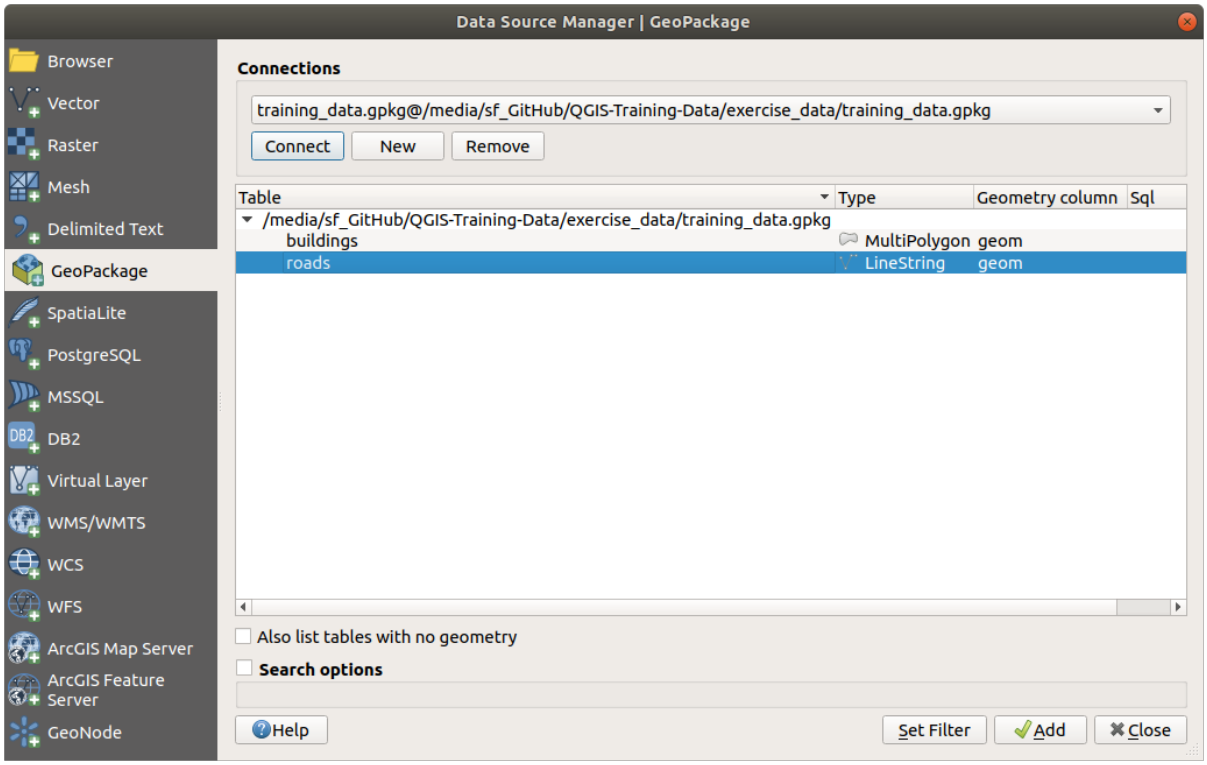
Открытый формат *GeoPackage* это контейнер, который позволяет хранить данные (слои) GIS в одном файле. В отличие от формата ESRI Shapefile (например, набор данных `protected_areas.shp`, который вы загрузили ранее), один файл GeoPackage может содержать различные данные (как векторные, так и растровые) в разных системах координат, а также таблицы без пространственной информации; все эти функции позволяют легко обмениваться данными и избегать дублирования файлов.

Чтобы загрузить слой из GeoPackage, вам сначала нужно создать к нему соединение:

1. Кликните на кнопку Диспетчер открытого источника данных  Open Data Source Manager.
2. Кликните слева на вкладку Геопакет  *GeoPackage*.
3. Кликните на кнопку Новое *New* и перейдите к файлу `training_data.gpkg` в папке `exercise_data` которую вы загрузили ранее.
4. Выберите файл и нажмите *Open*. Теперь путь файла добавлен к списку связей Geopackage, он появляется в раскрывающемся меню.

Теперь вы готовы добавлять любой слой из GeoPackage в QGIS.

1. Кликните по кнопке Соединить *Connect*. В центральной части окна вы должны увидеть список всех слоев, хранящихся в файле GeoPackage.
2. Выберите слой Дороги *roads* layer и кликните на кнопку *Add* button.



Слой *roads* добавлен на панель Слои *Layers* с функциями, отображаемыми на полотне карты.

3. Кликните *Close*.

Поздравляем! Вы загрузили первый слой из GeoPackage.

2.2.4 |Идем дальше|: Загружаем векторные данные из Базы данных SpatiaLite через браузер

QGIS предоставляет доступ ко многим другим форматам баз данных. Как и GeoPackage, формат базы данных SpatiaLite является расширением библиотеки SQLite. И добавление слоя из поставщика SpatiaLite следует тем же правилам, что описаны выше: Создать соединение –> Запустить его –> Добавить слой(и), т.е. Create the connection –> Enable it –> Add the layer(s).

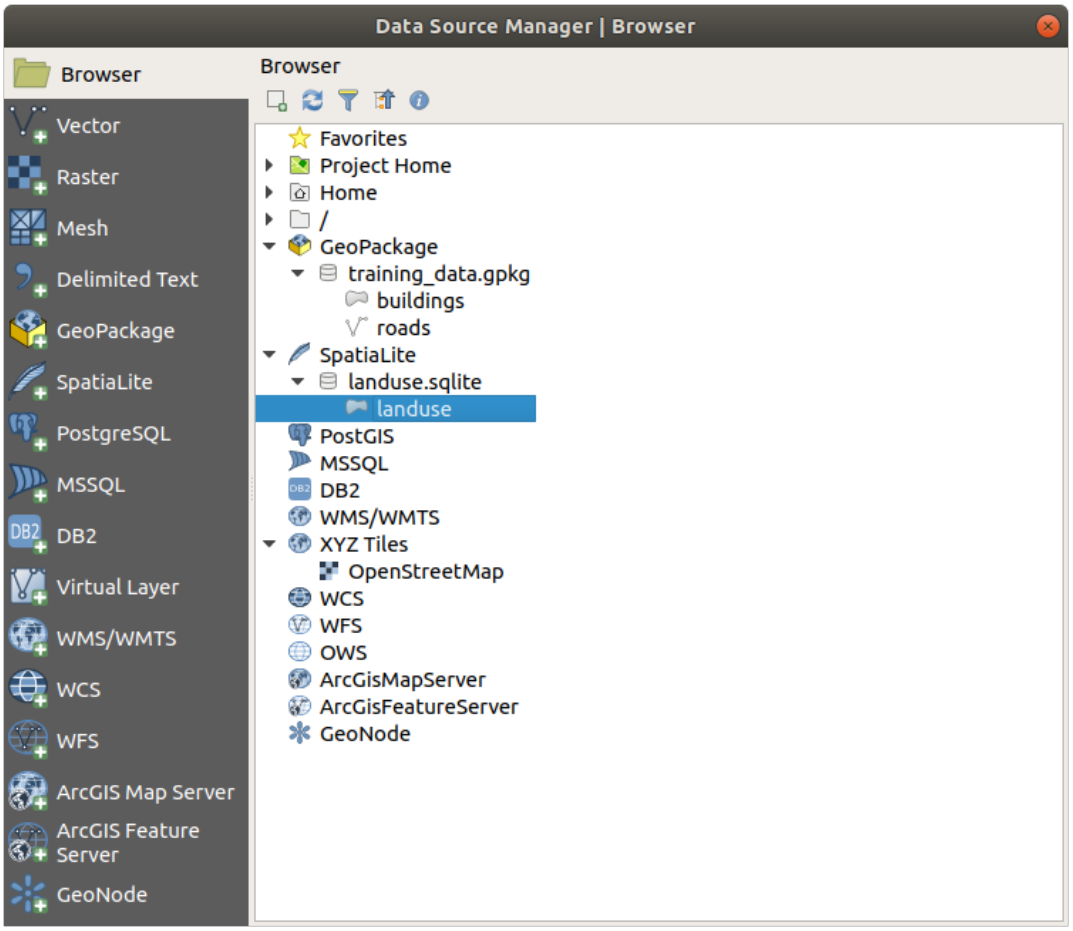
Это пока один из способов добавления данных SpatiaLite на вашу карту, давайте рассмотрим еще один эффективный способ добавления данных: Браузер *Browser*.

1. Кликните на иконку , чтобы открыть окно Диспетчера источника данных *Data Source Manager*.
2. Кликните на вкладку *Browser*.
3. В этой вкладке вы можете увидеть все жесткие диски, подключенные к вашему компьютеру, а также записи для большинства вкладок слева. Это обеспечивает быстрый доступ к подключенным базам данных или папкам.

   Например, кликните на раскрывающуюся иконку рядом со элементом Геопакет *GeoPackage*. Вы увидите файл `training-data.gpkg`, который мы предварительно соединили (а также его слои, если он расширяется).
4. Правый клик по элементу *SpatiaLite* еи выберите Новое соединение *New Connection*....
5. Перейдите к папке `exercise_data` , выберите файл Землепользование `landuse.sqlite` и кликните *Open*.

Обратите внимание, что элемент *landuse.sqlite* добавлен в *SpatiaLite*.

6. Разверните элемент *landuse.sqlite*.
7. Кликните дважды на слой *landuse* или выберите и перетащите его на полотно карты. Новый слой добавлен на панель *Layers* и его функции отображены на полотне карты.



Совет: Запустить панель *Browser* в Вид-Панели *View* ► *Panels* ► и используйте ее для добавления ваших данных. Это удобный ярлык для вкладки *Data Source Manager* ► *Browser* с той же функциональностью.

Примечание: Не забудьте часто сохранять проект! Файл проекта не содержит данных, но помнит, какие слои загружены на карту.

2.2.5 |Попробуйте сами|, загрузите больше векторных данных

Загрузите следующий набор данных из папки `exercise_data` на карту при помощи любого метода, описанного выше:

- Здания *buildings*
- Вода *water*

Answer

Your map should have seven layers:

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

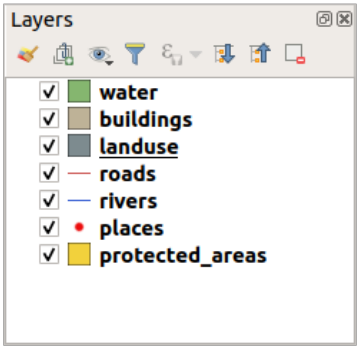
2.2.6 Идем дальше: Меняем порядок слоев

Слои в вашем списке Слоев отображаются на карте в определенном порядке. Слой внизу списка рисуется первым, а слой вверху рисуется последним. Изменяя порядок, в котором они отображаются в списке, вы можете изменить порядок, в котором они отображаются.

Примечание: Вы можете изменить это поведение, установив флажок Управление предложенным порядком *Control rendering order* под панелью Порядок слоев *Layer Order*. Однако мы пока не будем обсуждать эту функцию.

Порядок, в котором слои были загружены на карту, на данном этапе, вероятно, не логичен. Возможно, слой дороги полностью скрыт, потому что другие слои находятся поверх него.

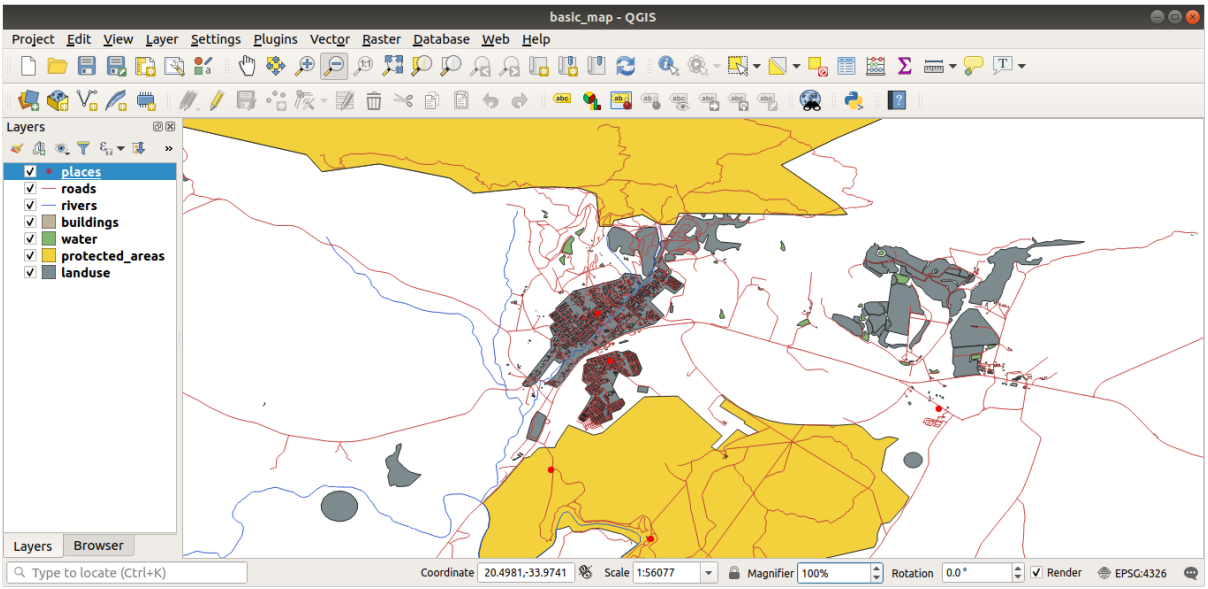
Например, такой порядок слоев…



… приведет к тому, что дороги и места будут скрыты, поскольку они находятся под полигонами слоя землепользования.

Чтобы решить данную проблему:

1. Кликните и перетащите слой из списка Слоев.
2. Измените их порядок, чтобы они выглядели следующим образом:



Вы увидите, что карта теперь визуально более понятна: дороги и здания появляются над территориями землепользования.

2.2.7 В заключении

Теперь вы добавили все необходимые слои из нескольких разных источников и создали базовую карту!

2.2.8 Что дальше?

Теперь вы знакомы с основными функциями кнопки Управление открытым источником данных *Open Data Source Manager*, но как насчет всех остальных? Как работает этот интерфейс? Прежде чем мы продолжим, давайте взглянем на некоторые основы взаимодействия с интерфейсом QGIS. Это тема следующего занятия.

2.3 Занятие: Навигация по полотну карты

В этом разделе главное внимание будет уделено основным инструментам навигации QGIS, используемым для навигации по полотну карты. Эти инструменты позволят вам визуально изучить слои в разных масштабах.

Цель данного занятия: узнать, как использовать инструменты Панорамирование и Масштабирование / Pan and Zoom в QGIS, и узнать о масштабе карты.

2.3.1 Идем дальше: Базовые инструменты навигации

Прежде чем научиться навигации по полотну карты, давайте добавим несколько слоев, которые мы сможем изучить в ходе обучения.

1. Откройте новый пустой проект и, при помощи шагов, описанных в разделе Создать карту *Create a Map*, загрузите в проект ранее просмотренные слои `protected_areas`, `roads` and `buildings` layers to the project. Результат должен выглядеть примерно так, как показано на Рис. 2.2 ниже (цвета не имеют значения):

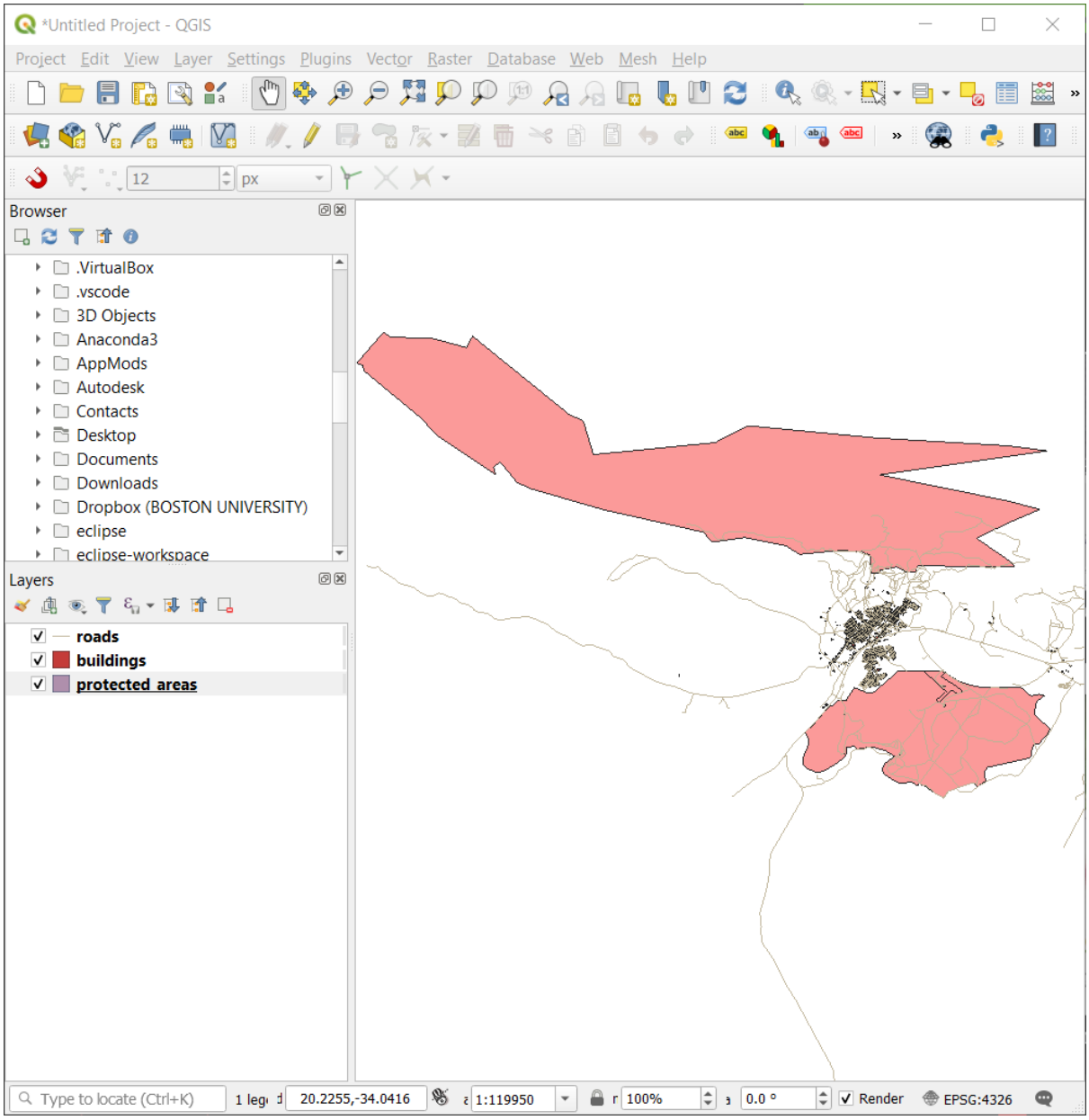


Рис. 2.2: Добавленные охраняемые территории, дороги и здания.

Давайте сначала узнаем, как применять инструмент Панорамирование / PanTool.

1. Убедитесь, что на *Панели инструментов навигации по карте*, активирована кнопка Pan.
2. Переместите мышь в центр полотна карты.

3. Кликните левой кнопкой мыши и, удерживая ее, перетащите мышь в любом направлении для панорамирования карты.

Затем давайте увеличим масштаб и подробнее рассмотрим импортированные нами слои.

1. Кликните кнопку Увеличить  на *Панели инструментов навигации по карте*.
2. Переместите курсор мыши примерно в левую верхнюю область, где самая высокая плотность зданий и дорог.
3. Щелкните левой кнопкой мыши и удерживайте.
4. Затем перетащите мышь, что создаст прямоугольник и захватит плотно заставленную местность зданий и дорог (Рис. 2.3).

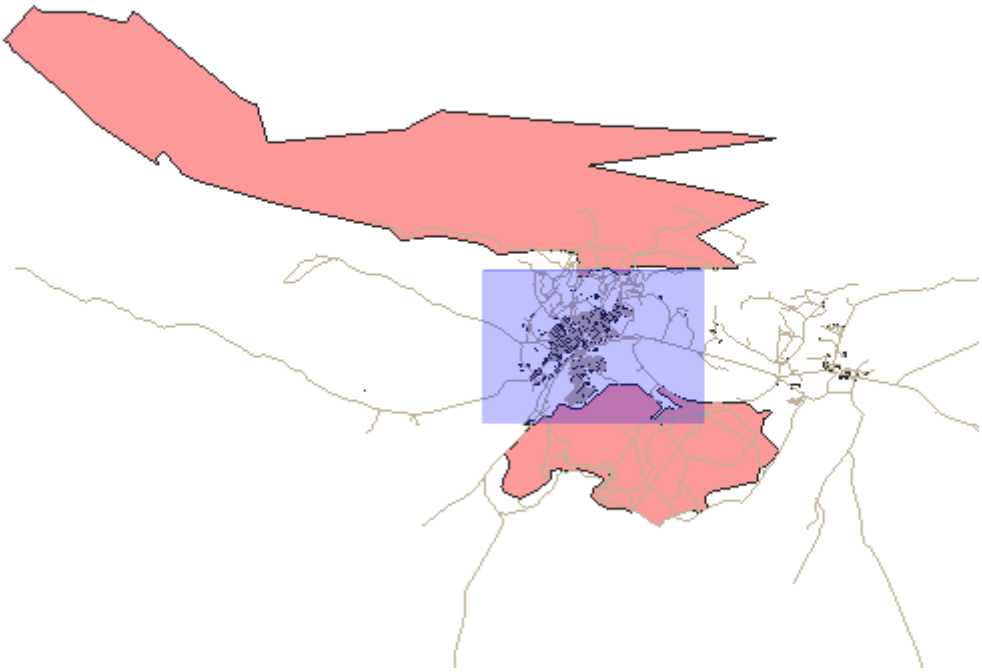
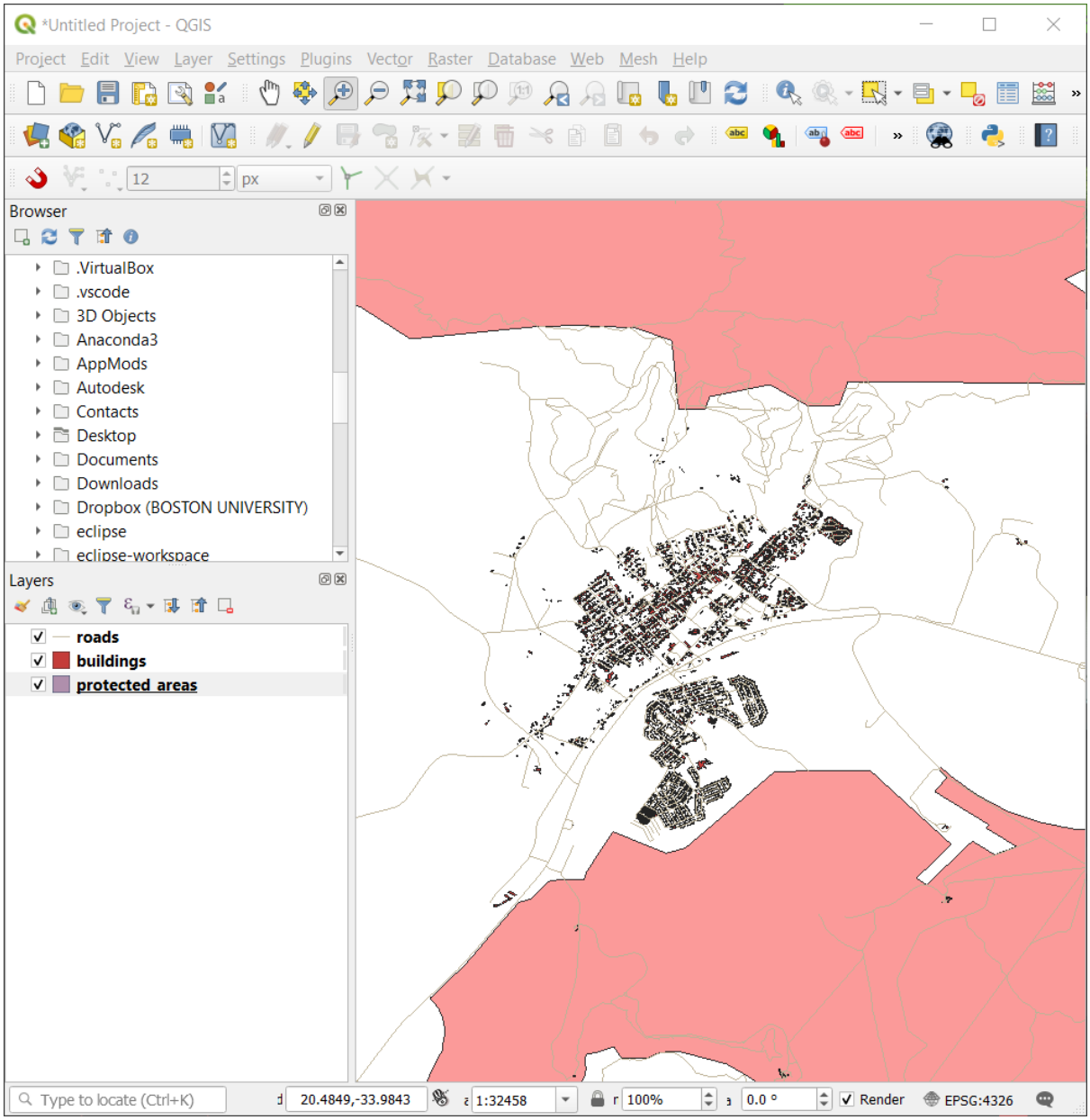


Рис. 2.3: Zoom in

5. Отпустите левую кнопку мыши. Это приблизит карту и включит местность, которую вы выбрали своим прямоугольником.



6. Чтобы уменьшить масштаб, выберите кнопку Отдалить Zoom Out и выполните то же действие, которое вы выполняли для приближения карты.

В то время как вы прокручиваете, приближаете или отдаляете карту, QGIS сохраняет все просмотры в истории. Что позволяет вернуться к предыдущему просмотру.

1. На *Панели инструментов навигации по карте*, кликните по кнопке Последний масштаб Zoom Last (то что вы приближали или отдаляли в последний раз), чтобы перейти к предыдущему виду.

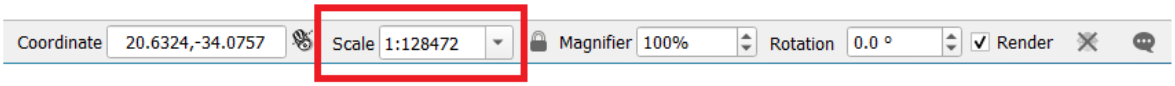
2. Кликните по кнопке Масштабировать далее Zoom Next, чтобы продолжить движение по истории вашего зума.

Иногда после изучения данных нам нужно сбросить просмотр до уровня всех слоев. Вместо того, чтобы пытаться многократно использовать инструмент Отдалить, QGIS предоставляет нам кнопку для выполнения данного действия за нас.

1. Click on the Zoom Full Extent button.

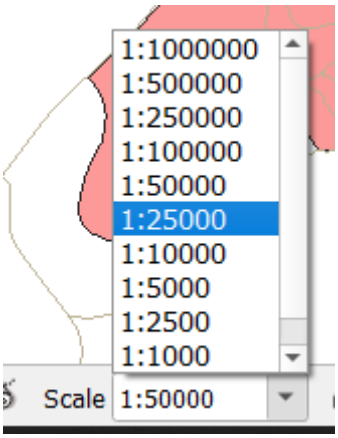
При приближении и отдалении, обратите внимание, что значение Масштаб *Scale* в строке состояния изменя-

ется. Значение Масштаб *Scale* представляет масштаб карты. В общем, число справа от двоеточия *:* представляет, во сколько раз объект, который вы видите на полотну карты меньше фактического объекта в реальности.



Вы также можете использовать данное поле для установки масштаба карты вручную.

1. В строке состояния, кликните на текстовое поле *Scale*.
2. Введите `50000` и нажмите `Enter`. Это перерисует функции на полотне карты, чтобы отразить масштаб, который вы ввели.
3. В качестве альтернативы, можно кликнуть на стрелку параметров в поле *Scale*, чтобы увидеть предустановленные масштабы карты.



4. Выберите *1:5000*. Это также обновит масштаб карты на полотне карты.

Теперь вы знаете основы навигации по полотну карты. Ознакомьтесь с Руководством пользователя по Zooming and Panning (масштабирование и панорамирование), чтобы узнать об альтернативных способах навигации по полотну карты.

2.3.2 В заключении

Знание навигации по полотну карты важно, поскольку оно позволяет изучать и визуально проверять слои. Такое действие можно выполнить для проведения первоначального исследования данных или для проверки результатов пространственного анализа.

2.4 Занятие: Символика

Слой символа - это его внешний вид на карте. Основное преимущество GIS перед другими способами представления данных с пространственными аспектами заключается в том, что с помощью GIS у вас есть динамическое визуальное представление данных, с которыми вы работаете.

Поэтому внешний вид карты (который зависит от символики отдельных слоев) очень важен. Конечный пользователь создаваемых вами карт должен иметь возможность легко увидеть, что представляет собой карта. Не менее важно, чтобы вы могли изучать данные во время работы с ними, а хорошая символика очень помогает.

Другими словами, иметь правильную символику - это не роскошь и не просто хорошо, что она есть. Фактически, для вас очень важно правильно использовать GIS и создавать карты и информацию, которые люди смогут использовать.

Цель данного занятия: иметь возможность создавать любую символику для любого векторного слоя.

2.4.1 Идем дальше: Меняем цвета

Откройте Свойства слоя *Layer Properties*, чтобы изменить символику слоя. Начнем с изменения цвета слоя Землепользование *landuse*.

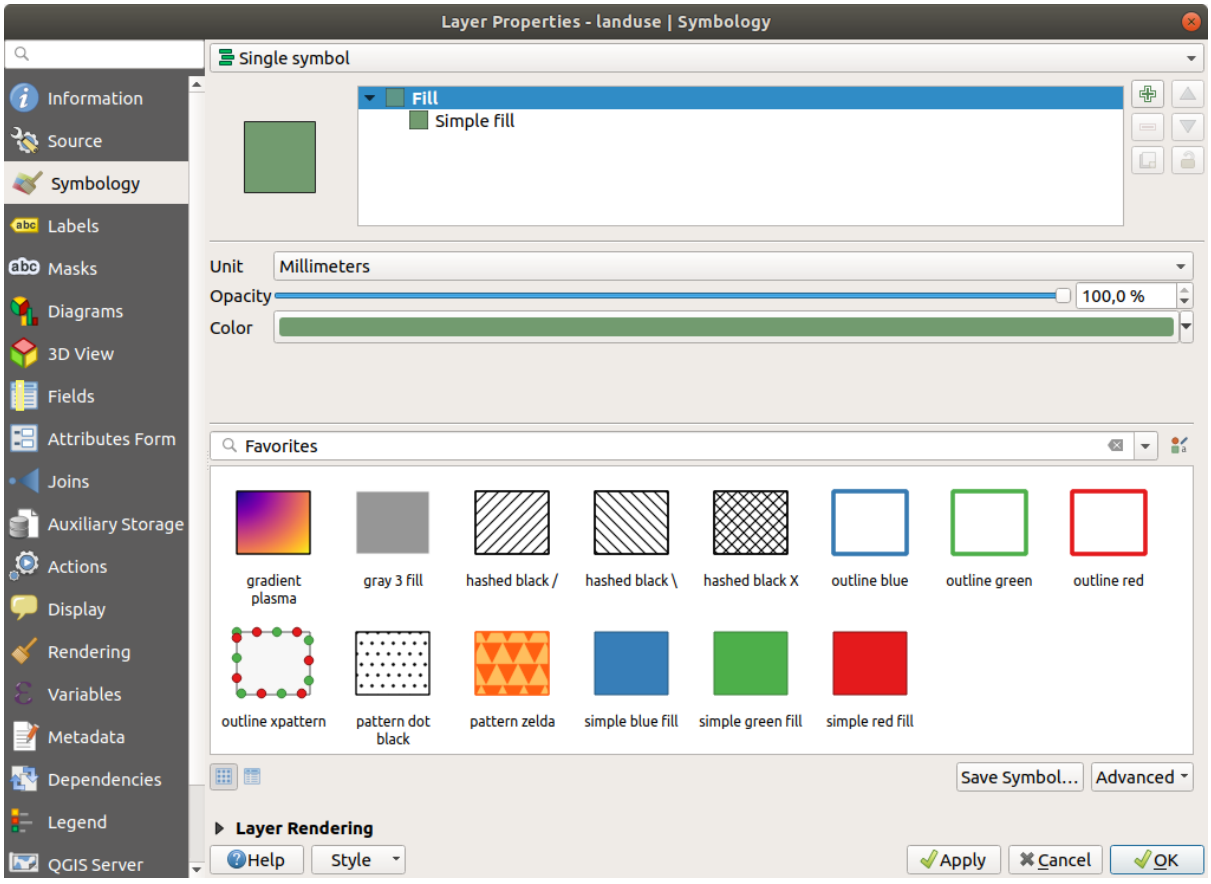
1. Кликните правой кнопкой мыши на слой *landuse* в списке слоев.
2. В открывшемся меню выберите пункт Свойства… / *Properties…*.

Примечание: Вы также можете получить доступ к свойствам слоя по умолчанию, дважды кликнув на слой в списке Слоев.

Совет: Кнопка в верхней части панели *Layers* открывает панель Стили слоя *Layer Styling*. Вы можете использовать эту панель для изменения некоторых свойств слоя: по умолчанию, изменения вступают в силу немедленно!

---

3. Выберите, вкладку Символика *Symbology* tab: в окошке Свойства слоя *Layer Properties*.



4. Кликните на кнопку выбора цвета рядом с ярлыком *Color*. Появится стандартное диалоговое окно цвета.
5. Выберите серый цвет и нажмите *OK*.

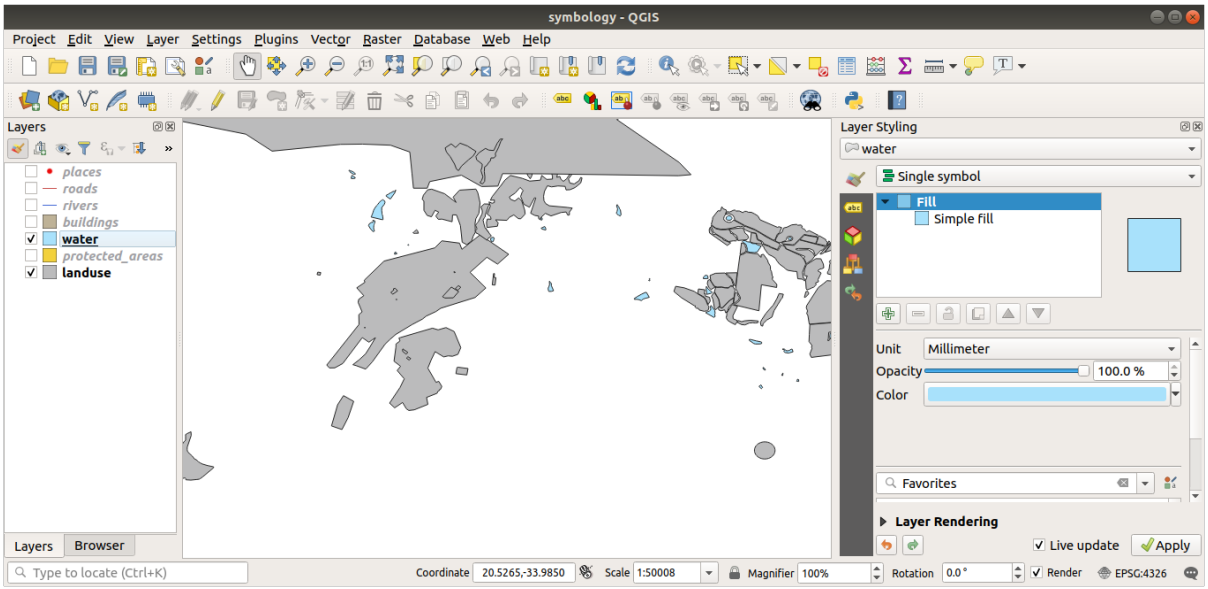
6. Еще раз нажмите *OK* в окне Свойства слоя *Layer Properties*, и вы увидите, что изменение цвета применяется к слою.

2.4.2 Попробуйте сами

Измените цвет слоя *water* на светло голубой. Попробуйте использовать панель Стиль слоя *Layer Styling* вместо меню Свойства слоя *Layer Properties*.

Solution

- Verify that the colors are changing as you expect them to change.
- It is enough to select the *water* layer in the legend and then click on the  Open the Layer Styling panel button. Change the color to one that fits the water layer.



If you want to work on only one layer at a time and don't want the other layers to distract you, you can hide a layer by clicking in the checkbox next to its name in the layers list. If the box is blank, then the layer is hidden.

2.4.3 Идем дальше: Меняем структуру символа

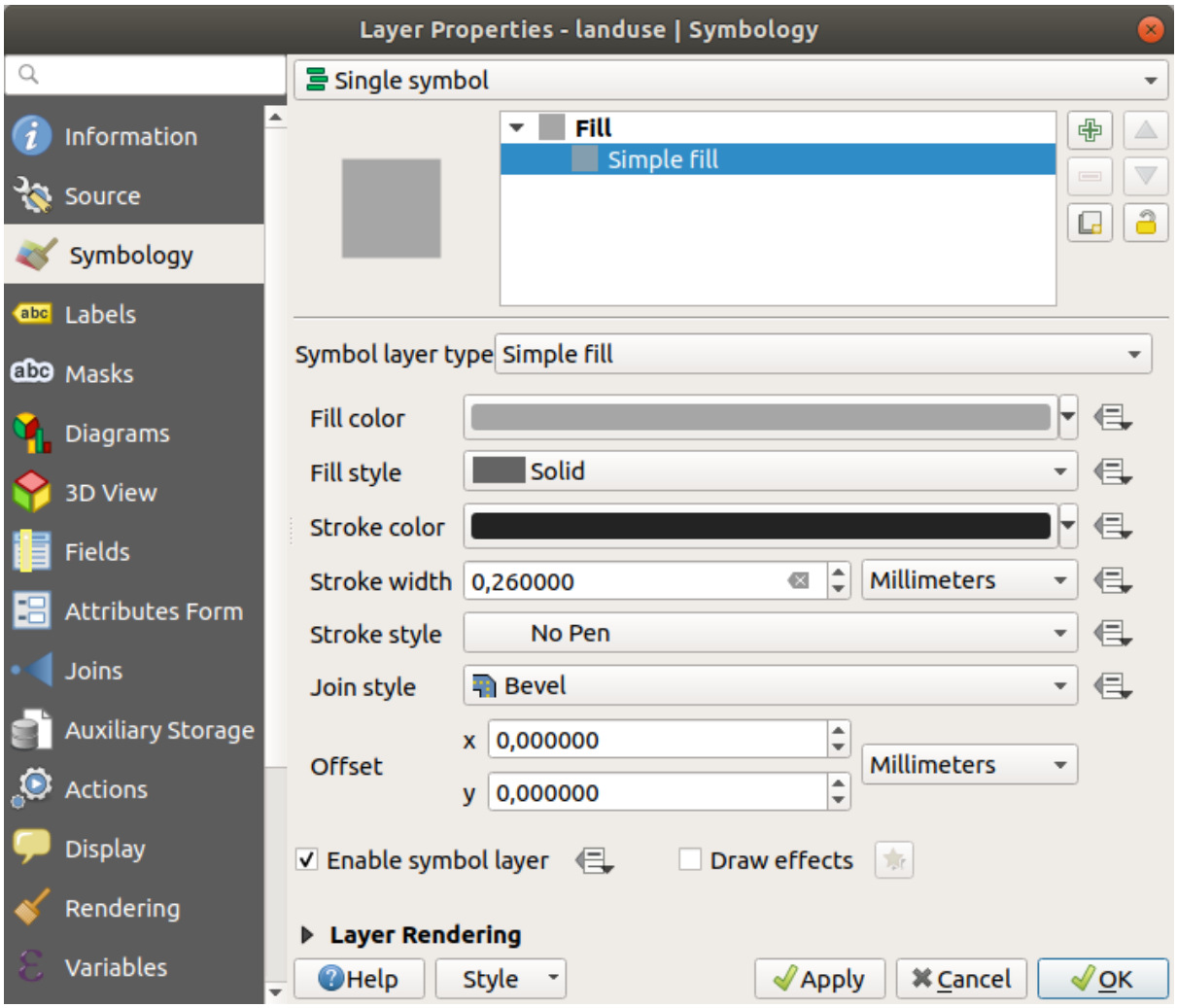
Пока что все неплохо, но слой символа - это не только его цвет. Далее мы хотим удалить линии между различными местностями землепользования, чтобы сделать карту менее загроможденной визуально.

1. Откройте окно *Layer Properties* для слоя *landuse*.

   Под вкладкой  *Symbology*, вы увидите похожее диалоговое окно, которое вы видели ранее. Однако, на этот раз вы делаете больше, чем просто быстро меняете цвет.

2. В слоях символа откройте раскрывающийся список Заливка *Fill* и выберите параметр Простая заливка *Simple fill*.
3. Кликните раскрывающийся список Стиль обводки *Stroke style*. На данный момент он должен показывать короткую линию и слова Сплошная линия *Solid Line*.

4. Измените ее на Без пера *No Pen*.



5. Кликните *OK*.

Теперь слой *landuse* будет без линий между местностями.

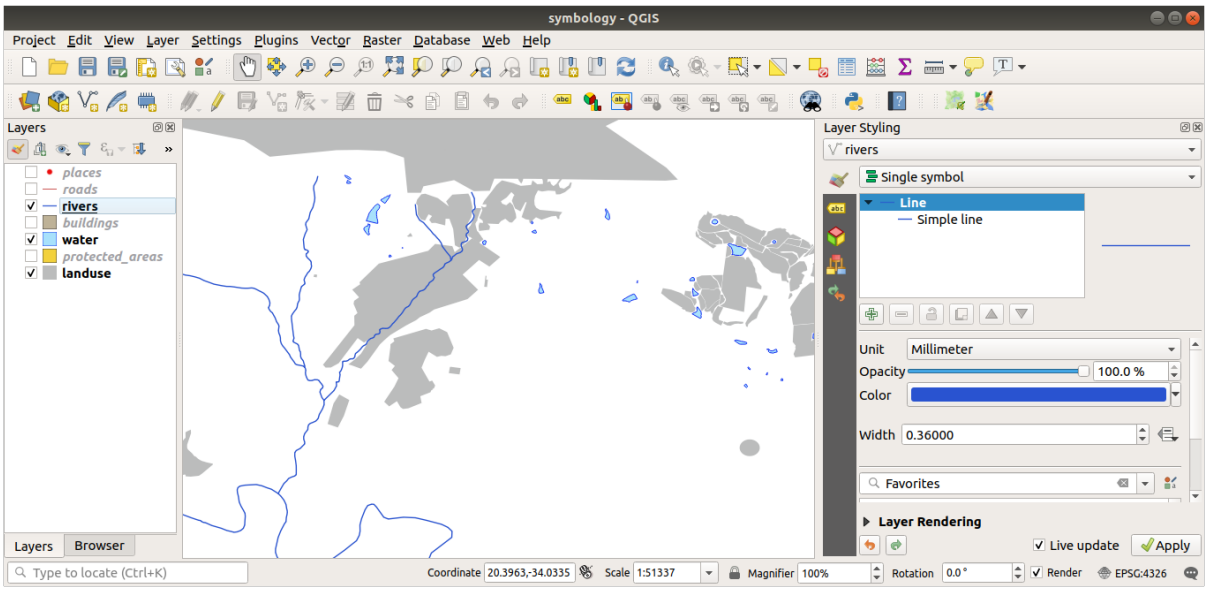
2.4.4 Попробуйте сами

- Снова измените слой символа *water*, чтобы она обрела более темно-синий контур.
- Измените слой символа реки *rivers*, чтобы он отображал водные пути более подробно.

Помните: вы можете использовать кнопку Открыть панель стилей слоя  Open the Layer Styling panel и увидеть все изменения сразу. Данная панель также позволяет вам отменять отдельные изменения при использовании символизирования слоя.

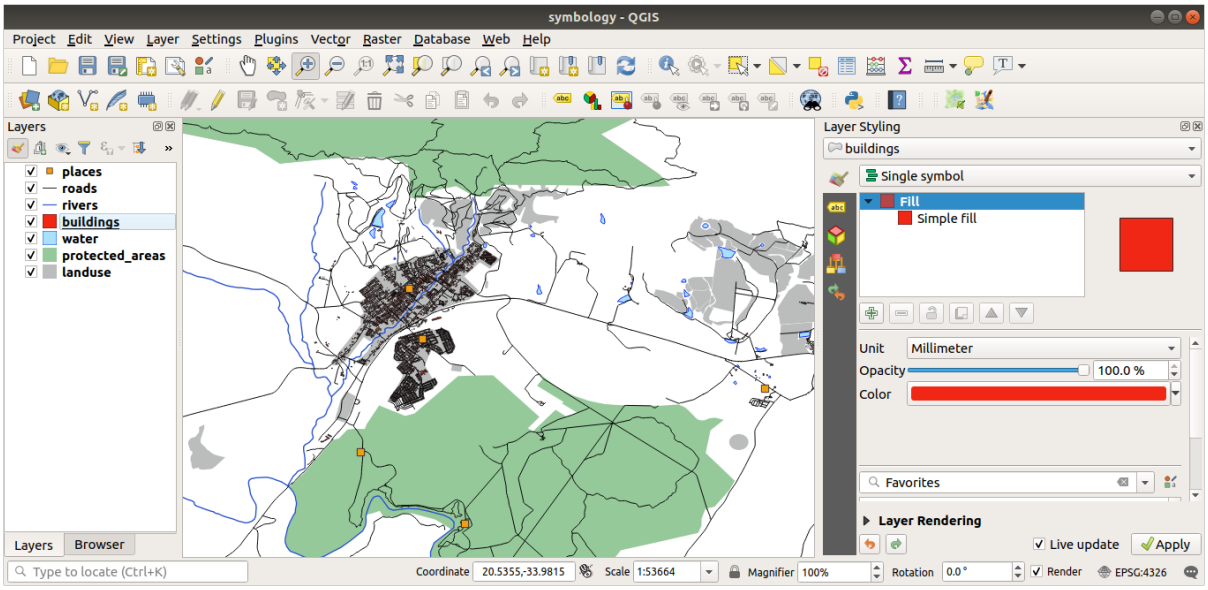
Answer

Your map should now look like this:



If you are a Beginner-level user, you may stop here.

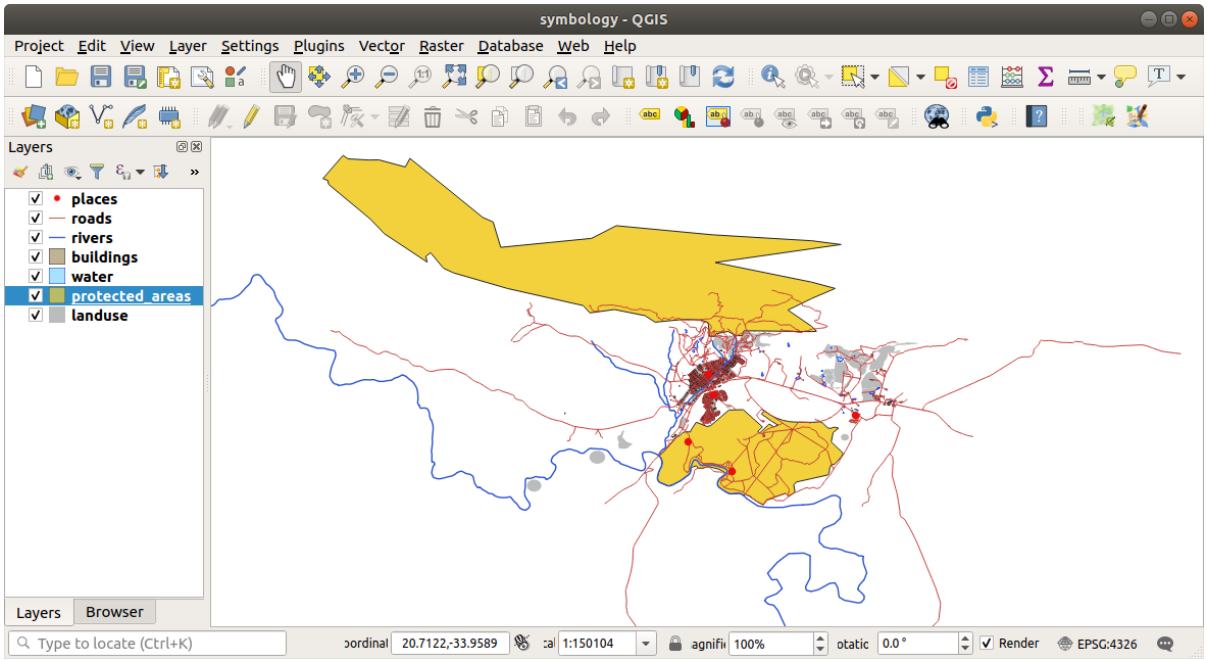
- Use the method above to change the colors and styles for all the remaining layers.
- Try using natural colors for the objects. For example, a road should not be red or blue, but can be gray or black.
- Also feel free to experiment with different *Fill style* and *Stroke style* settings for the polygons.



2.4.5 Идем дальше: Видимость на основе масштаба

Вы обнаружите, что иногда слой не будет подходить для данного масштаба. Например, набор данных по всем континентам может иметь скудную детализацию и быть не очень точным на уровне улиц. Когда такое произойдет, вы захотите иметь возможность скрыть набор данных несоответствующих масштабов.

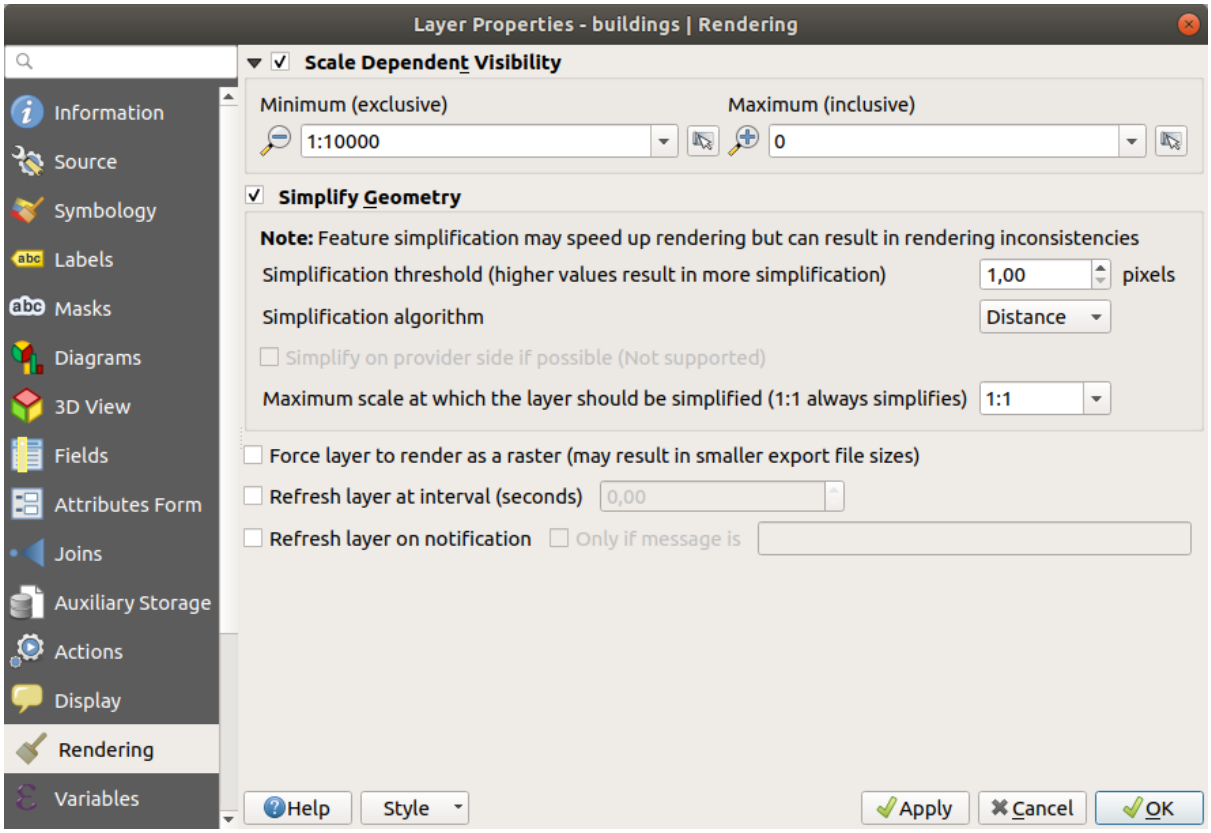
В нашем случае мы можем решить скрыть здания от просмотра в небольших масштабах. Например, эта карта…



… не очень удобная. Здания трудно различить в таком масштабе.

Чтобы запустить визуализацию на основе масштаба:

1. Откройте диалоговое окно Свойства слоя *Layer Properties* для слоя Здания *buildings*.
2. Активируйте вкладку *Rendering*.
3. Запустите визуализацию на основе масштаба, кликнув на флажок под названием Видимость в зависимости от масштаба *Scale dependent visibility*:
4. Измените минимальное значение *Minimum* на `1:10000`.



5. Кликните *OK*.

Проверьте данный эффект путем отдаления и приближения карты, отмечая, когда слой зданий исчезает и появляется снова.

Примечание: Вы можете использовать колёсико мыши для постепенного приближения, или использовать инструменты масштабирования, чтобы приблизить окно:

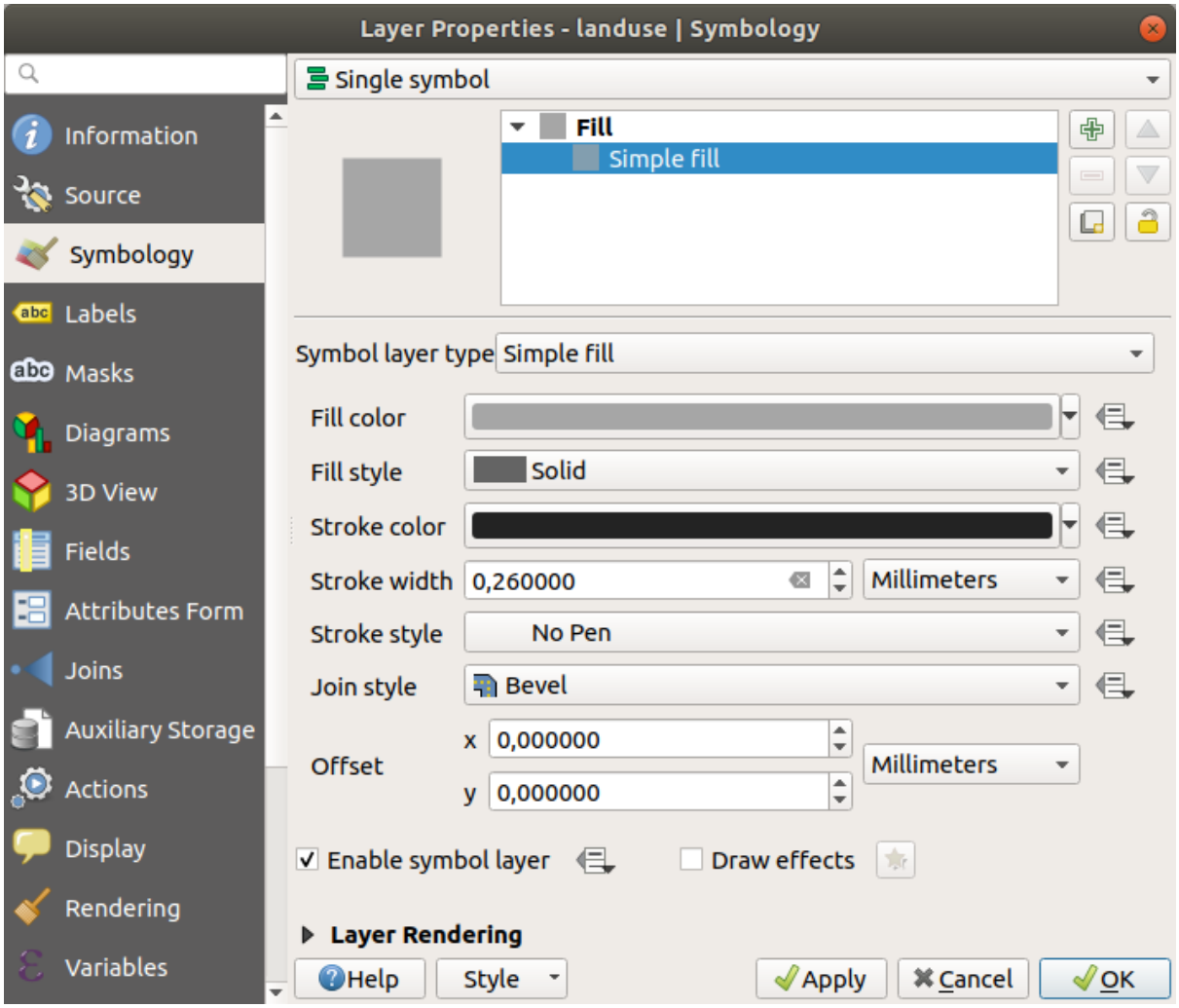


2.4.6  Идем дальше: Добавляем слои символа

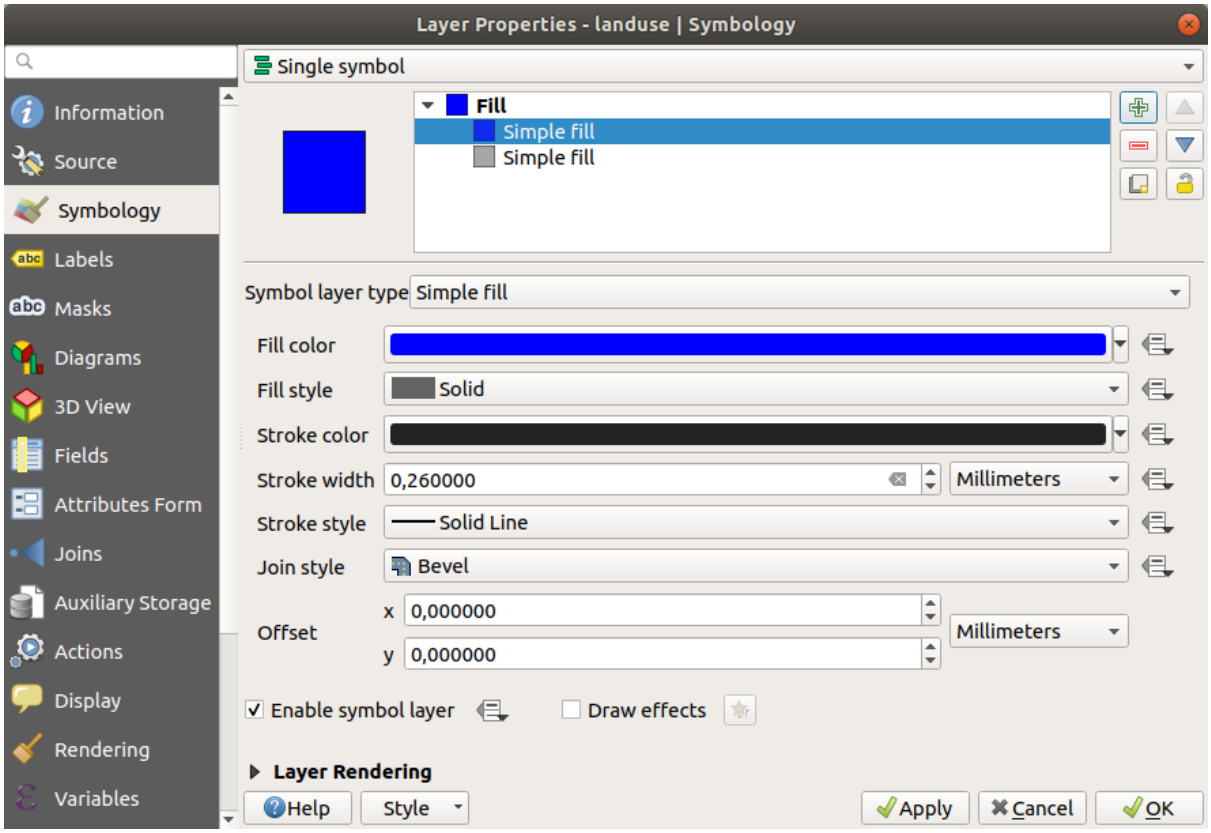
Теперь, когда вы знаете, как изменить простую символику слоев, следующим шагом будет создание более сложной символики. QGIS позволяет делать это с помощью слоев символа.

1. Вернитесь на панель свойства слоя символа *landuse* (кликнув Простая заливка *Simple fill* на дереве слоев символа).

   В данном примере, текущий символ не имеет контура (т.е. применяется стиль границы Без пера *No Pen*).



2. Выберите *Fill* на дереве и кликните на кнопку Добавить слой символа  Add symbol layer. Диалоговое окно изменится с добавлением нового слоя символа, и будет выглядеть примерно так:



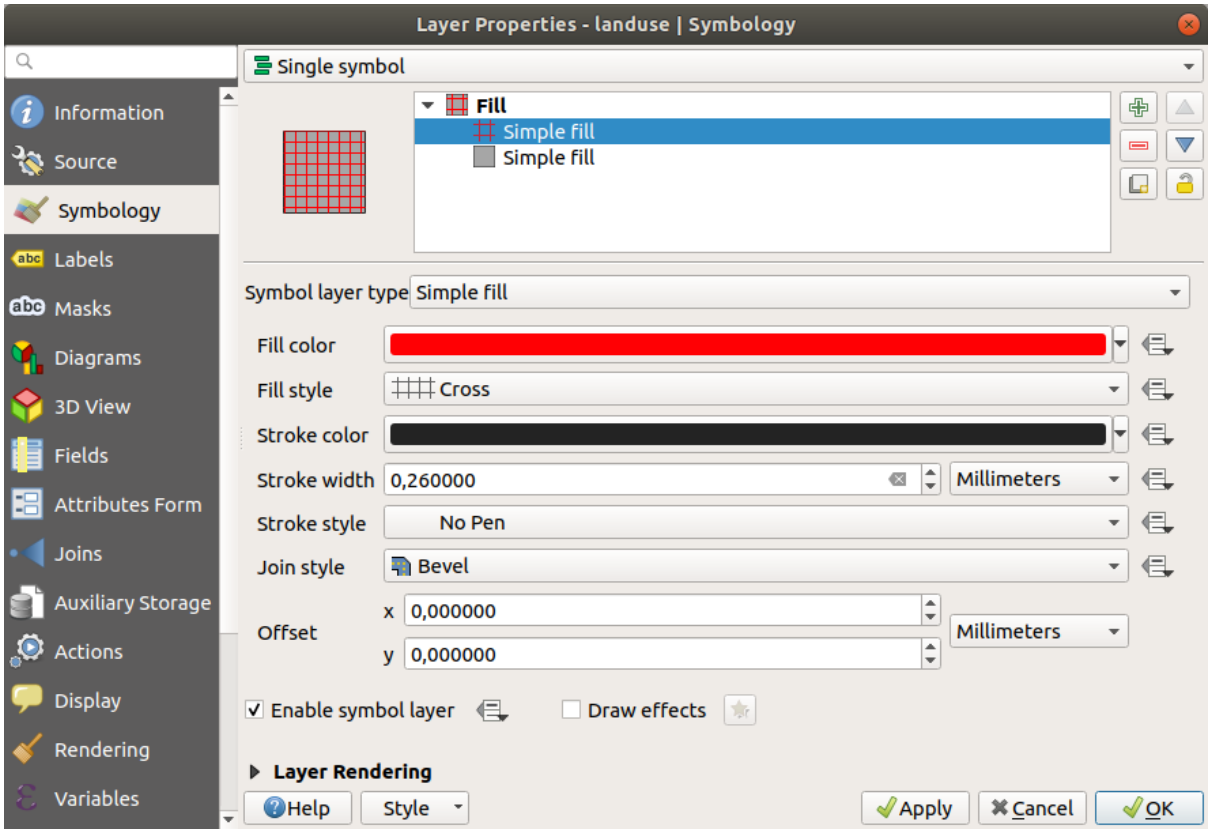
Например, оно может выглядеть несколько иначе по цвету, но вы все равно это измените.

Теперь есть второй слой символа. Поскольку здесь сплошной цвет, он, конечно же, полностью скроет предыдущий вид символа. Кроме того, у него есть стиль границы Сплошная линия *Solid Line*, который нам не нужен. Очевидно, что данный символ необходимо изменить.

Примечание: Важно не путать слой карты и слой символа. Слой карты - это вектор (или растр), загруженный на карту. Слой символа - это часть символа, используемого для представления слоя карты. В данном курсе обучения слой карты обычно называется просто слоем, но слой символа всегда будет называться слоем символа, чтобы избежать путаницы.

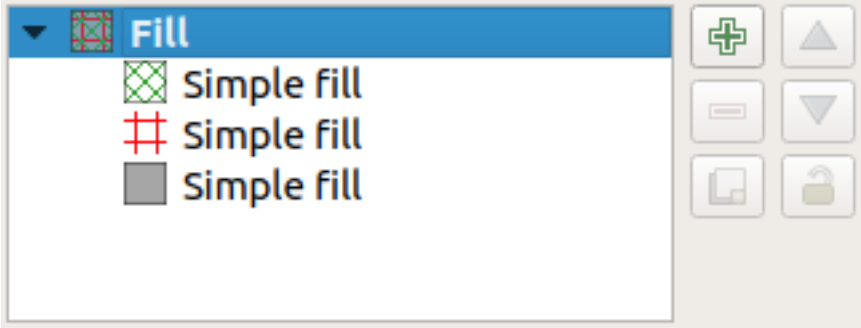
Выбрав новый слой символа Простая заливка *Simple Fill*:

1. Установите стиль границы *No Pen*, как и раньше.
2. Измените стиль заливки на любой, кроме Сплошная *Solid* или Без кисти *No brush*. Например:



3. Кликните *OK*.

Теперь вы можете видеть свои результаты и при необходимости поправлять их. Вы даже можете добавить несколько дополнительных слоев символов и таким образом создать своего рода текстуру для своего слоя.



Это весело! Но, вероятно, слишком много цветов, чтобы применять к реальной карте…

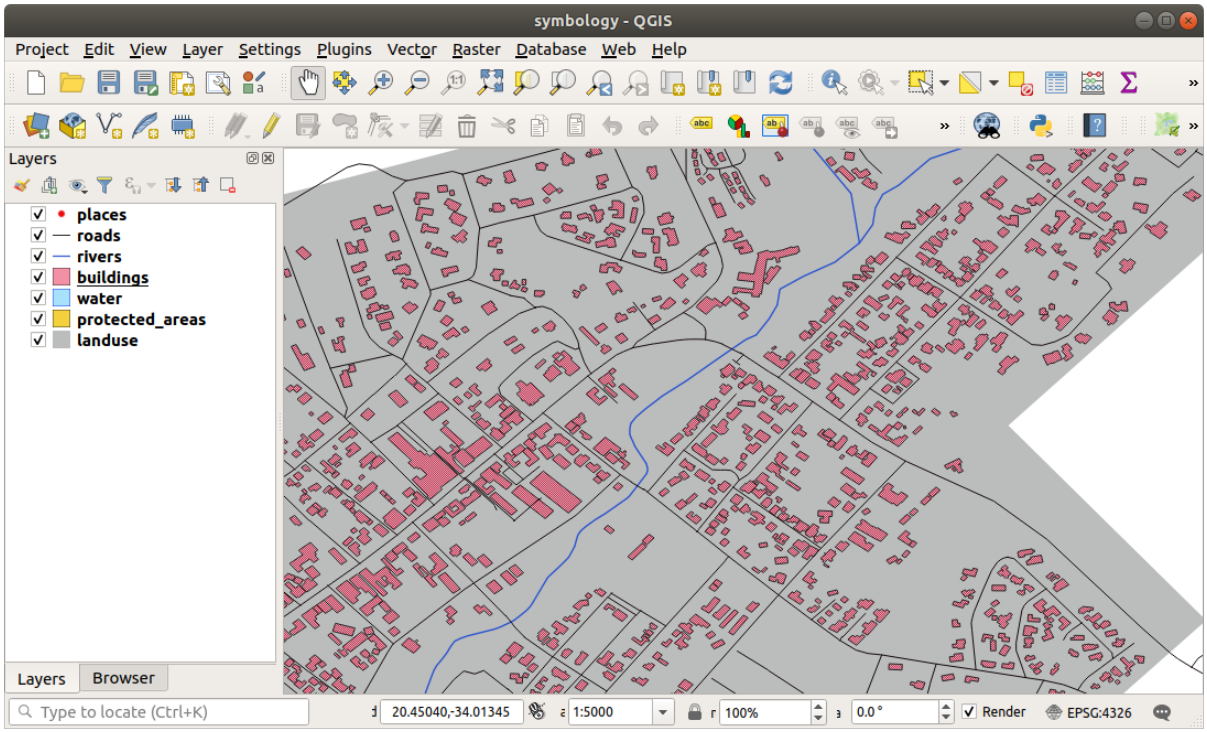
2.4.7 Попробуйте сами

Создайте простую, но не отвлекающую внимание, текстуру для слоя зданий, используя описанные выше методы и не забываем приближать, в случае необходимости.

Answer

Customize your *buildings* layer as you like, but remember that it has to be easy to tell different layers apart on the map.

Here's an example:



2.4.8  Идем дальше: Упорядочиваем уровни символа

В случае, когда визуализируются слои символов, они также отображаются в последовательности, аналогично тому, как визуализируются различные слои карты. Это означает, что в некоторых случаях наличие множества слоев символов в одном символе может привести к неожиданным результатам.

1. Задайте слою *roads* дополнительный слой символа (используя метод добавления слоев символа, показанный выше).
2. Задайте для базовой линии Ширину штриха *Stroke width* of `1.5` и черный цвет.
3. Задайте новому верхнему слою толщину `0.8` и белый цвет.

Вы заметите, следующее:

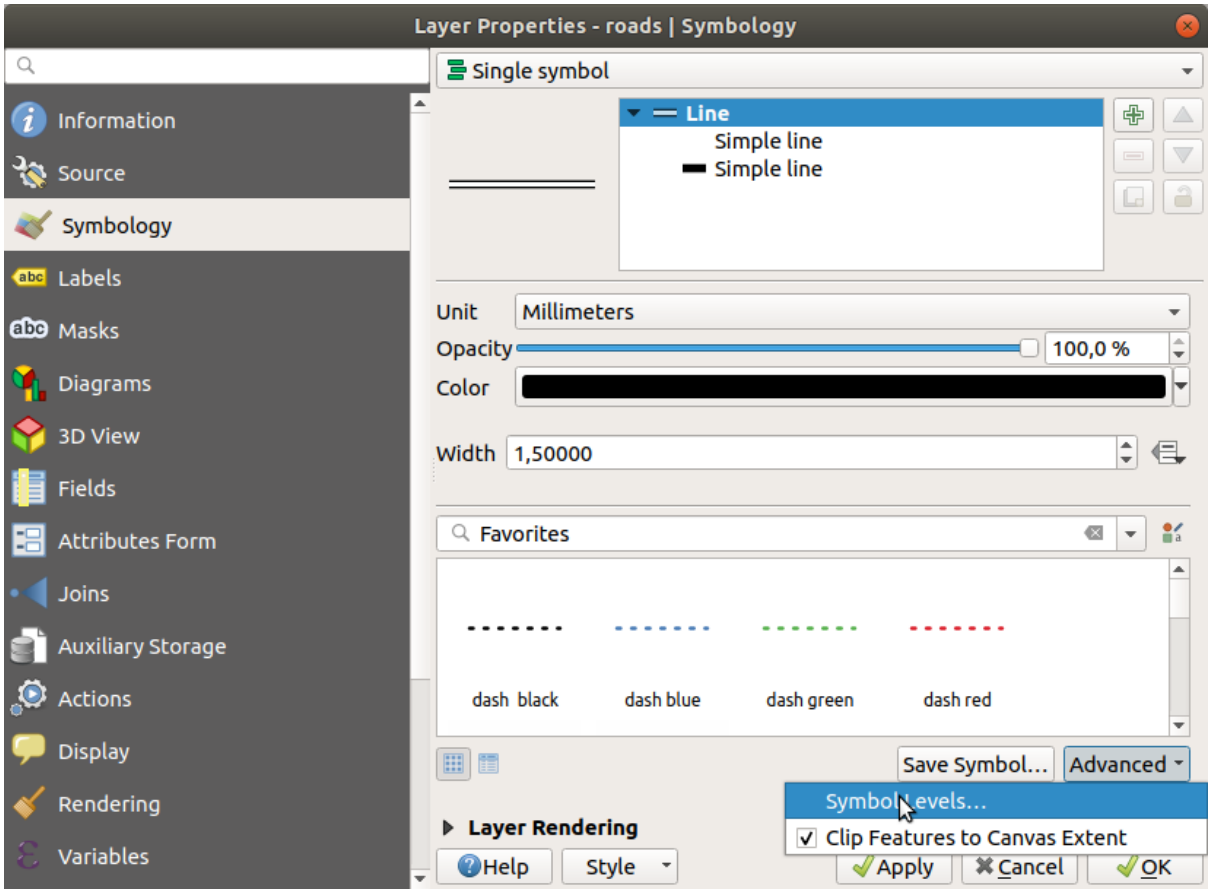


Что ж, дороги теперь имеют символику, похожую на улицу, но вы видите, что линии перекрывают друг друга на каждом перекрестке. Это совсем не то, что мы хотим!

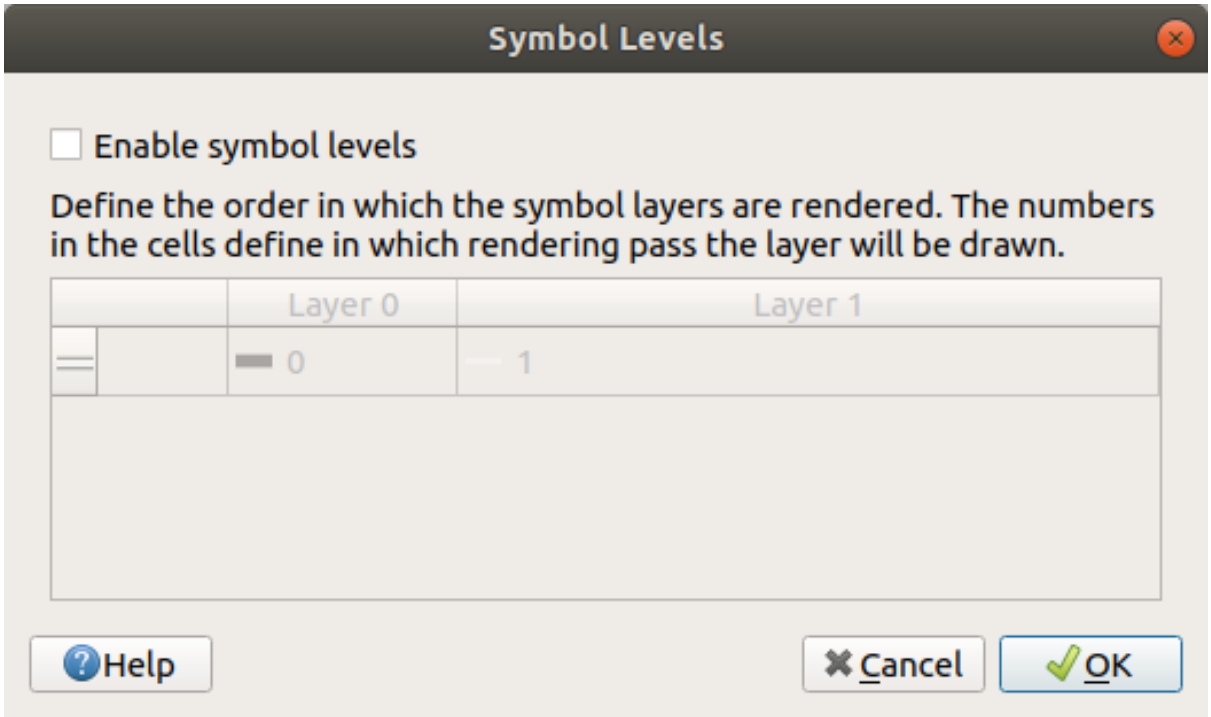
Чтобы подобного не происходило, вы можете отсортировать уровни символов и, таким образом, управлять порядком, в котором отображаются различные слои символов.

Чтобы изменить порядок слоев символов:

1. Выберите самый верхний слой Линия *Line* на дереве слоев символов.
2. Кликните *Advanced* ► *Symbol levels…* в правом нижнем углу окна.

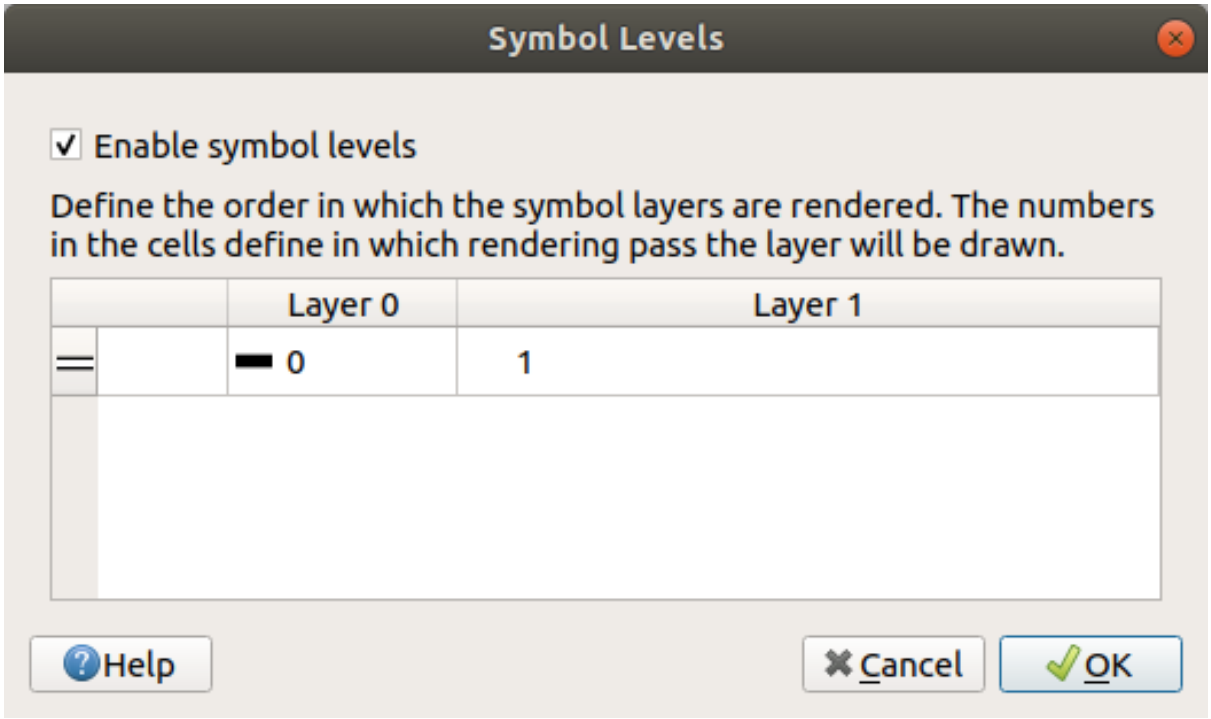


Это откроет диалоговое окно:



3. Проверьте Включены ли уровни символа *Enable symbol levels*. Затем вы можете установить порядок слоев каждого символа, введя соответствующий номер уровня. Ноль (0) - нижний слой.

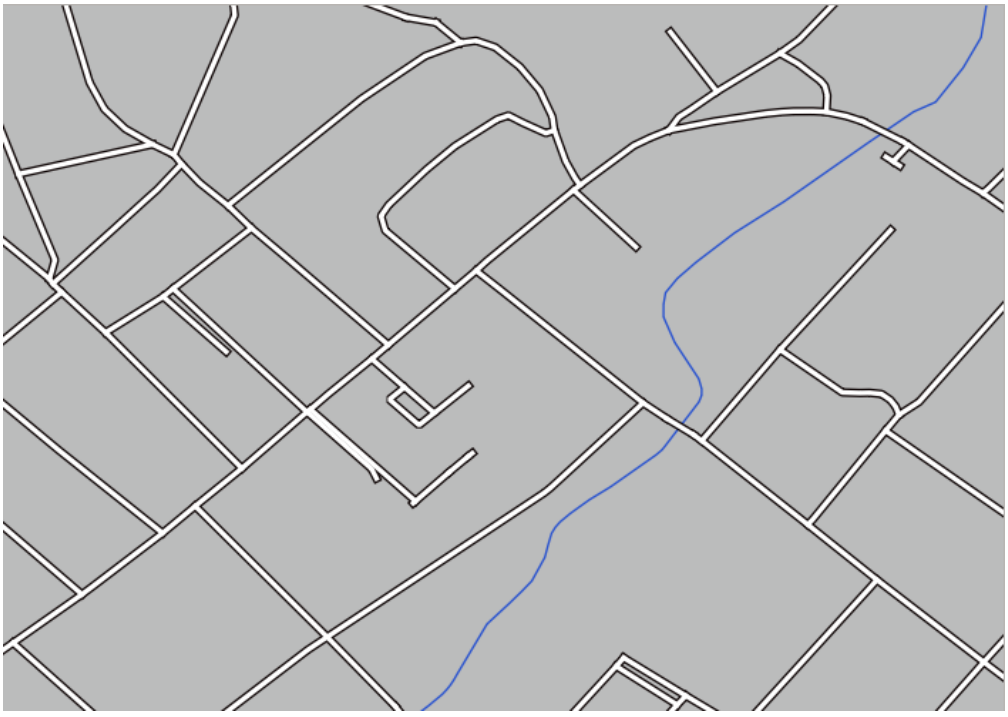
В нашем случае, мы просто хотим активировать следующую опцию:



Это отобразит белую линию над границами толстой черной линии:

4. Дважды кликните *OK*, чтобы вернуться к карте.

   Карта теперь будет выглядеть так:



Когда вы закончите, не забудьте сохранить сам символ, чтобы не потерять работу, если вы снова будете менять символ. Вы можете сохранить текущий стиль символа, нажав кнопку Сохранить стиль… *Save Style…* в нижней части диалогового окна Свойства слоя *Layer Properties*. Мы будем применять формат *QGIS QML Style File*.

Сохраните свой стиль в папке `solution/styles/better_roads.qml`. Вы можете загрузить ранее сохраненный стиль в любое время, кликнув на кнопку *Load Style…*. Прежде чем менять стиль, имейте в виду, что любой несохраненный стиль, который вы меняете, будет утерян.

2.4.9  Попробуйте сами

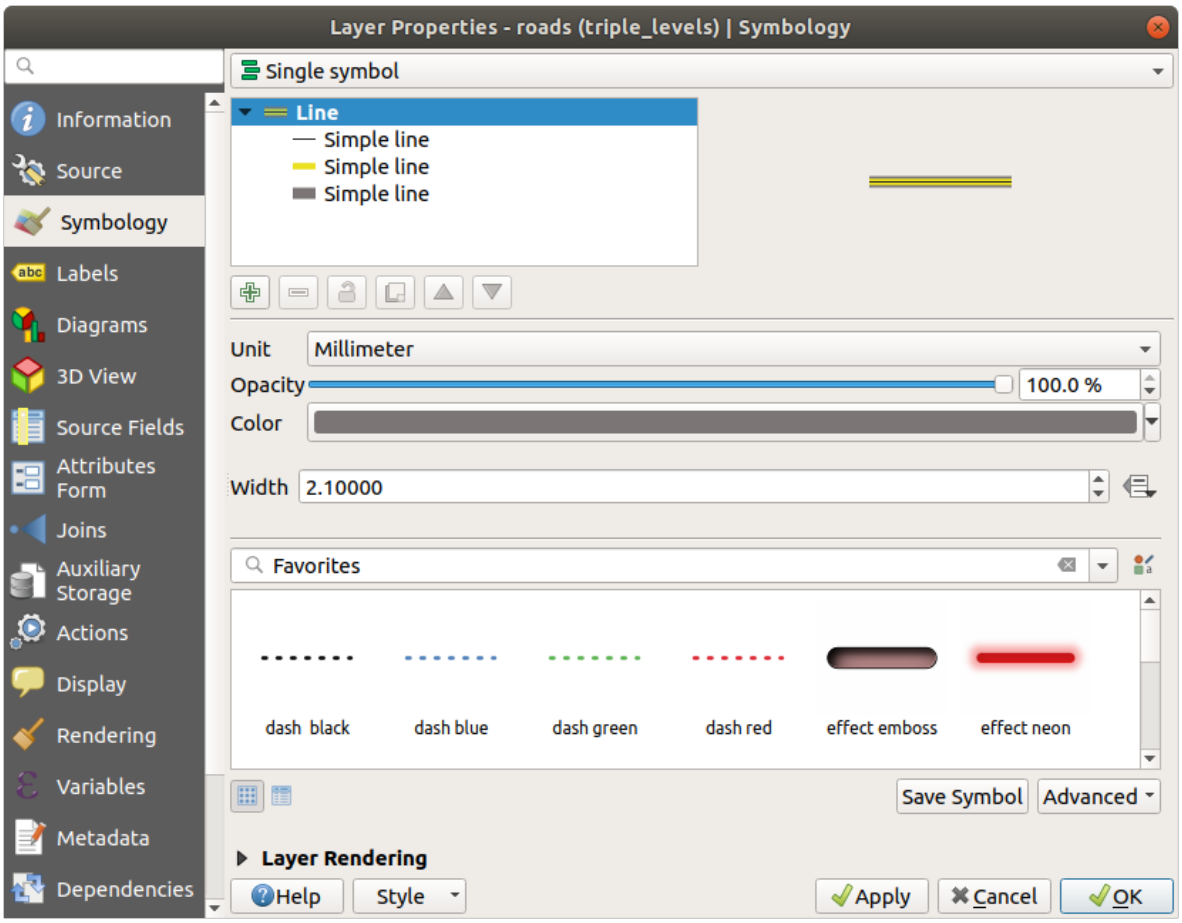
Снова измените внешний вид слоя *roads*.

Сделайте дороги узкими и желтыми, с тонким бледно-серым контуром и тонкой черной линией посередине. Помните, что вам может потребоваться изменить порядок слоев визуализации в диалоговом окне *Advanced* ► *Symbol levels…*.



Answer

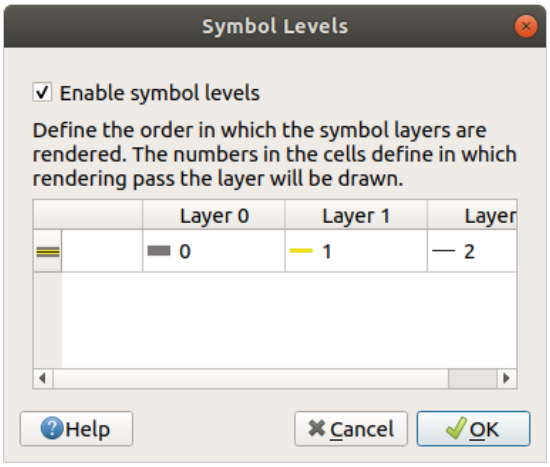
To make the required symbol, you need three symbol layers:



The lowest symbol layer is a broad, solid gray line. On top of it there is a slightly thinner solid yellow line and finally another thinner solid black line.

If your symbol layers resemble the above but you're not getting the result you want:

1. Check that your symbol levels look something like this:



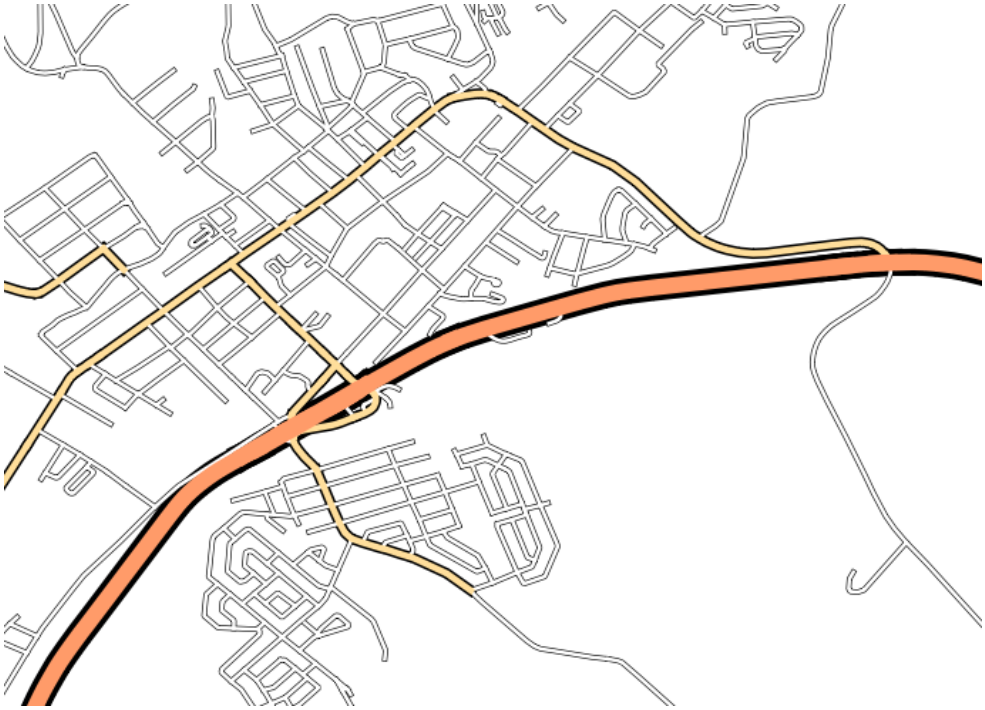
2. Now your map should look like this:



2.4.10 Попробуйте сами

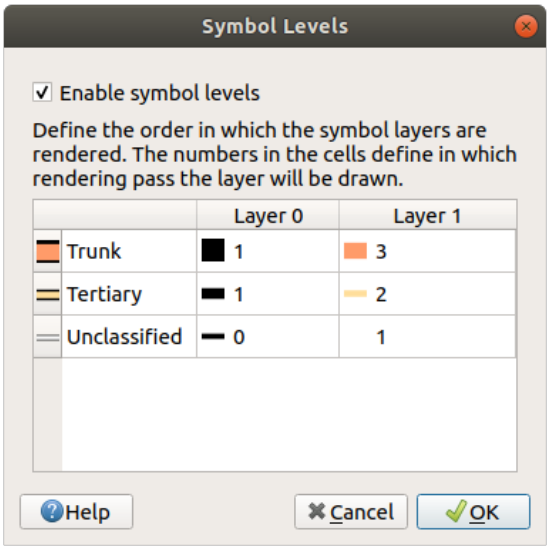
Уровни символов также работают для классифицированных слоев (т.е. слоев, содержащих несколько символов). Поскольку мы еще не рассмотрели классификацию, вы будете работать с некоторыми элементарными предварительно классифицированными данными.

1. Create a new project and add only the *roads* dataset.
2. Apply the style file `advanced_levels_demo.qml` provided in `exercise_data/styles` to the layer. This can be done through the *Style* ► *Load Style...* combobox at the bottom of the *Layer Properties* dialog.
3. Приблизите местность города СвеллендамSwellendam Swellendam.
4. Используя слои символов, убедитесь, что контуры слоев переходят друг в друга, как показано на изображении ниже:



Answer

1. Adjust your symbol levels to these values:



1. Experiment with different values to get different results.
2. Open your original map again before continuing with the next exercise.

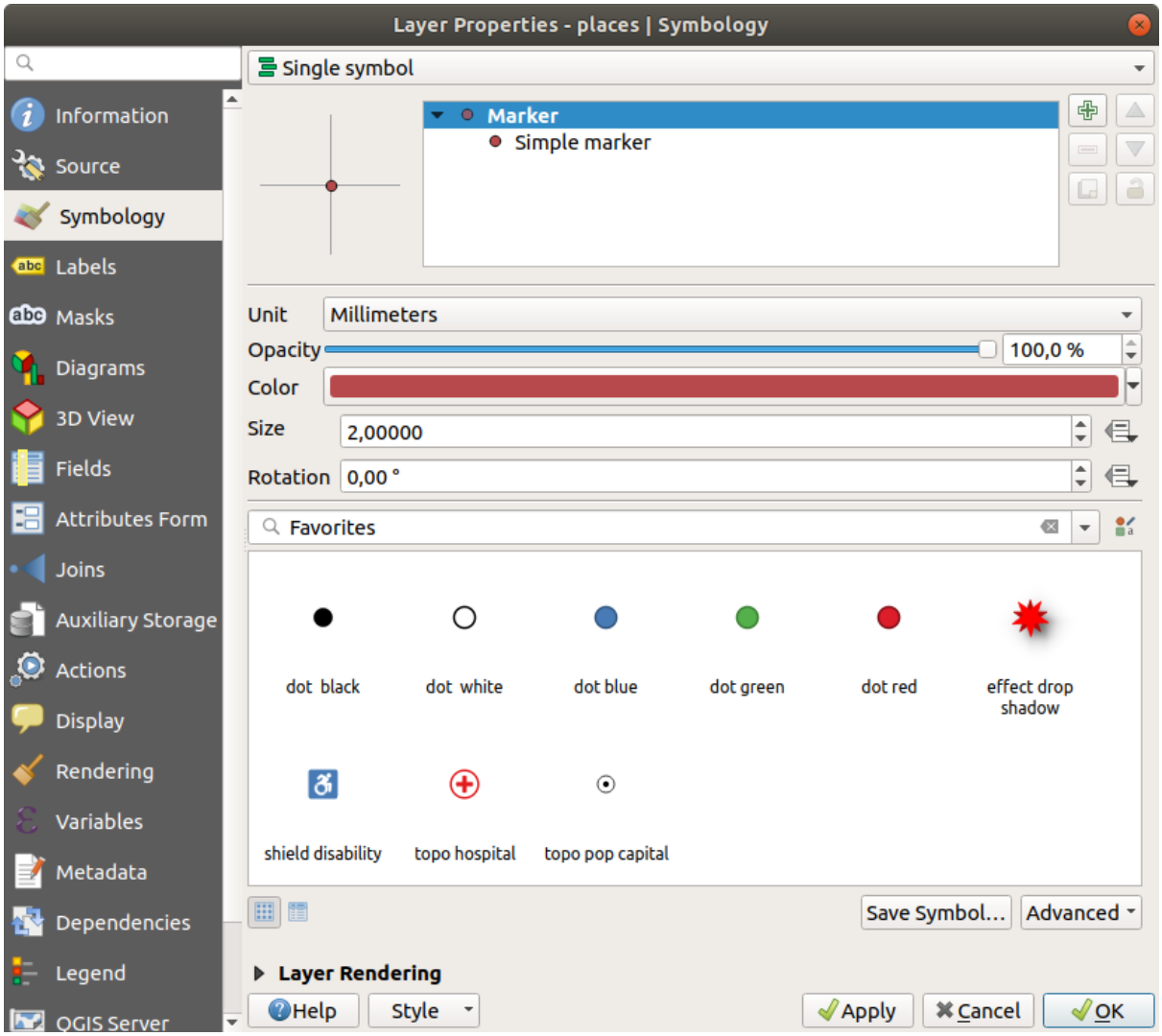
2.4.11 Идем дальше: Типы слоя символа

Помимо установки цветов заливки и использования предустановленных шаблонов, вы можете использовать абсолютно различные типы слоев символов. До сих пор мы использовали только тип Простой заливки *Simple Fill*. Более продвинутые типы слоев символов позволяют еще больше настраивать символы под вас.

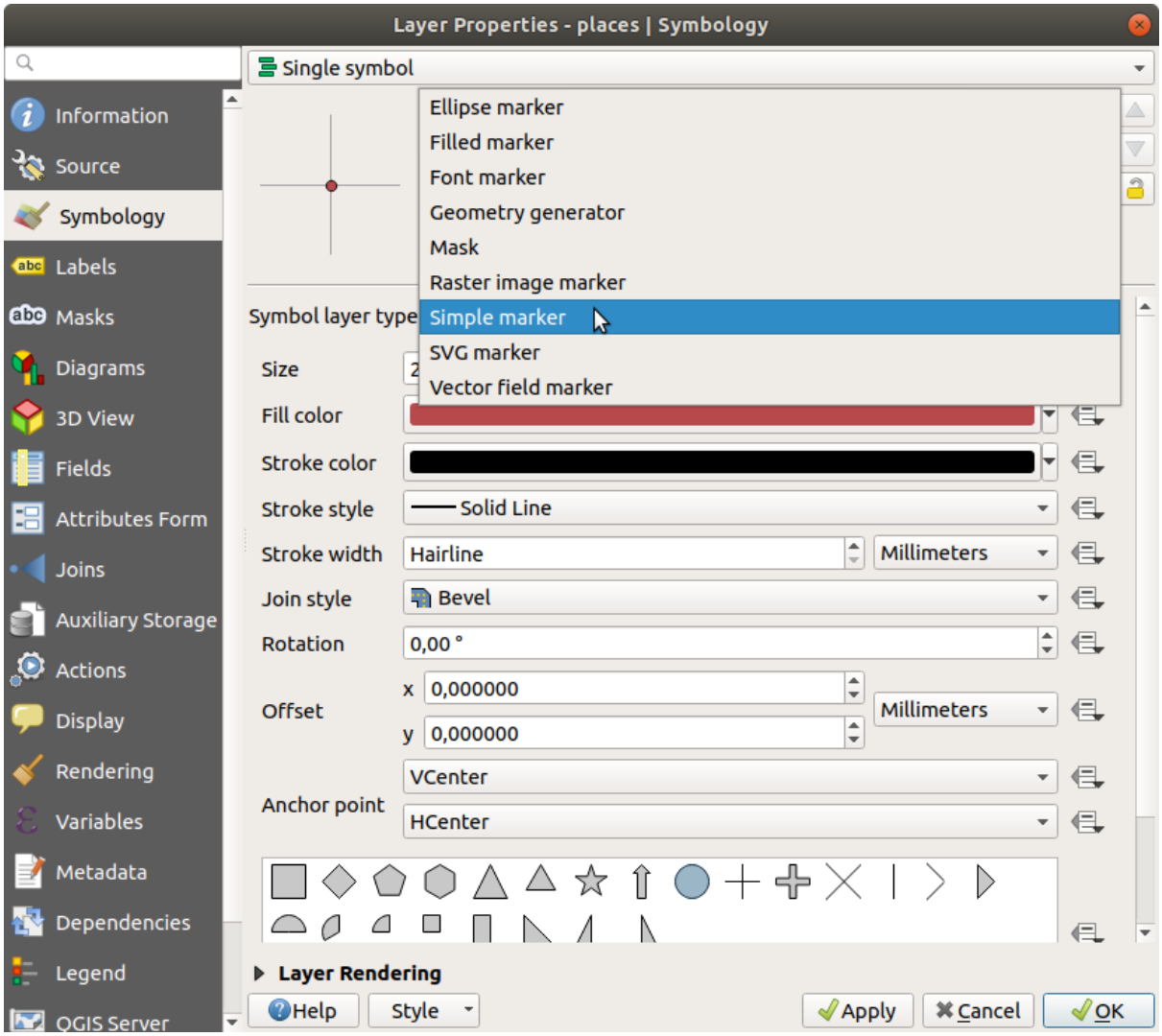
Каждый тип вектора (точка, линия и полигон) имеет свой собственный набор типов слоев символов. Сначала мы рассмотрим типы, которые существуют для точек.

Типы Точечных слоев символа

1. Снимите отметку со всех слоев, кроме Места *places*.
2. Измените свойства символа для слоя *places*:



3. Вы можете получить доступ к различным типам слоев символов, выбрав слой Простой маркер *Simple marker* на дереве слоев символов, а затем кликните на раскрывающееся меню *Symbol layer type*:



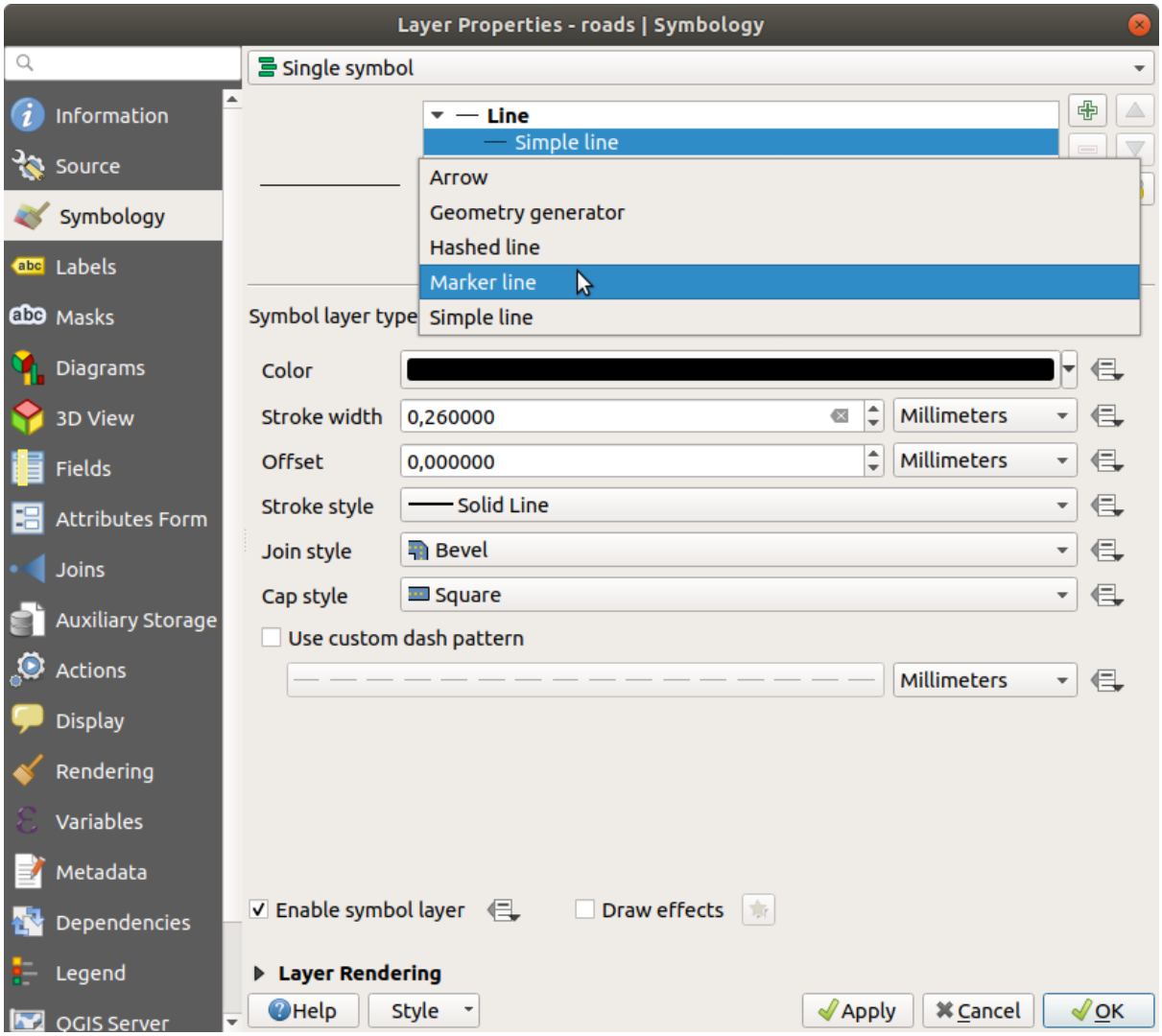
4. Изучите различные доступные вам варианты и выберите символ со стилем, который вы считаете подходящим.

5. В случае сомнений используйте круглый Простой маркер *Simple marker* с белой рамкой и бледно-зеленой заливкой, с размером *Size* в `3.00` и шириной штриха *Stroke width* в `0.5`.

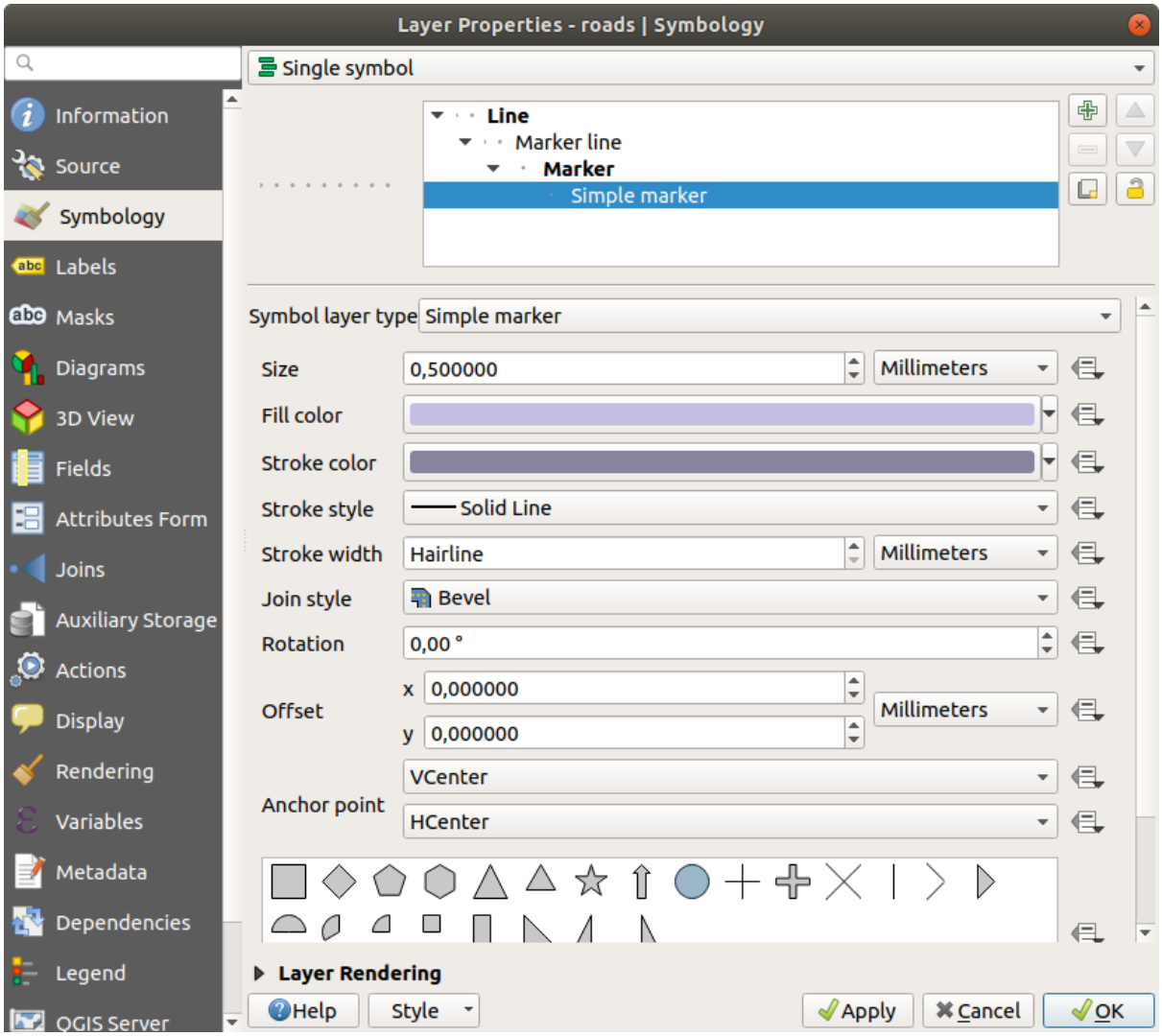
Типы слоев линейных символов

Чтобы увидеть различные параметры, доступные для линейных данных необходимо:

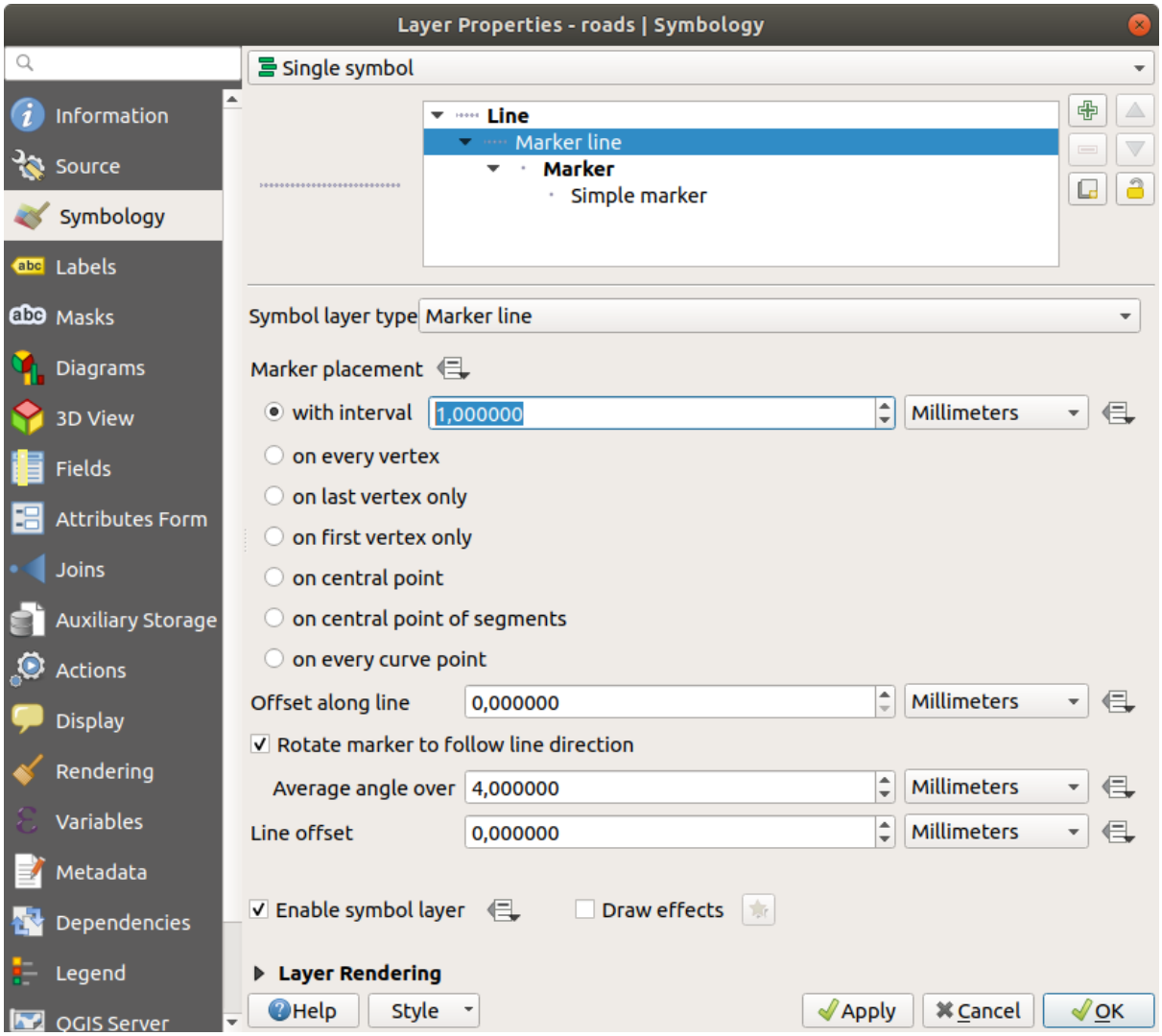
1. Изменить Тип слоя символа *Symbol layer type* с самого верхнего слоя символа *roads* на Линию маркера *Marker line*:



2. Выбрать слой *Simple marker* на дереве слоев символа. Измените свойства символа в соответствии с данным диалоговым окном:



3. Выбрать слой *Marker line* и изменить интервал на `1.00`:



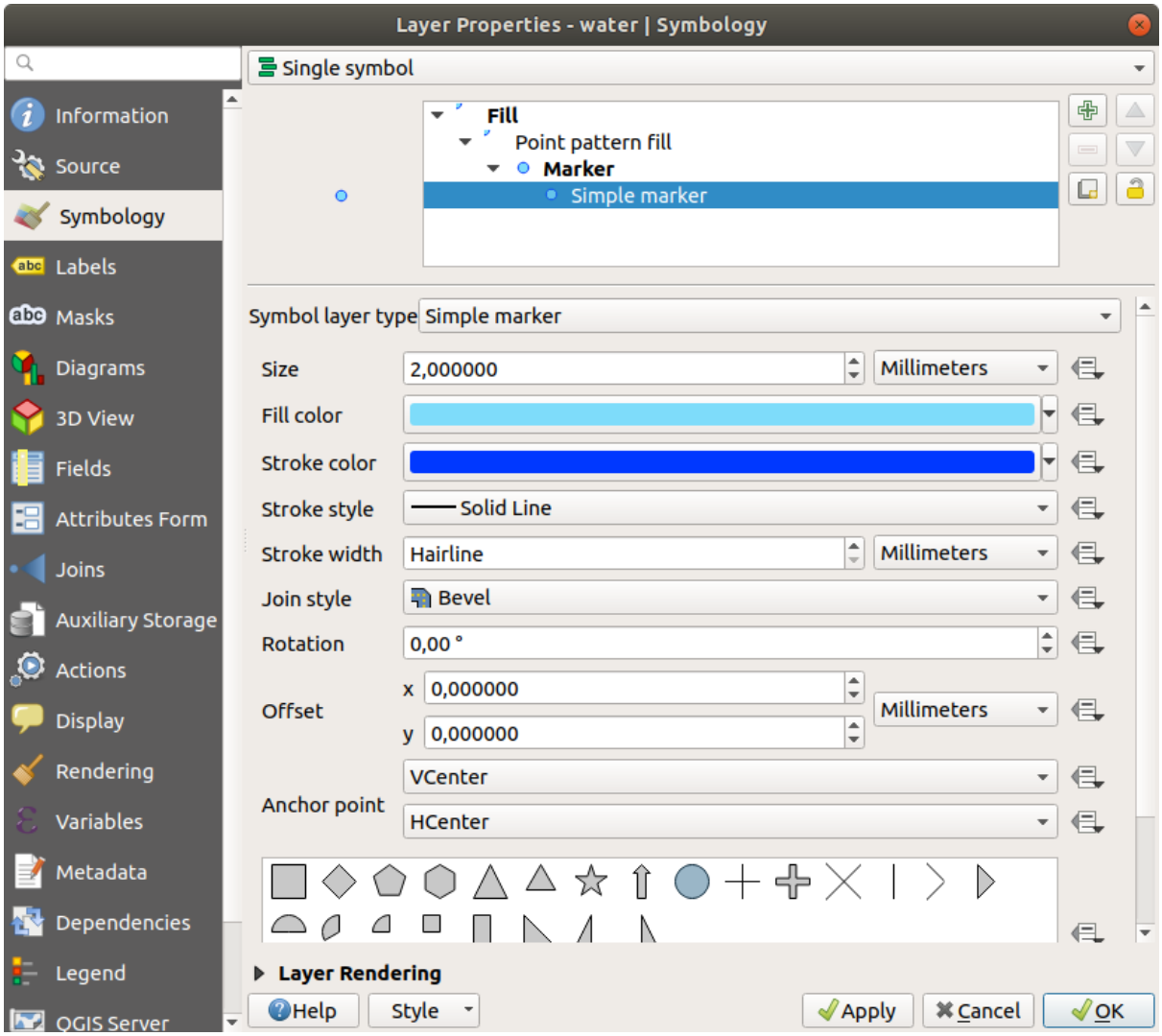
4. Убедиться, что уровни символа указаны верно (при помощи диалогового окна *Advanced* ► *Symbol levels*, которые мы использовали ранее) пред тем, как задать стиль.

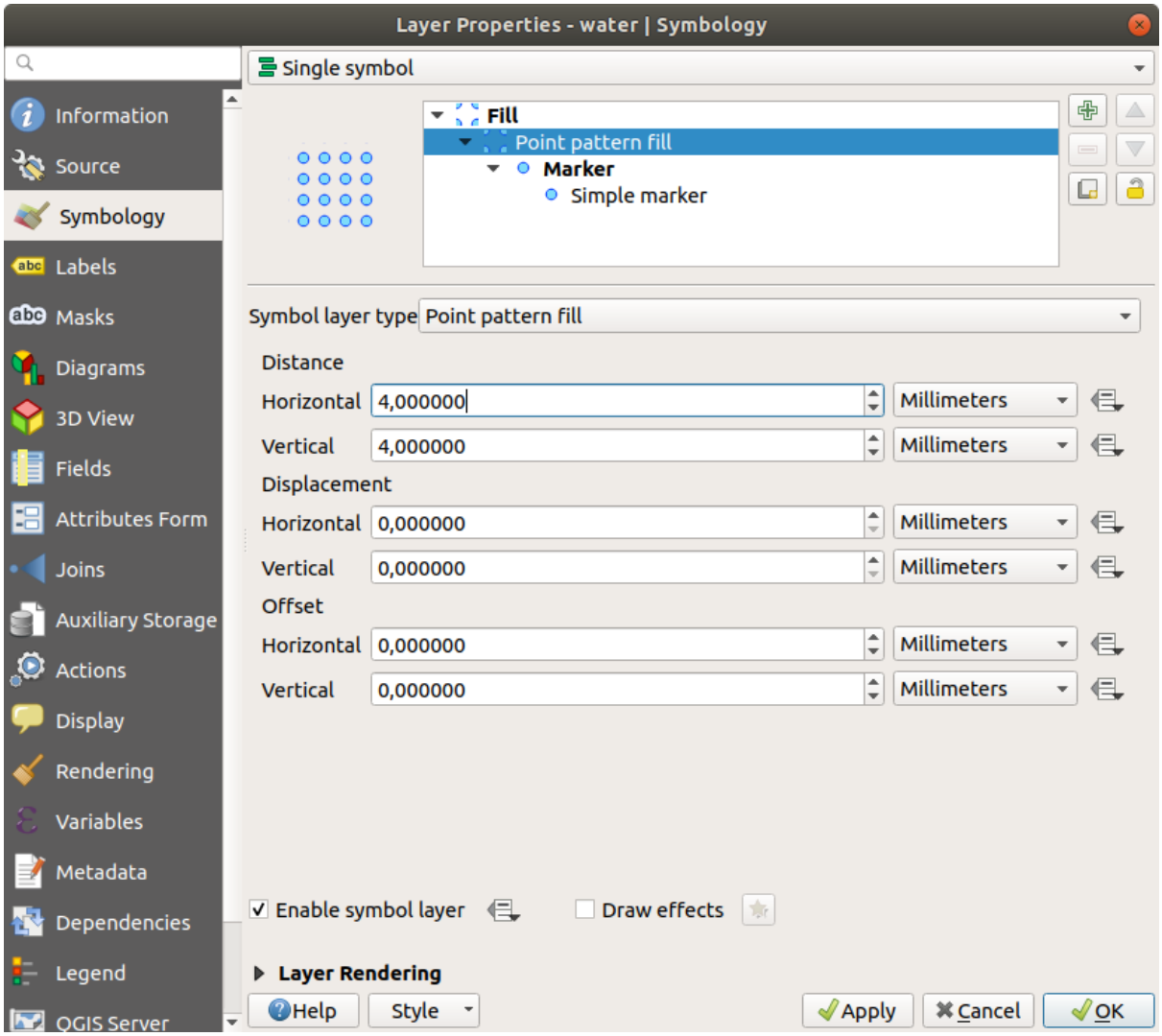
После того, как вы задали стиль, проверьте ваши результаты на карте. Как видите, эти символы меняют направление вместе с дорогой, но не всегда изгибаются вместе с ней. Это удобно для некоторых целей, но не для всех. При желании вы можете изменить слой символа, о котором идет речь, на тот, который был ранее.

Типы слоев символа полигона

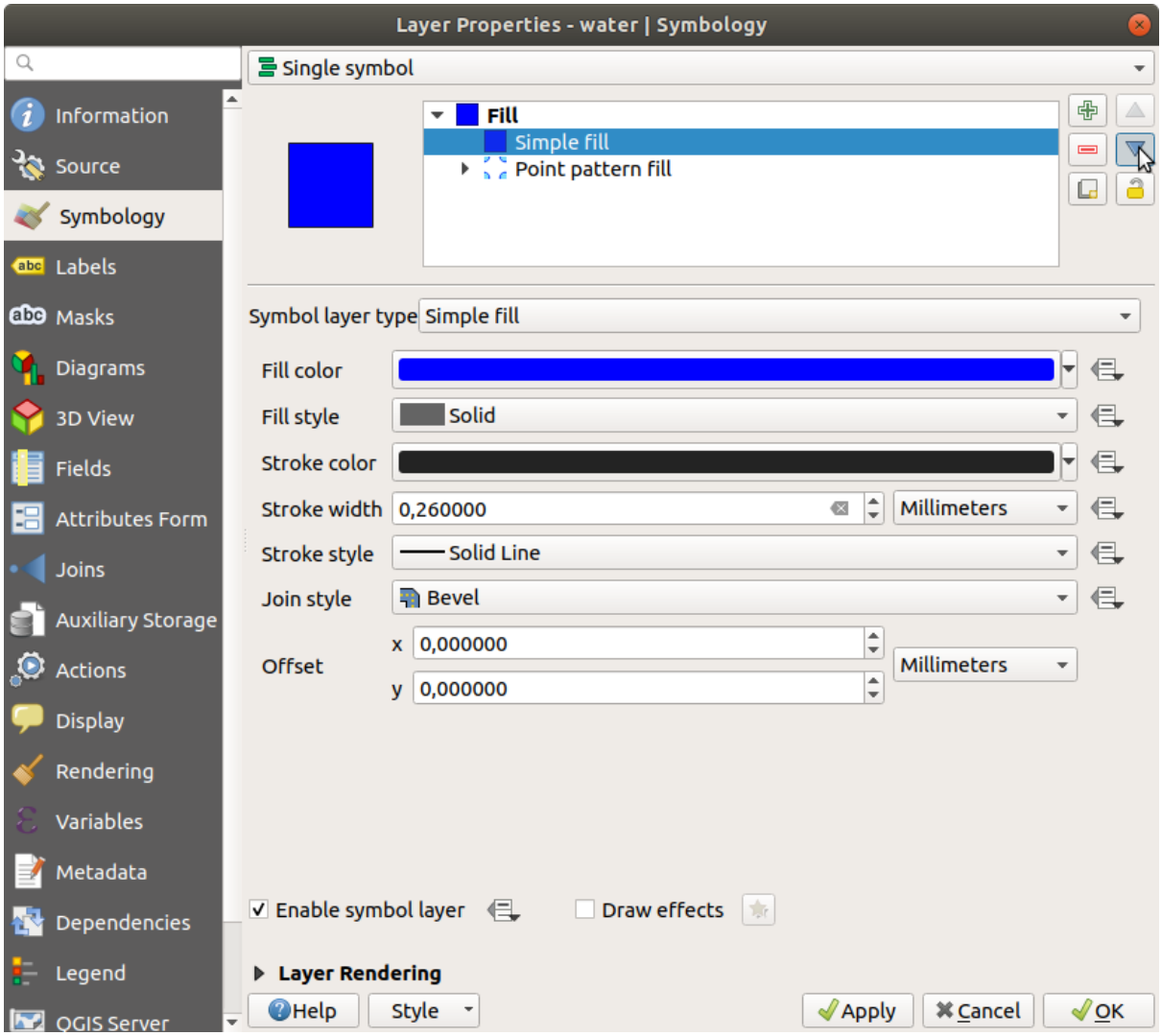
Чтобы увидеть различные параметры, доступные для данных полигона необходимо:

1. Изменить с Тип слоя символа *Symbol layer type* на слой *water*, как вы делали ранее для других слоев.
2. Изучить, что могут сделать различные варианты в списке.
3. Выбрать один из вариантов, который вам подходит.
4. Если вы сомневаетесь, то можно применить Заливку точечным узором *Point pattern fill*, со следующими параметрами:


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



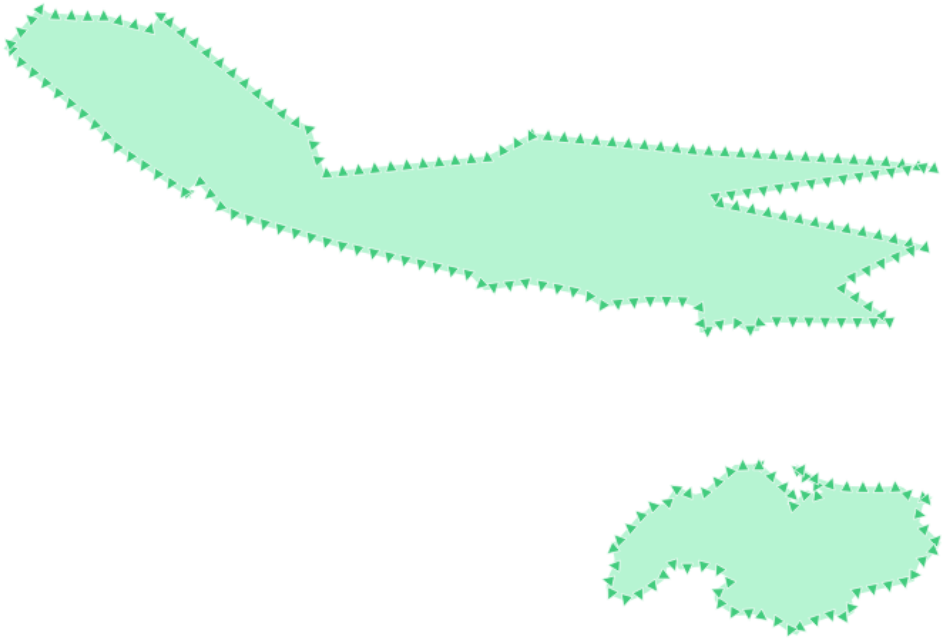
5. Добавить новый слой символа с обычным режимом Простой заливки *Simple fill*.
6. Сделать его таким же светло-голубым с темно-синей рамкой.
7. Переместить его под слой символа Заливка точечным узором с помощью кнопки Переместить вниз *Move down*:



В результате у вас есть текстурированный символ для слоя воды с дополнительным преимуществом, заключающимся в том, что вы можете изменять размер, форму и расстояние до отдельных точек, составляющих текстуру.

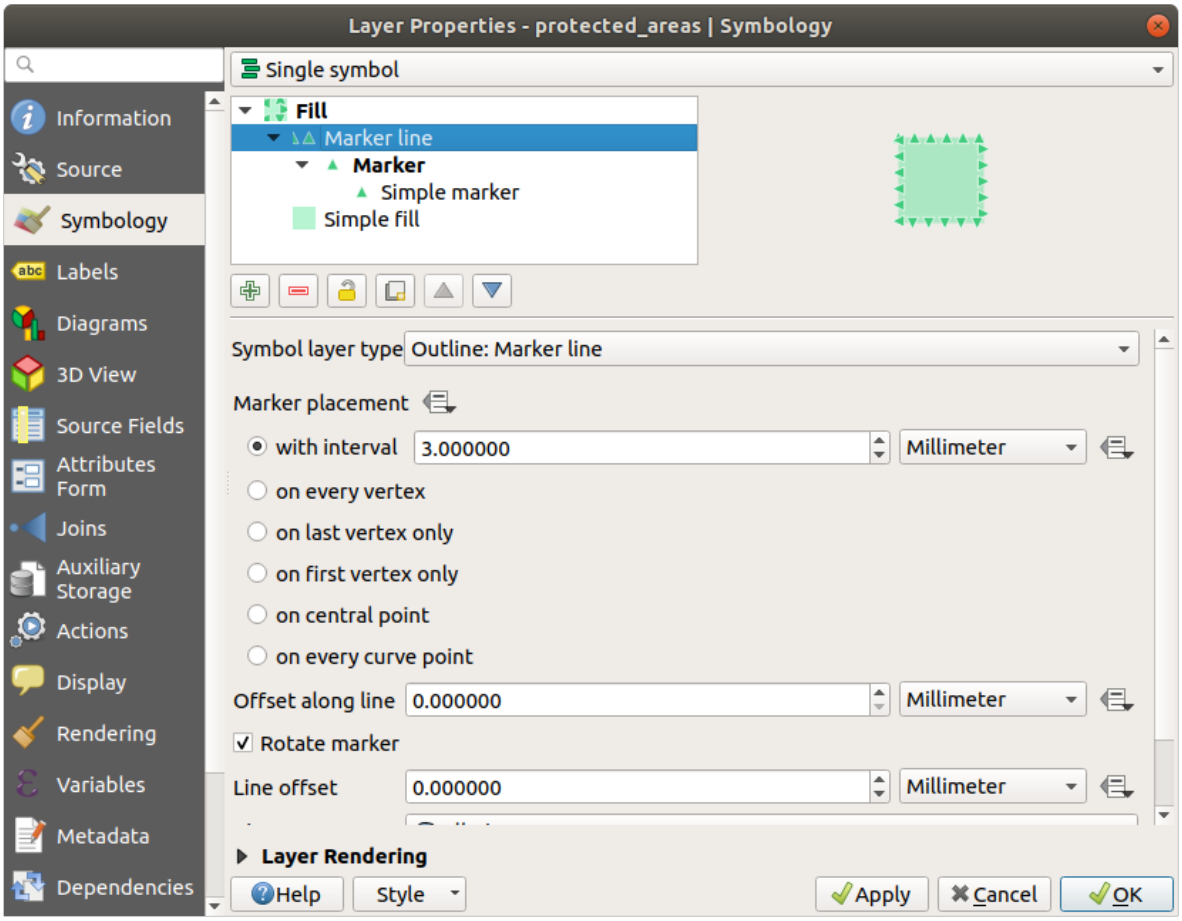
2.4.12 Попробуйте сами

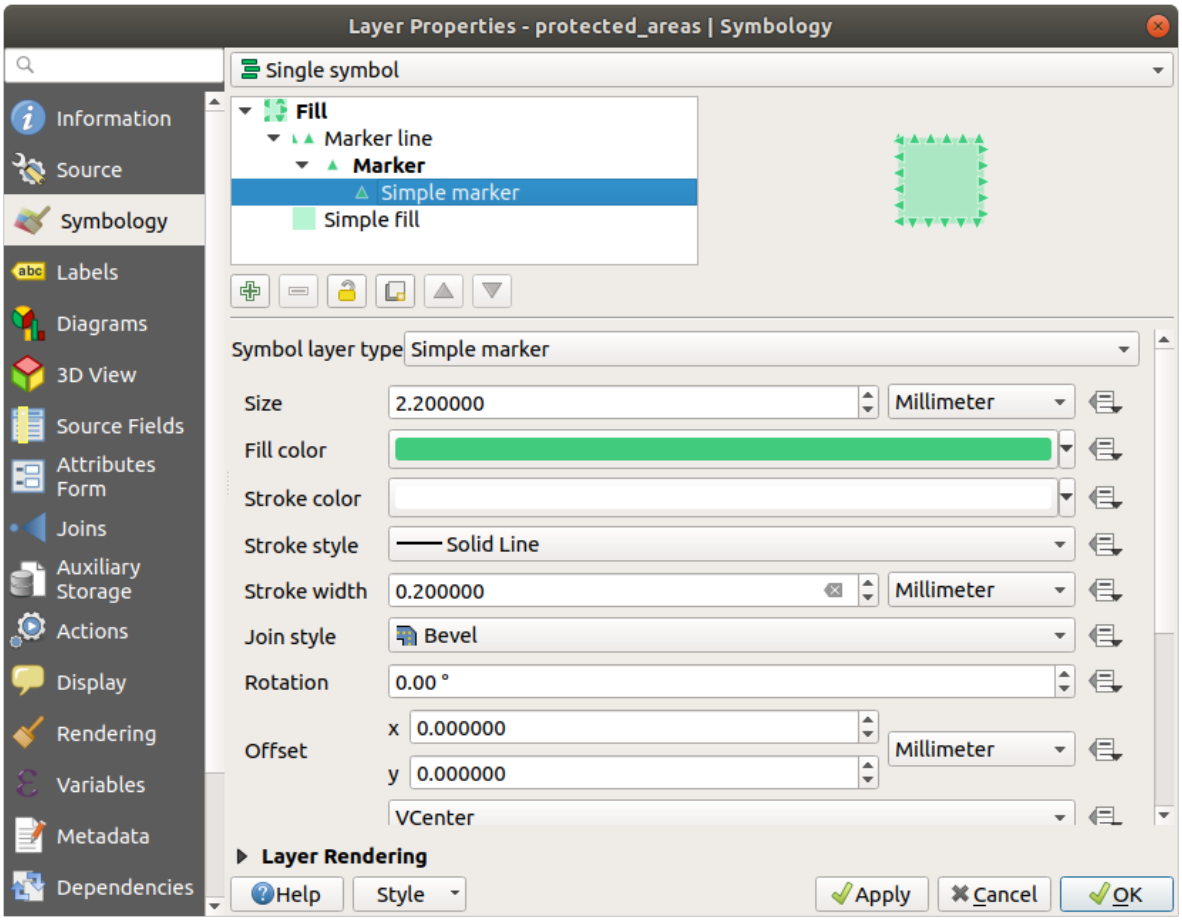
Примените зеленый прозрачный цвет заливки к слою охраняемые территории *protected_areas* и измените контур, чтобы он выглядел так:



Answer

Here are examples of the symbol structure:





2.4.13 Идем дальше: Символика генератора геометрии

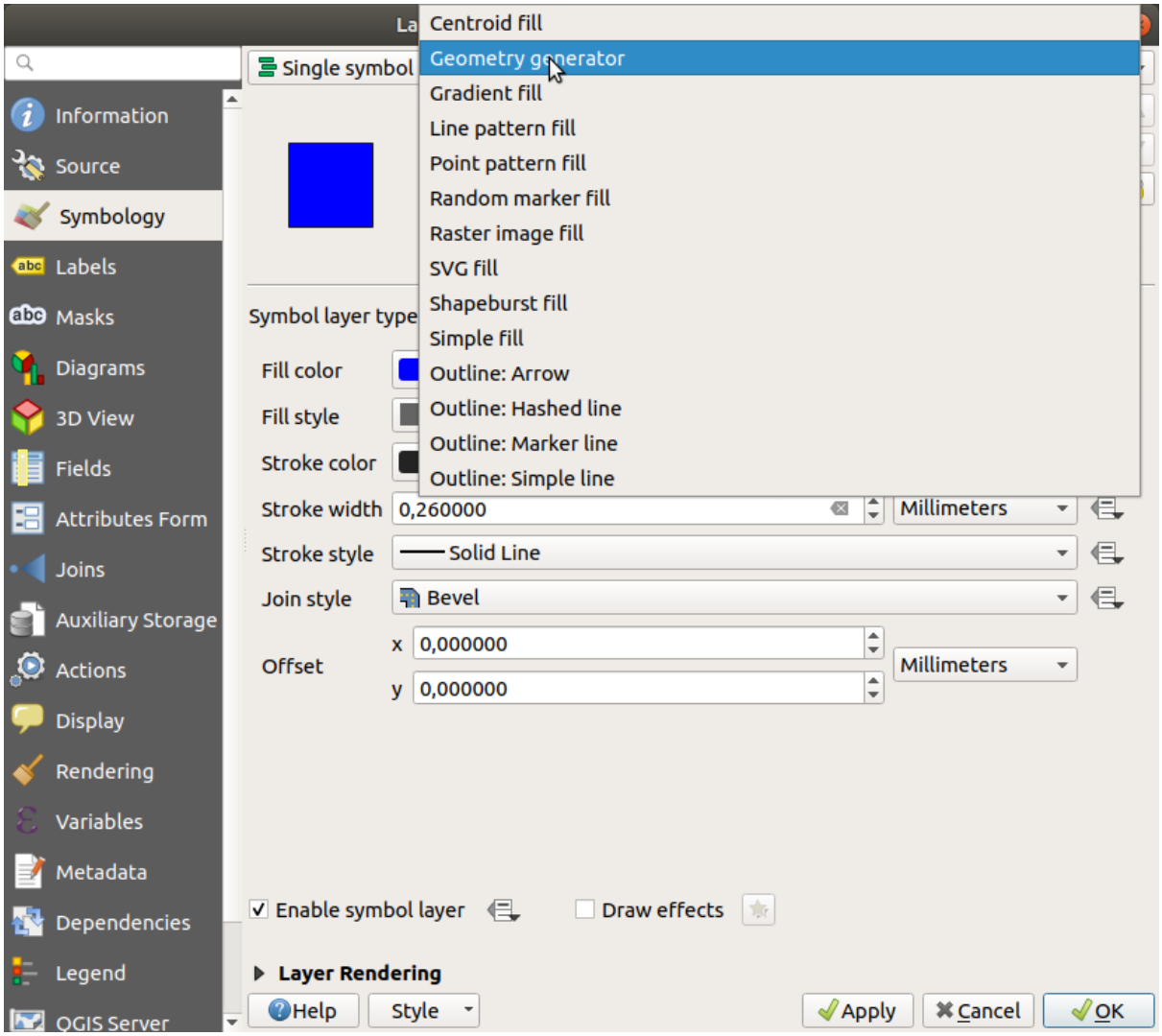
Вы можете использовать символику генератора геометрии со всеми типами слоев (точки, линии и полигоны). Полученный символ напрямую зависит от типа слоя.

Вкратце, символика генератора геометрии позволяет выполнять некоторые пространственные операции внутри самих символов. Например, вы можете запустить реальную центроидную пространственную операцию на слое полигона, не создавая точечный слой.

Более того, у вас есть все возможности стилизации для изменения внешнего вида получившегося символа.

Давайте попробуем!

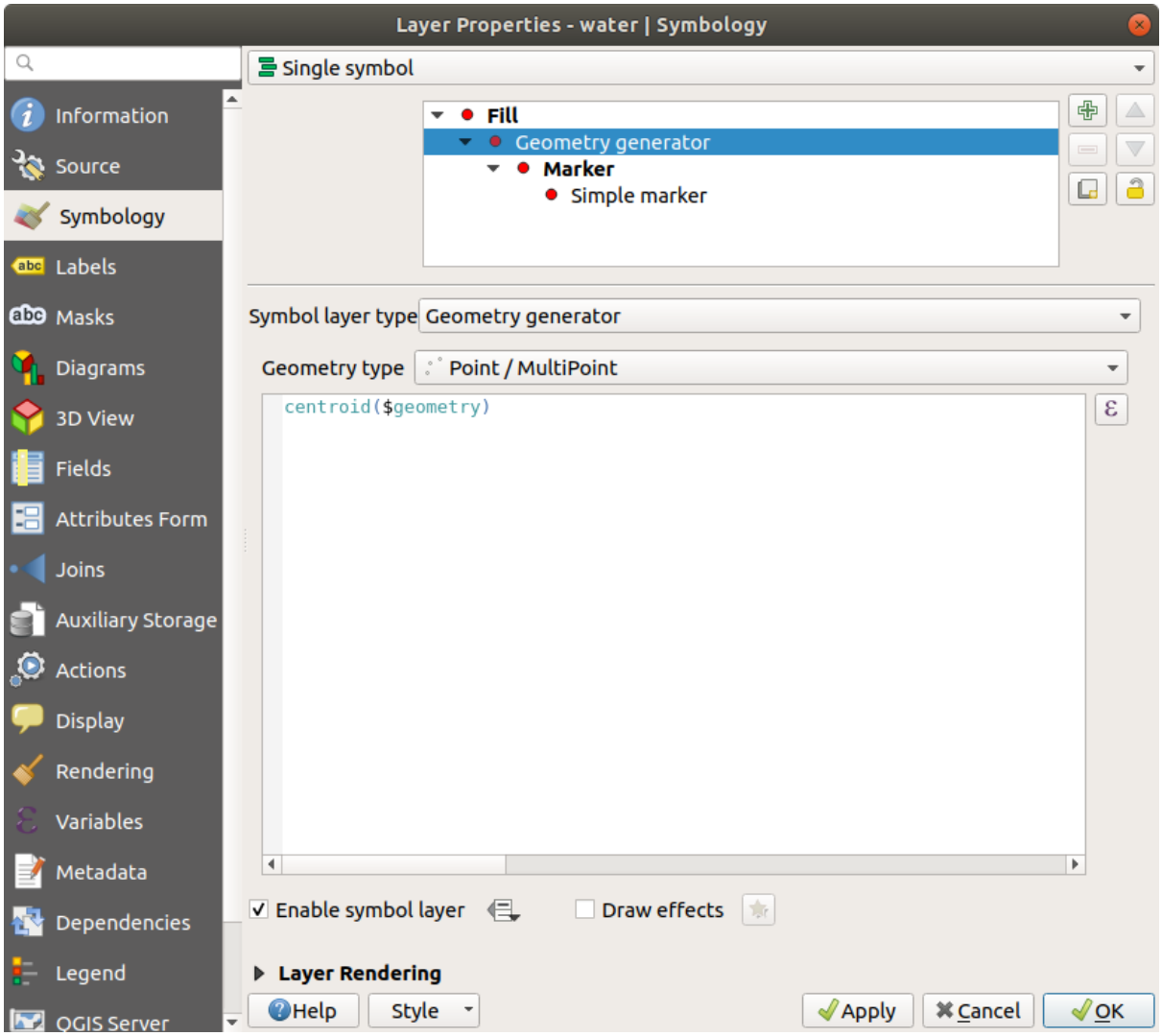
1. Выберите слой *water*.
2. Кликните по *Simple fill* и измените *Symbol layer type* на *Geometry generator*.



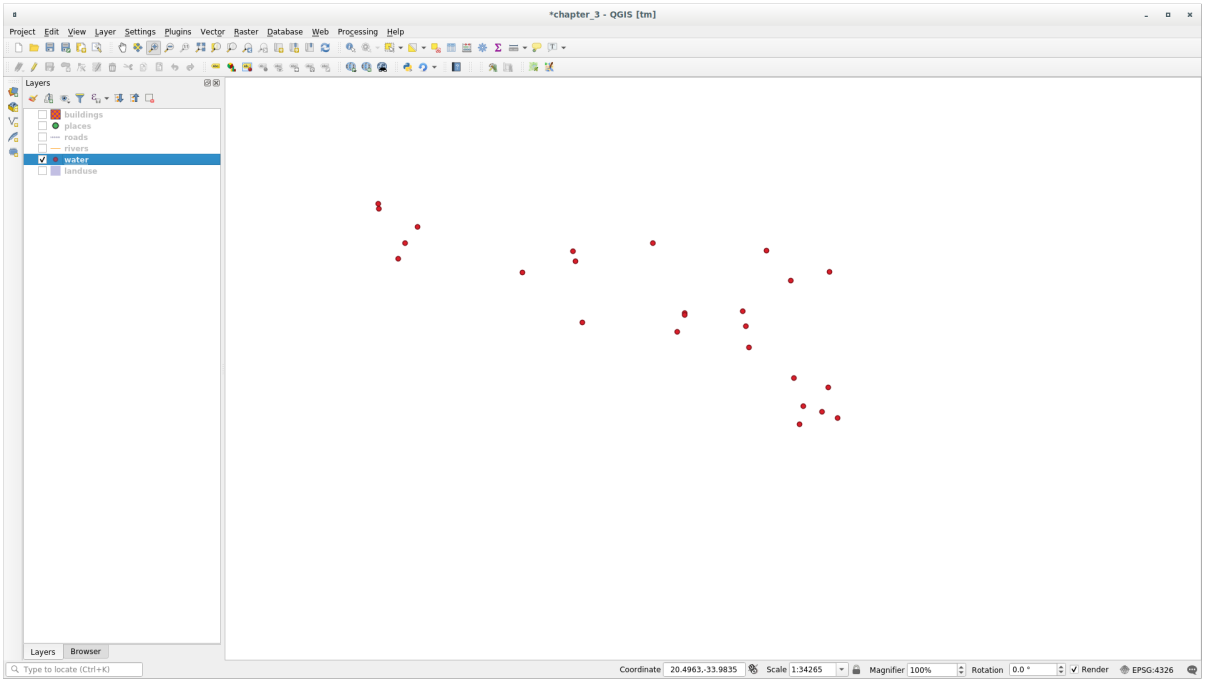
3. Перед тем, как приступить к написанию пространственного запроса, мы должны выбрать Тип геометрии на выходе, в результатах. В этом примере мы собираемся создать центроиды для каждого объекта, поэтому измените Тип геометрии на *Point / Multipoint*.

4. Теперь давайте напишем запрос на панели запросов:

```
centroid($geometry)
```



5. Когда вы нажмете *OK*, вы увидите, что слой *water* water отображается как точечный слой! Мы только что выполнили пространственную операцию в самом слое символики, разве не удивительно?



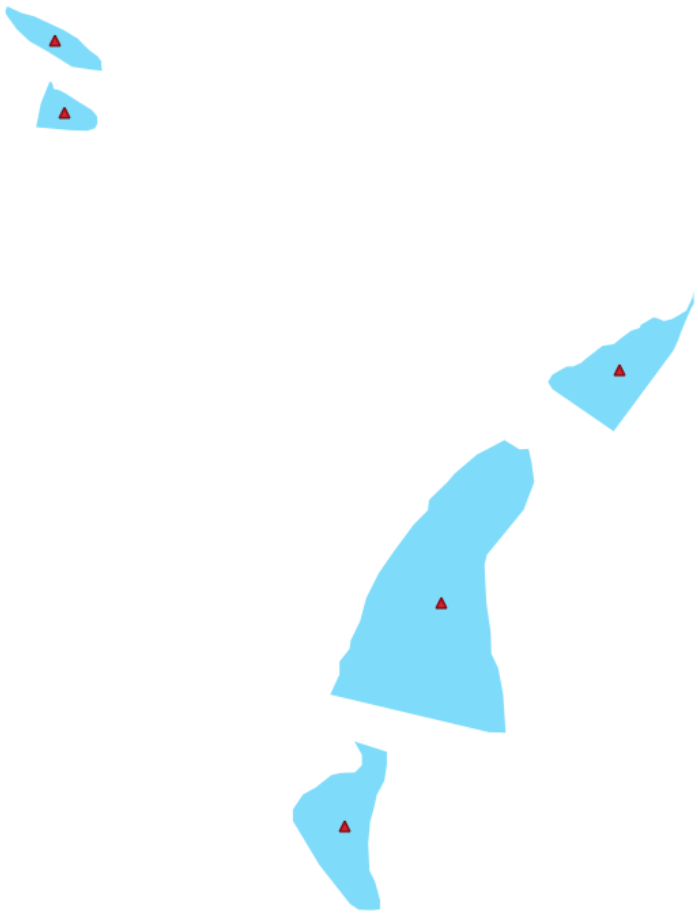
Благодаря символике генератора геометрии вы действительно можете выйти за рамки *обычной* символики.

Попробуйте сами

Генератор геометрии - это еще один уровень символа. Попробуйте добавить еще один *Simple fill* под *Geometry generator*.

Измените также внешний вид Простого маркера символики Генератора геометрии.

Окончательный результат должен выглядеть следующим образом:



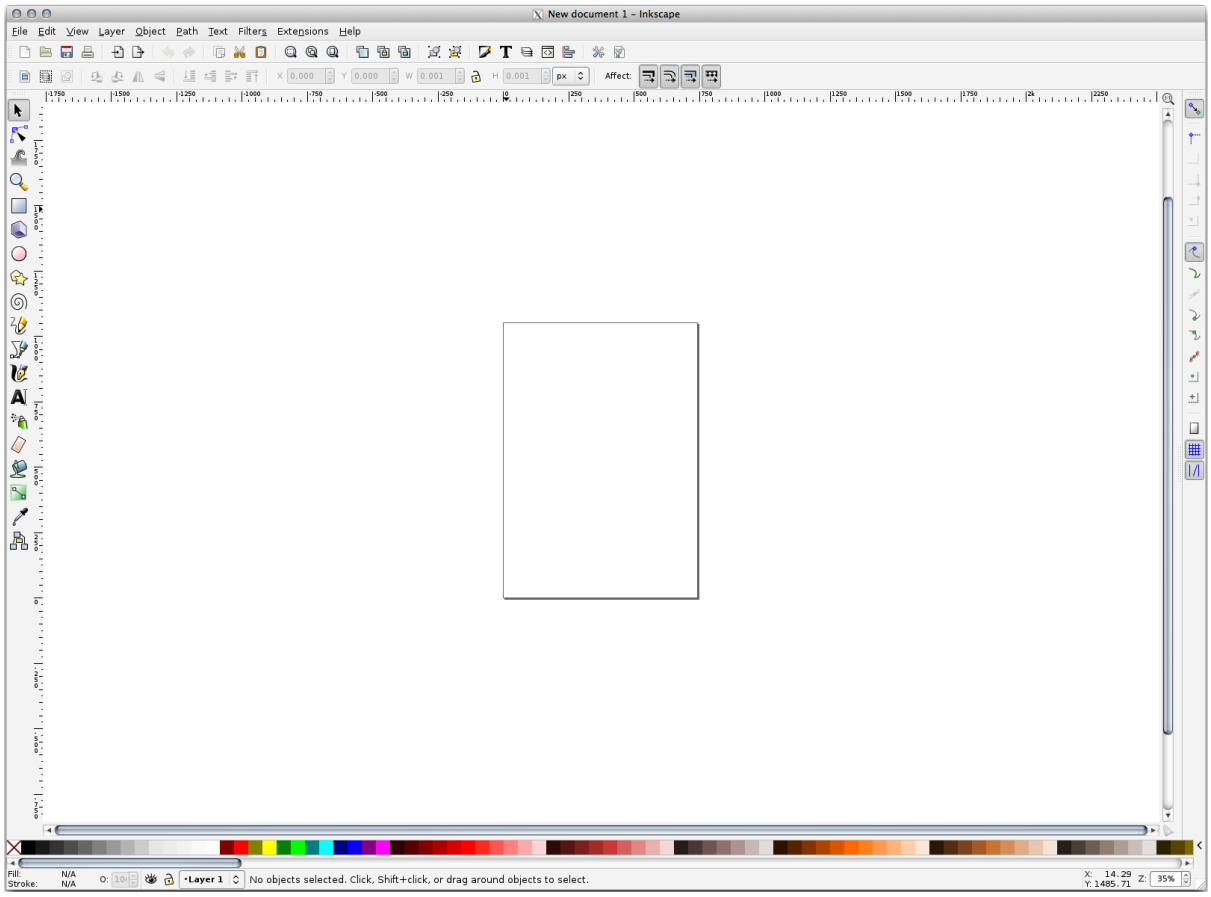
Answer

1. Click on the button to add another Symbol level.
2. Move the new symbol at the bottom of the list clicking the button.
3. Choose a good color to fill the water polygons.
4. Click on *Marker* of the Geometry generator symbology and change the circle with another shape as your wish.
5. Try experimenting other options to get more useful results.

2.4.14 Идем дальше: Создаем пользовательскую заливку формата SVG

Примечание: Для выполнения данного упражнения вам потребуется установить бесплатную программу для редактирования векторных изображений `Inkscape `_.

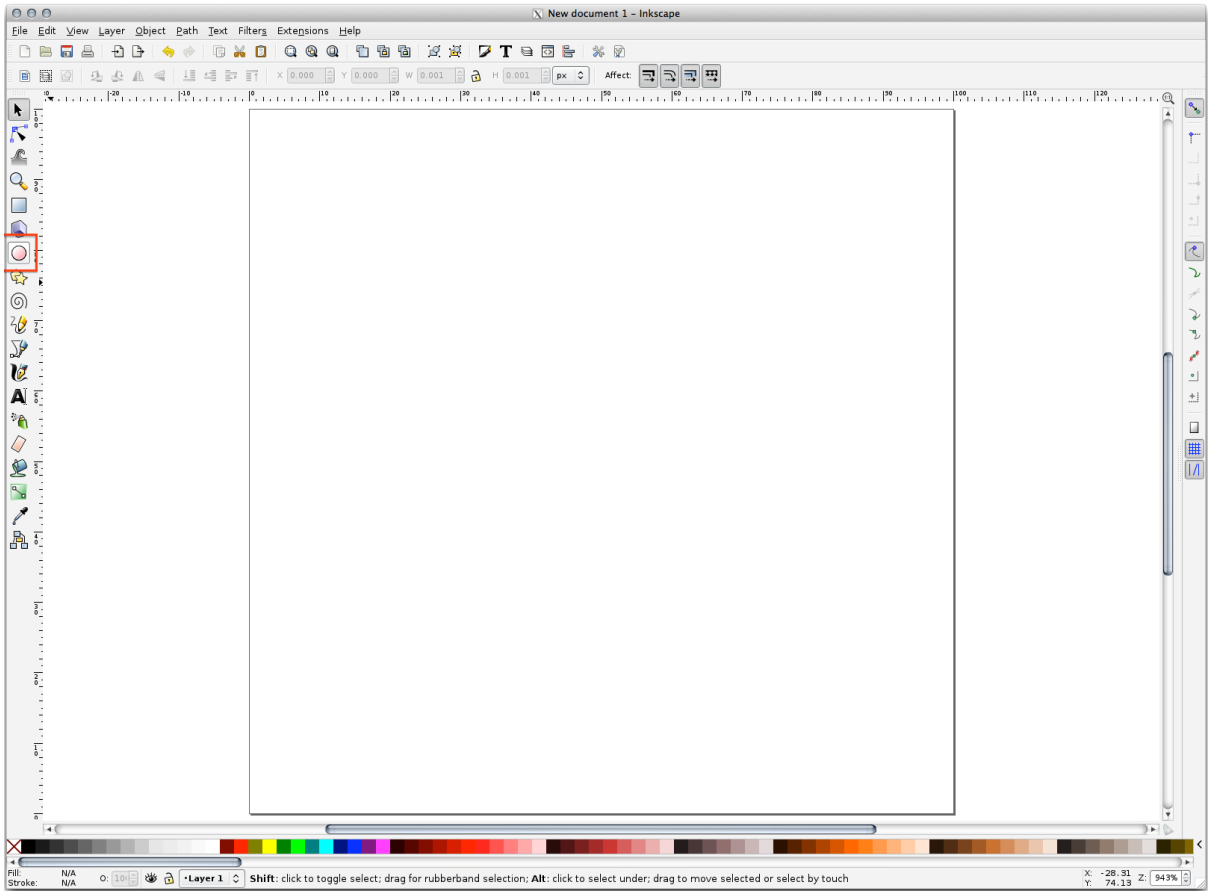
1. Запустите программу Inkscape. Вы увидите следующий интерфейс:



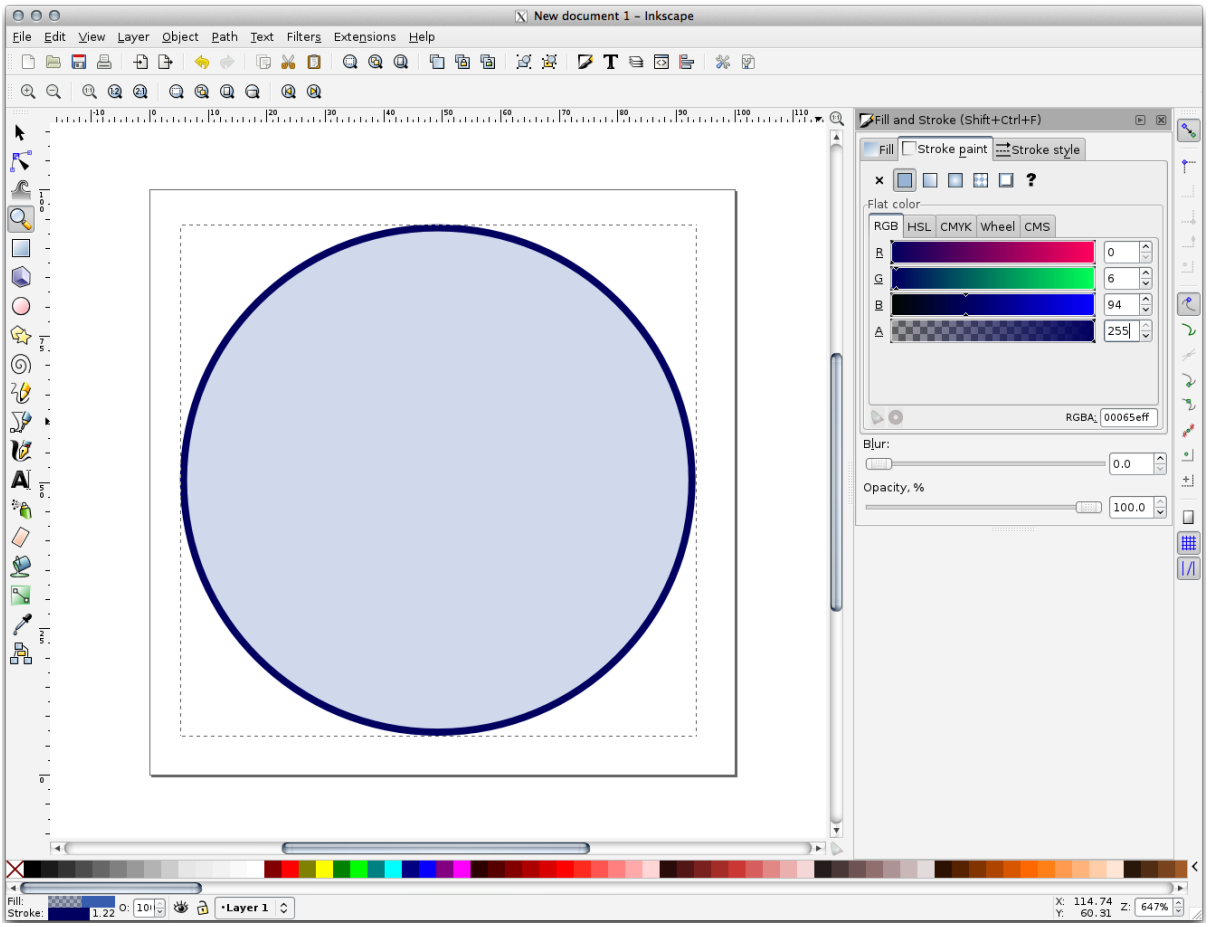
Вам это должно быть знакомо, если вы использовали другие программы для редактирования векторных изображений, такие как Corel.

Сначала мы изменим размер полотна карты на размер, подходящий для небольшой текстуры.

2. Кликните на элемент меню *File* ► *Document Properties*. Откроется диалоговое окно *Document Properties*.
3. Измените Единицы *Units* на Пиксели *px*.
4. Измените Ширину *Width* и Высоту *Height* на `100`.
5. Закройте диалоговое окно по завершении.
6. Кликните на элемент меню *View* ► *Zoom* ► *Page*, чтобы просмотреть страницу, с которой вы работаете.
7. Выберите инструмент Круг *Circle*:

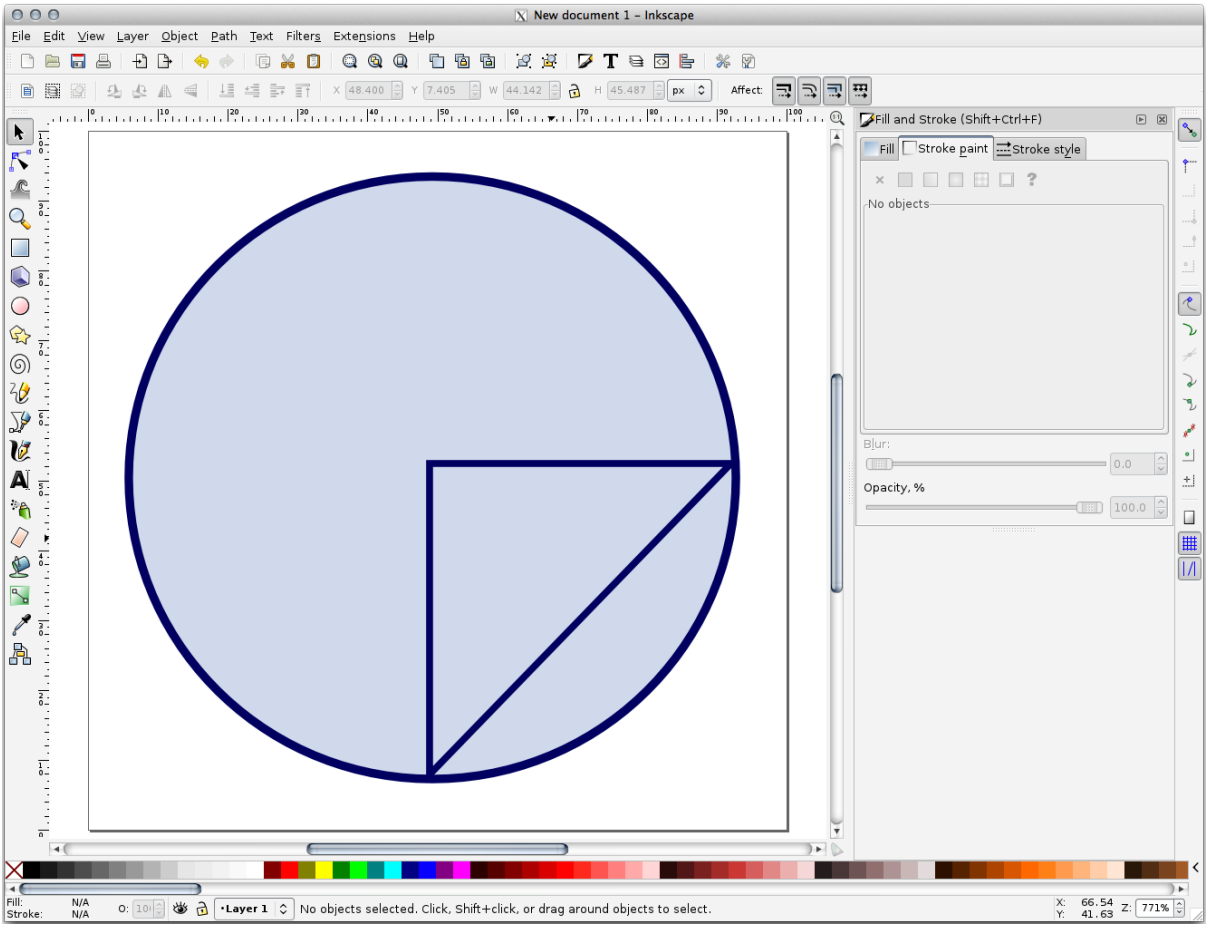


8. Кликните и тащите по странице, чтобы нарисовать эллипс. Чтобы превратить эллипс в круг удерживайте кнопку `Ctrl`, чтобы эллипс превратился в круг, пока вы его рисуете.
9. Кликните правой кнопкой мыши на только что созданный круг и откройте его параметры Заливка и Обводка *Fill and Stroke*. Вы можете изменить его визуализацию, например:
    1. Изменить цвет Заливки :guilabel:`Fill`на бледно-серо-голубой,
    2. Задать более темный цвет рамки на вкладке Мазки *Stroke paint*,
    3. Нарисуйте линию при помощи инструмента Ручка *Stroke style*.



10. Нарисуйте линию при помощи инструмента Ручка *Pencil*:

    1. Кликните один раз, чтобы начать линию. Удерживайте `Ctrl` чтобы сделать привязку к приращению в 15 градусов.
    2. Переместите указатель по горизонтали и поставьте точку простым нажатием на клик.
    3. Кликните и привяжите к вершине линии и проведите вертикальную линию, заканчивающуюся простым кликом.
    4. Теперь соедините две конечные вершины.
    5. Измените цвет и ширину символа треугольника, чтобы он соответствовал обводке круга, и переместите его по мере необходимости, чтобы в итоге получился следующий символ:

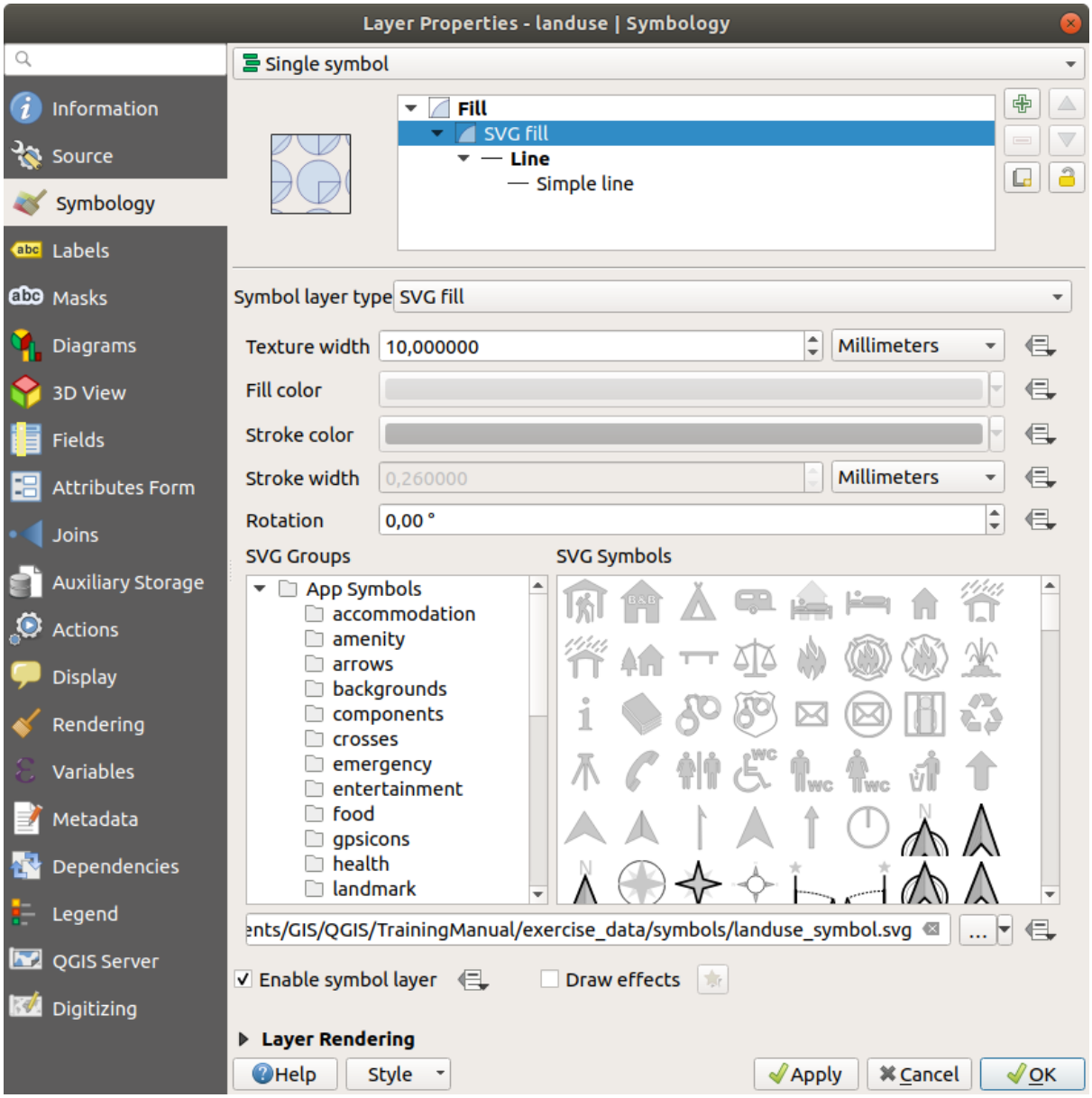


11. Если получившийся символ вас удовлетворил, то сохраните его как *landuse_symbol* в каталоге, в котором находится данный курс, в `exercise_data/symbols`, как файл SVG.

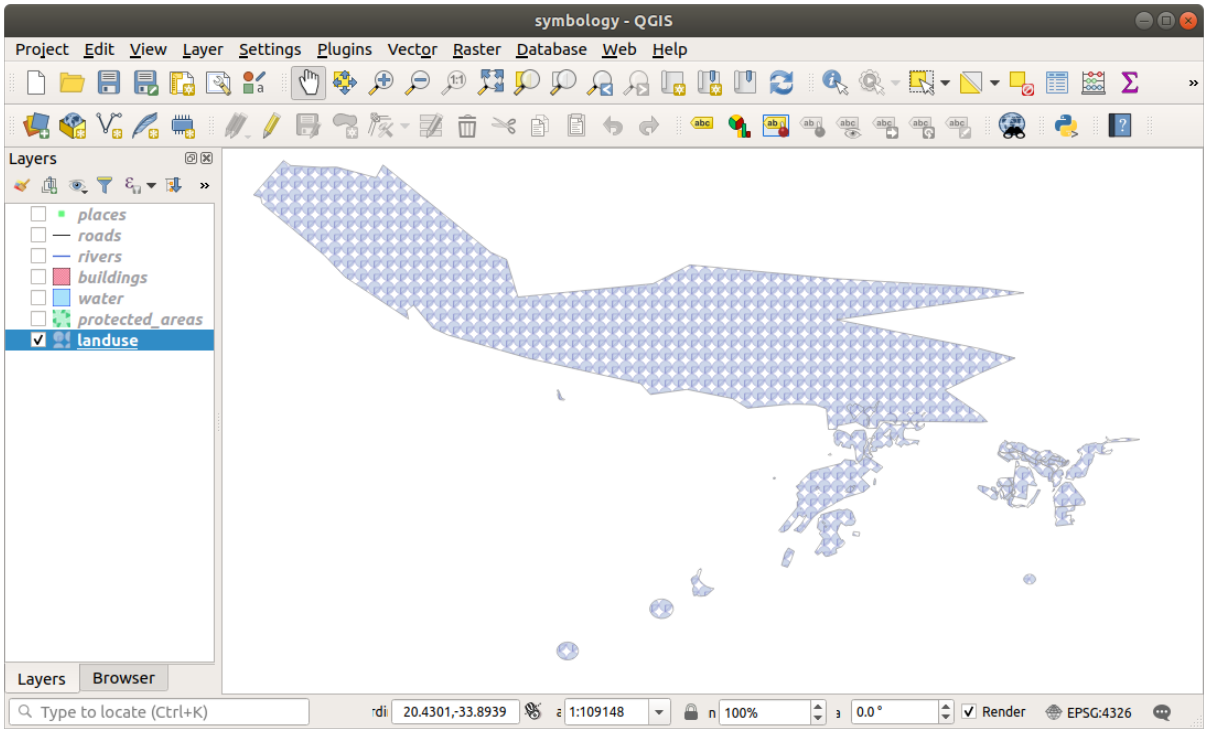
Выполните следующее в QGIS:

1. Откройте *Layer Properties* для слоя `landuse`.
2. Во вкладке  *Symbology*, измените структуру символа через замену Типа слоя символа *Symbol Layer Type* на Файл *SVG Fill*``, как указано ниже.
3. Кликните кнопку … и затем Выбрать файл *Select File…*, чтобы выбрать картинку SVG.

   Файл добавлен в дерево символов, и теперь вы можете настраивать его различные характеристики (цвета, угол, эффекты, единицы измерения …).



После того, как вы подтвердите действие в диалоговом окне, функции в слое *landuse* теперь должны быть набором символов, показывающих текстуру, подобную той, что на следующей карте. Если текстур не видно, вам может потребоваться приблизить полотно карты или установить в свойствах слоя большую Ширину текстуры *Texture width*.



2.4.15 В заключении

Изменение символов для разных слоев, превратили коллекцию векторных файлов в удобочитаемую карту. Вы не только видите, что происходит, но и даже любуетесь этим!

2.4.16 Дополнительные материалы для чтения

Примеры красивых карт *Examples of Beautiful Maps*

2.4.17 Что дальше?

Изменение символов для целых слоев полезно, однако информация, содержащаяся в каждом слое, еще не доступна для тех, кто читает эти карты. Как называются улицы? К каким административным районам относятся определенные районы? Какова относительная площадь ферм? Вся эта информация по-прежнему скрыта. На следующем занятии мы разберем, как представить эти данные на вашей карте.

Примечание: Вы не забыли сохранить вашу карту недавно?

ГЛАВА 3

Модуль: Классификация векторных данных

Классификация векторных данных позволяет присваивать различные символы функциям (разные объекты в одном и том же слое) в зависимости от их атрибутов (атрибутов). Это позволяет любому, кто использует карту, легко видеть атрибуты различных функций.

3.1 Занятие: Атрибуты векторных данных

Векторные данные, вероятно, являются наиболее распространенным видом данных в повседневном использовании GIS. Векторная модель представляет расположение и форму географических функций с помощью точек, линий и полигонов (а для трехмерных данных также поверхность и объем), в то время как другие ее свойства включены как атрибутыатрибуты (часто представлены в виде таблицы в QGIS).

До сих пор ни одно из изменений, которое мы внесли в карту, не повлияло на отображаемые объекты. Другими словами, все участки земли похожи друг на друга, и все дороги похожи друг на друга. Глядя на карту, смотрящий ничего не знает о дорогах, которые он видит, только то, что на определенной местности есть дорога определенной формы.

Но вся сила GIS состоит в том, что все объекты, которые видны на карте, также имеют атрибуты. Карты в GIS - это не просто изображения. Они представляют не только объекты на местах, но и информацию об этих объектах.

Цель данного занятия: узнать о структуре векторных данных и изучить атрибуты данных объекта.

3.1.1  Идем дальше: Обзор атрибутов слоя

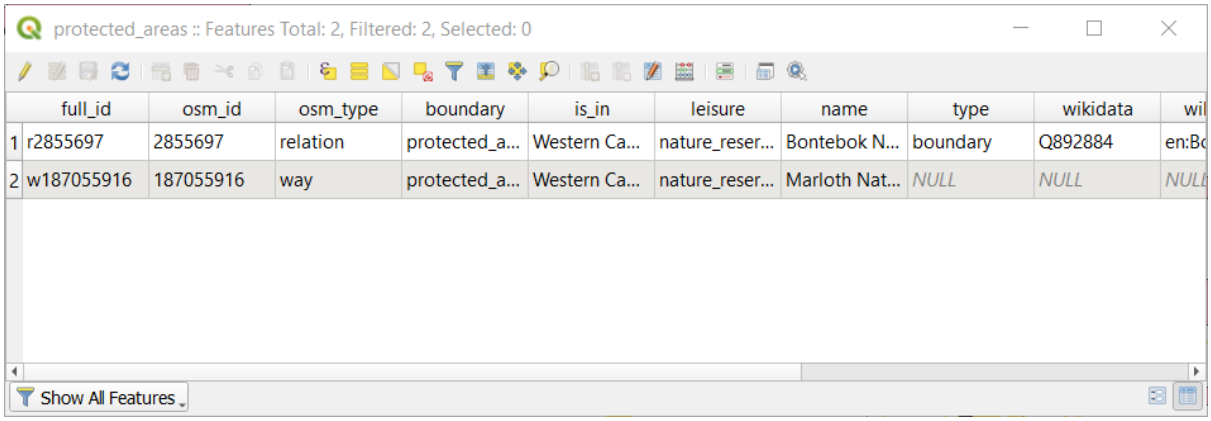
Важно знать, что данные, с которыми вы будете работать, не только показывают, **где** находятся объекты в пространстве, но также говорят вам, **что** это за объекты.

Исходя из предыдущего упражнения на вашу карту должен быть загружен слой `protected_areas`. Если он не загружен, то вы можете найти набор данных формата ESRI Shapefile `protected_areas.shp` *ESRI Shapefile* в каталоге `exercise_data/shapefile`.

Полигоны, представляющие охраняемые территории, составляют **пространственные данные**, но мы можем узнать больше об охраняемых территориях, изучив **таблицу атрибутов**.

1. На панели Слои *Layers* кликните слой `protected_areas`, чтобы выбрать его.

2. На Панели Инструменты атрибутов *Attributes Toolbar* кликните на кнопку Открыть таблицу атрибутов  Open Attribute Table. Откроется новое окно с таблицей атрибутов слоя `protected_areas`.



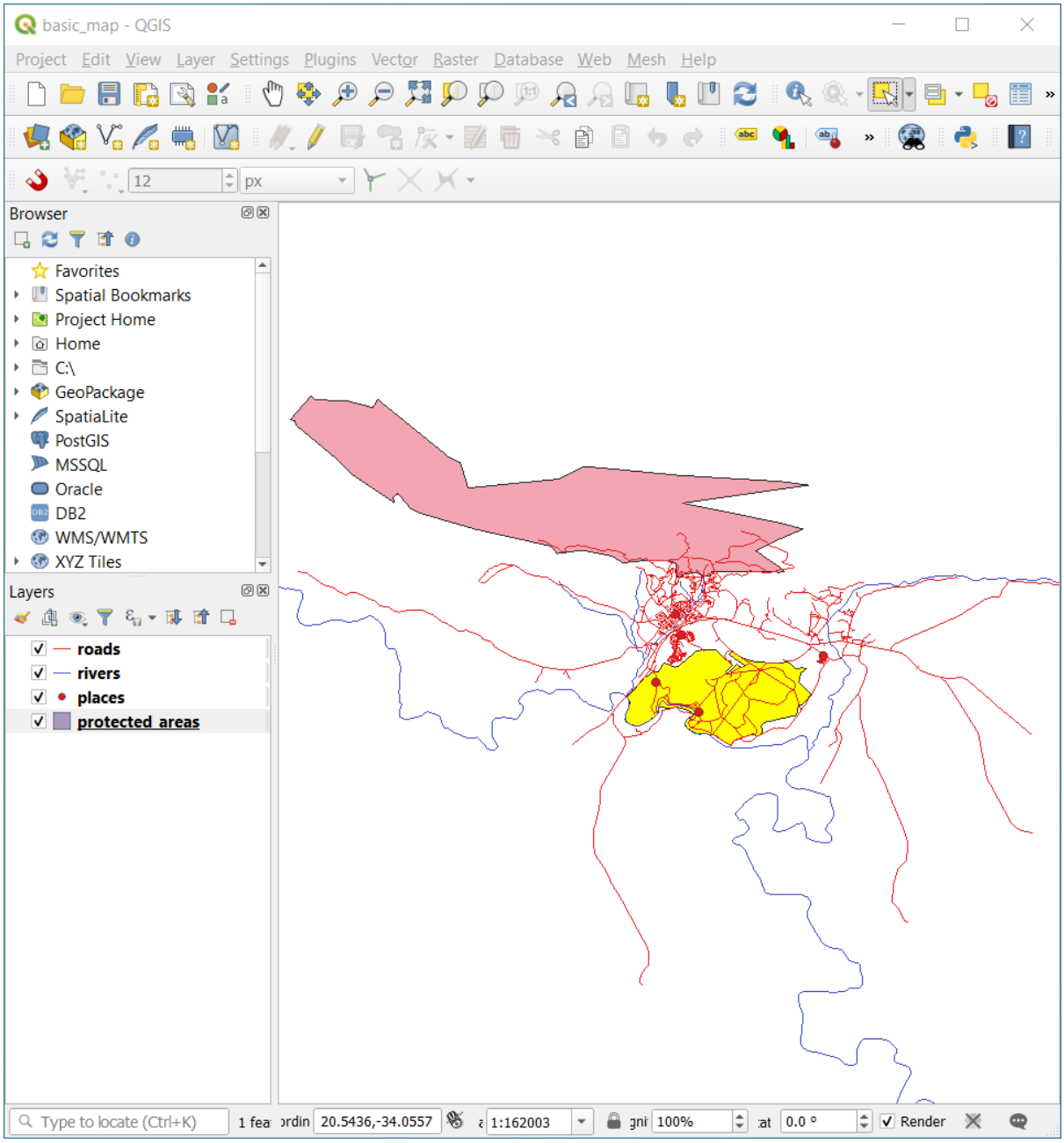
Строка называется **Запись**, она связана с **Функцией** на Полотне Карты, например, как полигон. Столбец называется **Поле** (или **Атрибуты**) и имеет наименование, которое помогает описать его, например Имя `name` или Идентификатор `id`. Значения в ячейках называются **Значения атрибутов**. Данные определения обычно используются в GIS, поэтому полезно с ними ознакомиться.

В слое `protected_areas` имеются две **функции**, представленные двумя полигонами, которые мы видим на Полотне карты.

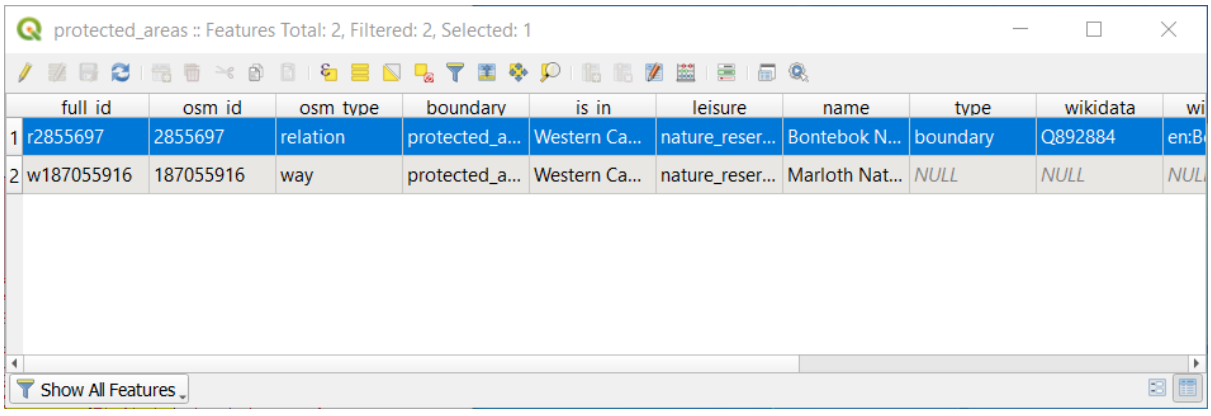
Примечание: Чтобы понять, что представляют собой *поля* и *значения атрибутов*, может потребоваться документация (или метаданные), описывающая смысл значений атрибутов. Обычно она доступна у создателя набора данных.

Далее рассмотрим, как запись в таблице атрибутов связана с полигоном, который мы видим на полотне карты.

1. Вернитесь к главному окну QGIS.
2. На панели *Attributes Toolbar* кликните на кнопку Выбрать функцию  Select Feature.
3. Убедитесь в том, что слой `protected_areas` по-прежнему выбран на панели *Layers*.
4. Переместите мышь на Полотно Карты и кликните левой кнопкой мыши на меньшем из двух полигонов. Полигон станет желтым, указывая на то, что он выбран.

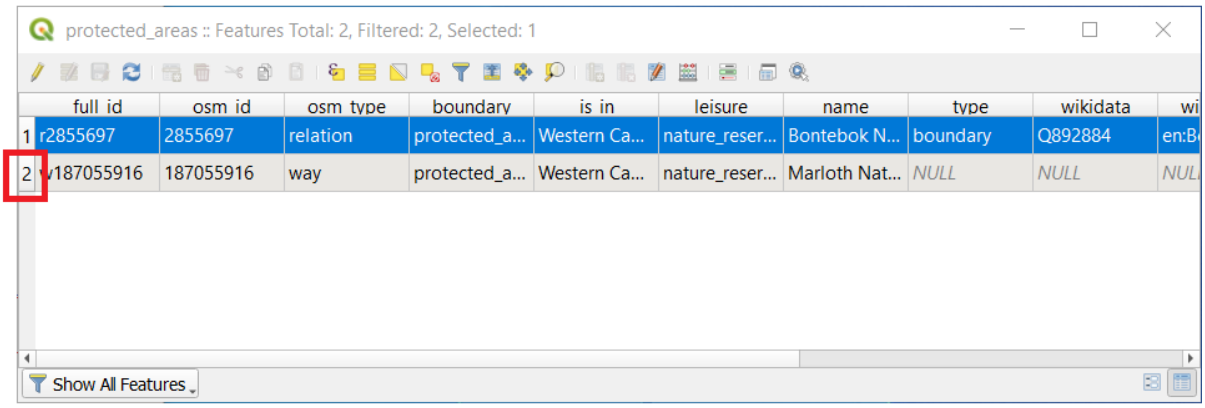


5. Вернитесь к окну Таблица атрибутов *Attribute Table*, вы увидите выделенную запись (строку). Это значения атрибутов выбранного полигона.



Можно также выбрать функцию с помощью Таблицы атрибутов.

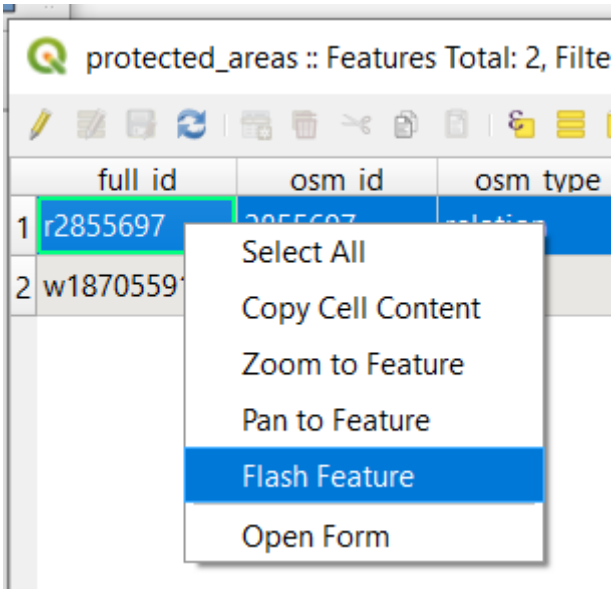
1. В окне *Attribute Table* в крайнем левом углу кликните по номеру строки записи, которая в данный момент не выбрана.



2. Вернитесь к главному окну QGIS и посмотрите на Полотно Карты. Вы должны увидеть больший из двух полигонов, окрашенный в желтый цвет.
3. Чтобы отменить выделение функции, идите к окну *Attribute Table* и кликните на кнопку «Отменить выделение всех функций из слоя» Deselect all features from the layer.

Иногда в полотне карты отображено много функций, и может быть трудно увидеть, какая функция выбрана из Таблицы атрибутов. Другим способом определения местоположения функции является использование инструмента *Flash Feature*.

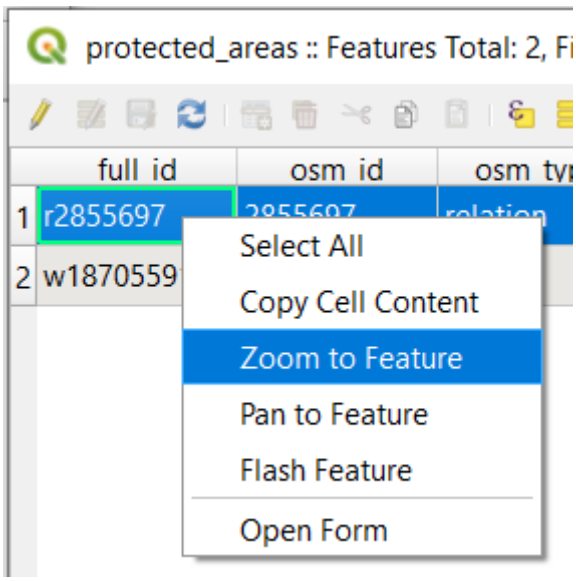
1. В *Attribute Table* кликните правой кнопкой мыши любую ячейку в строке со значением атрибута `r2855697` для поля `full_id`.
2. В контекстном меню выберите *Flash Feature* 1. и просмотрите Полотно Карты.



Вы должны увидеть, как полигон несколько раз мигнет красным. Если вы пропустили мигание, попробуйте еще раз.

Очередной полезный инструмент – это Приблизить до функции *Zoom to Feature* которая сообщает QGIS о необходимости масштабирования до интересующей функции.

1. В Таблице *Attribute Table* кликните правой кнопкой мыши любую ячейку в строке, которая имеет значение атрибута `r2855697` для поля `full_id`.
2. В контекстном меню кликните на *Zoom to Feature*.



Посмотрите на Полотно Карты. Теперь полигон должен занимать всю площадь полотна карты.

Теперь вы можете закрыть таблицу атрибутов.

3.1.2 Попробуйте сами: изучите атрибуты векторных данных

1. Сколько полей доступно в слое *rivers*?
2. Расскажите нам немного о местах Города `town` в вашем наборе данных.
3. Откройте таблицу атрибутов для слоя *places*. Какое поле было бы наиболее полезно представить в форме надписей и почему?

Answer

- There should be 9 fields in the *rivers* layer:
  1. Select the layer in the *Layers* panel.
  2. Right-click and choose *Open Attribute Table*, or press the button on the *Attributes Toolbar*.
  3. Count the number of columns.

  A quicker approach could be to double-click the *rivers* layer, open the *Layer properties* ► *Fields* tab, where you will find a numbered list of the table's fields.
- Information about towns is available in the *places* layer. Open its attribute table as you did with the *rivers* layer: there are two features whose *place* attribute is set to `town`: *Swellendam* and *Buffeljagsrivier*. You can add comment on other fields from these two records, if you like.
- The `name` field is the most useful to show as labels. This is because all its values are unique for every object and are very unlikely to contain *NULL* values. If your data contains some *NULL* values, do not worry as long as most of your places have names.

3.1.3 В заключении

Теперь вы знаете, как использовать таблицу атрибутов, чтобы увидеть, что на самом деле содержится в используемых вами данных. Любой набор данных будет вам полезен только в том случае, если в нем есть интересующие вас атрибуты. Если вы знаете, какой атрибут вам нужен, вы можете быстро решить, можете ли вы использовать данный набор данных или вам нужно найти другой, имеющий необходимые данные атрибутов.

3.1.4 Что дальше?

Разные атрибуты полезны для разных целей. Некоторые из них могут быть отображены непосредственно в виде текста для просмотра пользователем карты. Вы узнаете, как это сделать, на следующем занятии.

3.2 Занятие: Надписи/метки

Можно добавить надписи/метки для отображения любой информации об объекте. Надписи могут быть связаны с любым векторным слоем. Содержания этих надписей зависят от данных атрибутов слоя.

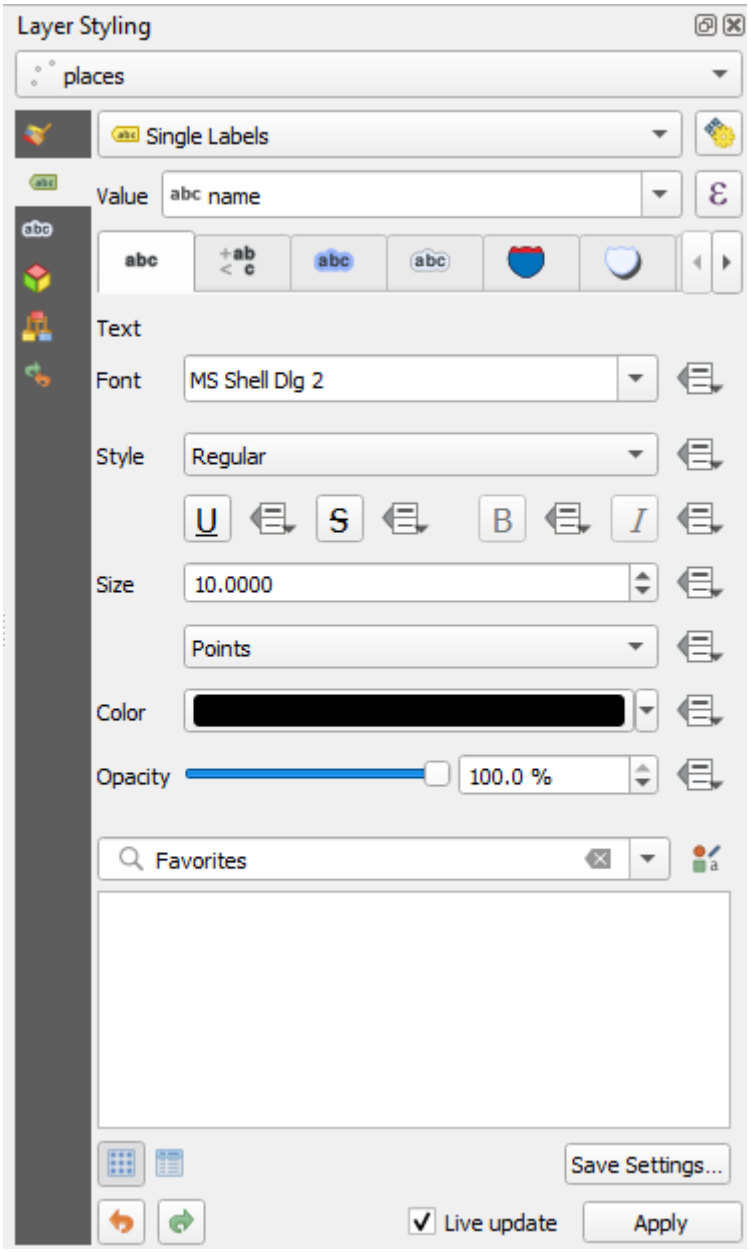
Цель данного занятия: Использовать в слое нужные и красивые надписи/ метки.

3.2.1  Идем дальше: Используем надписи/метки

В первую очередь убедитесь, что вам видна кнопка  в системе GUI:

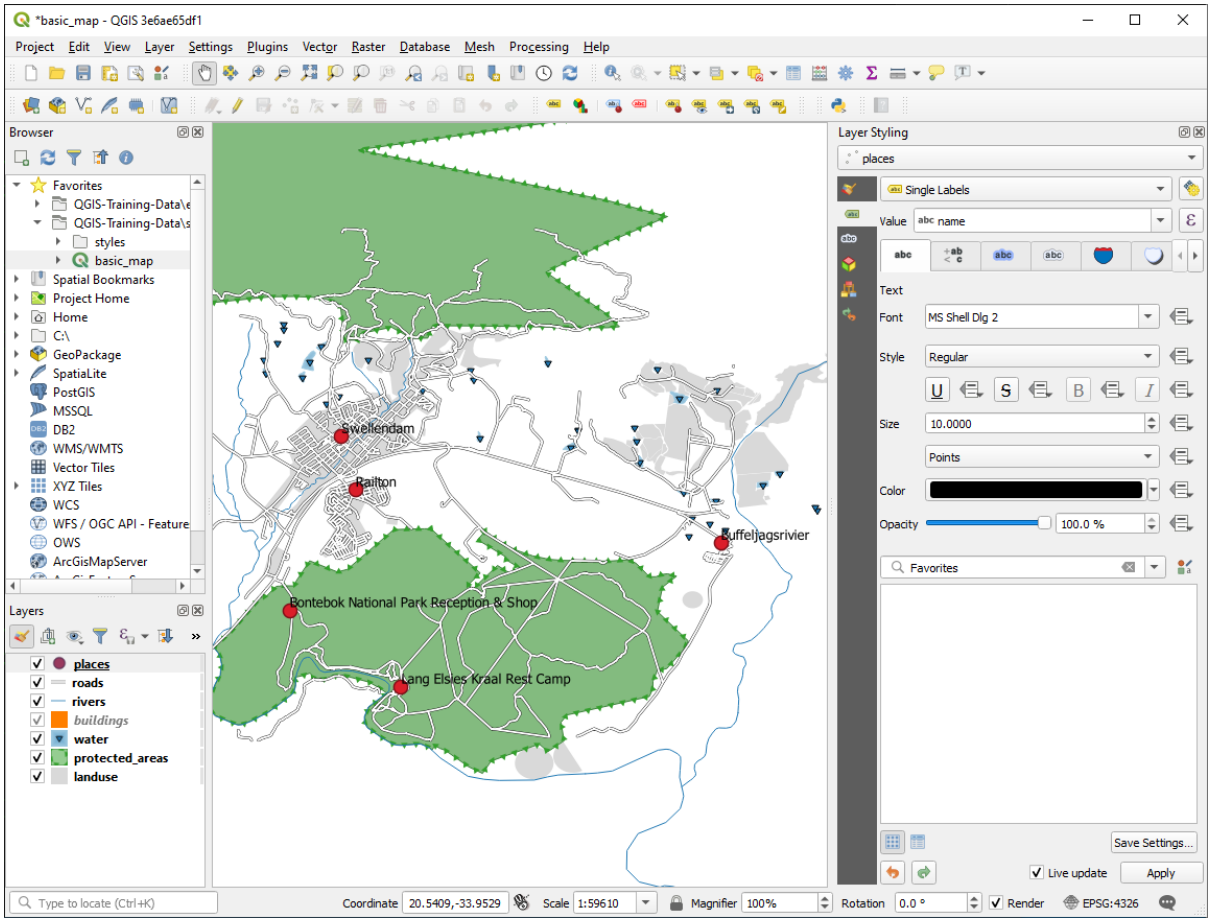
1. Пройдите к элементу меню *View* ► *Toolbars*.
2. Убедитесь, что рядом с панелью инструментов *Label Toolbar* установлен флажок. Если его нет, кликните на *Label Toolbar*, чтобы активировать флажок.
3. Кликните на слой `places` на панели *Layers*, чтобы его выделить.
4. Кликните на кнопку на панели инструментов , чтобы открыть вкладку *Labels* на панели Стиль слоя *Layer Styling*.
5. Переключитесь с *No Labels* на  *Single Labels* (Одиночные надписи/метки).

   Вам нужно будет выбрать, какое поле в атрибутах будет использоваться для надписей. На предыдущем занятии вы решили, что поле Имя `name` наиболее подходит для данной цели.
6. Выберите `name` из списка Значения:



7. Кликните Применить *Apply*.

На карте должны появиться надписи/ метки следующим образом:



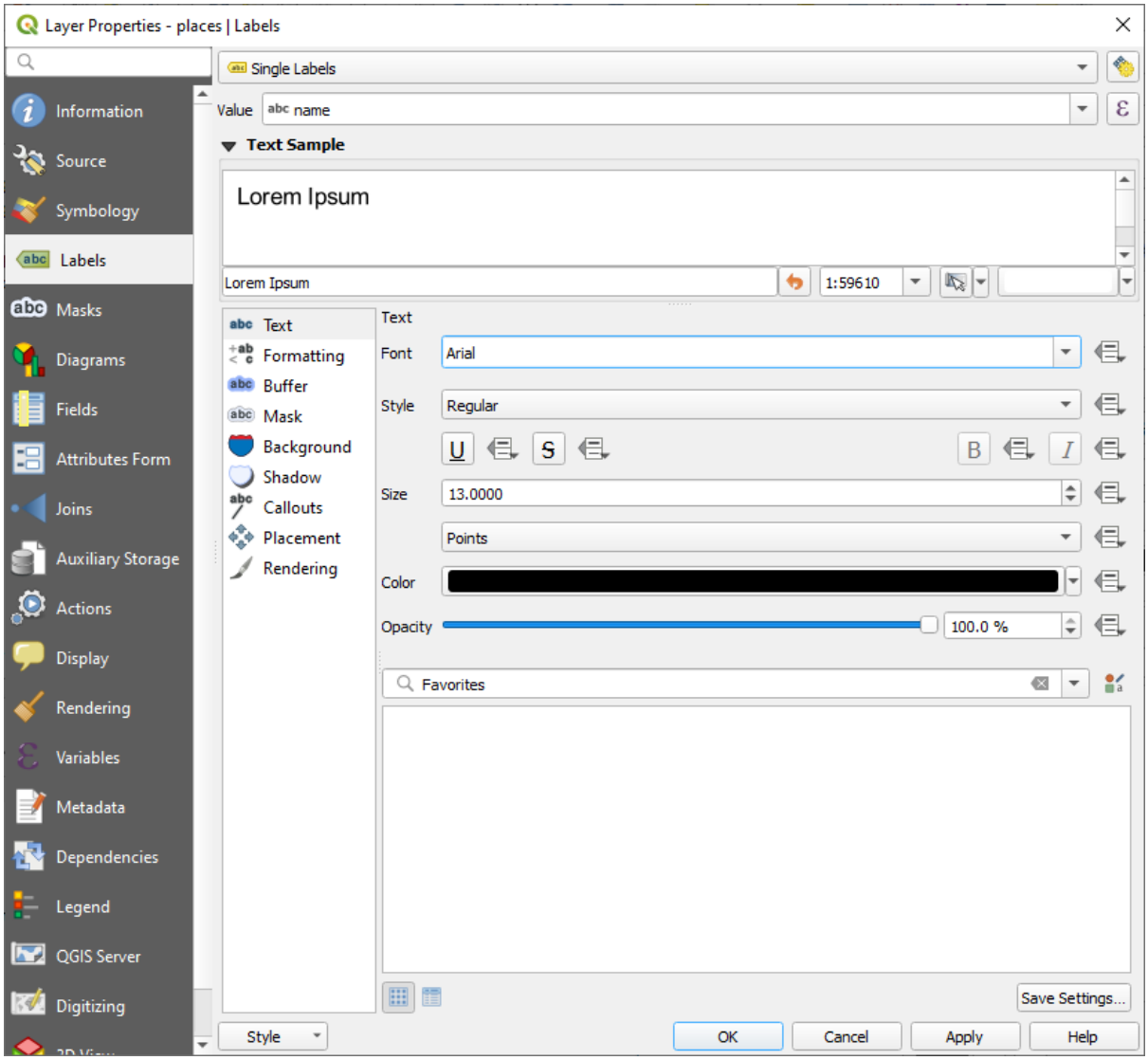
3.2.2 Идем дальше: Меняем параметры надписей

В зависимости от стилей, которые вы выбрали для своей карты на предыдущих занятиях, вы можете обнаружить, что надписи/ метки имеют неправильный формат и они либо перекрываются, либо находятся слишком далеко от своих точечных маркеров.

Примечание: Ранее, вы применяли кнопку abc на панели *Label Toolbar*, чтобы открыть панель Стили слоя *Layer Styling*. Также, как и с Символикой *Symbology*, те же параметры надписей доступны через панель *Layer Styling* и через диалоговое окно Свойства слоя *Layer Properties*. Здесь вы будете использовать диалоговое окно *Layer Properties*.

---

1. Откройте диалоговое окно *Layer Properties* двойным нажатием на слой `places`.
2. Выберите вкладку abc *Labels*.
3. Убедитесь, что выбран *Text* в списке параметров слева, затем обновите параметры форматирования текста, чтобы они соответствовали показанным здесь:

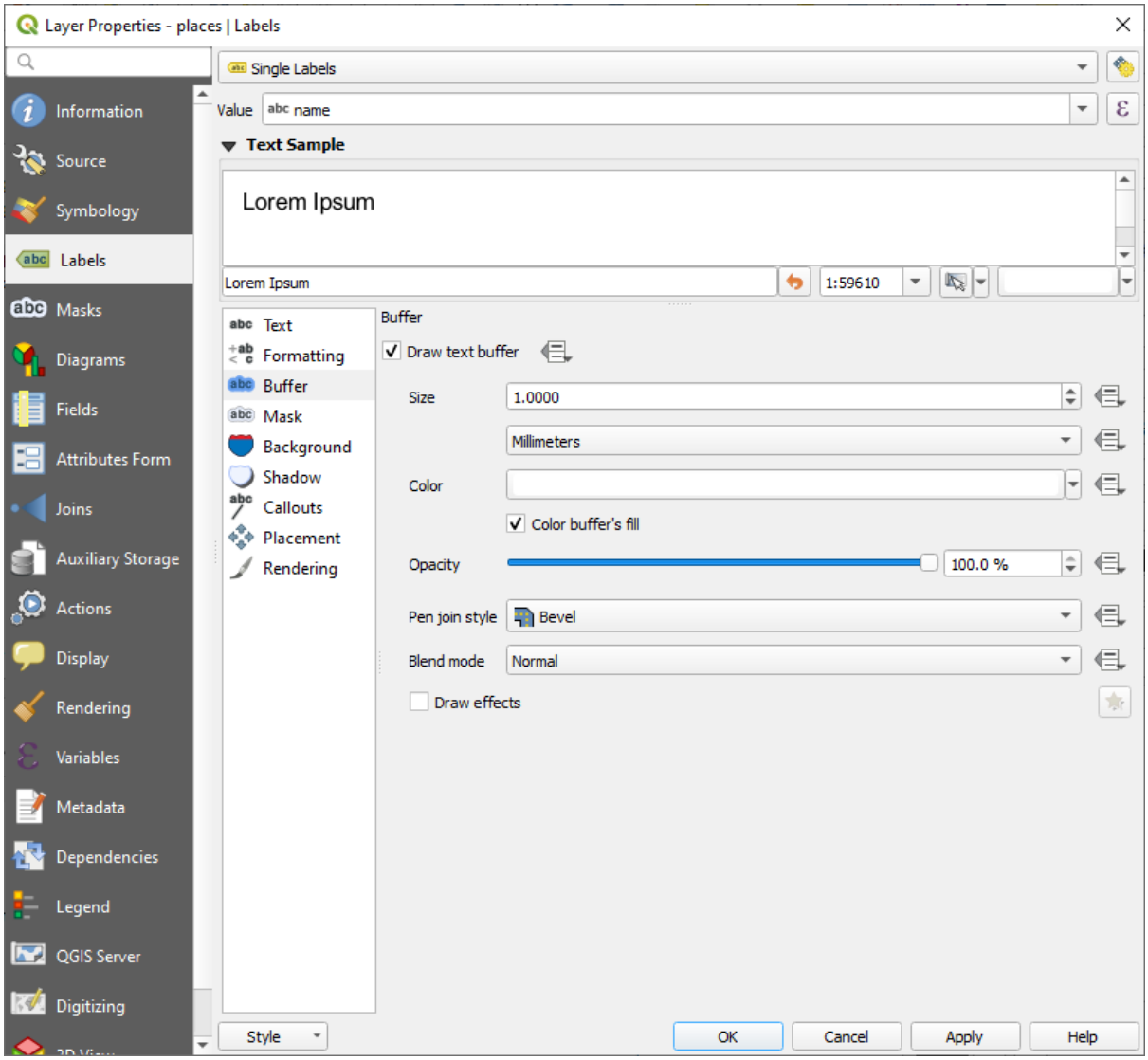


4. Кликните Применить *Apply*.

   Такой шрифт может быть крупнее и более знакомым пользователям, но его читабельность по-прежнему зависит от того, какие слои отображаются под ним. Чтобы решить эту проблему, давайте взглянем на опцию Буфер *Buffer*.

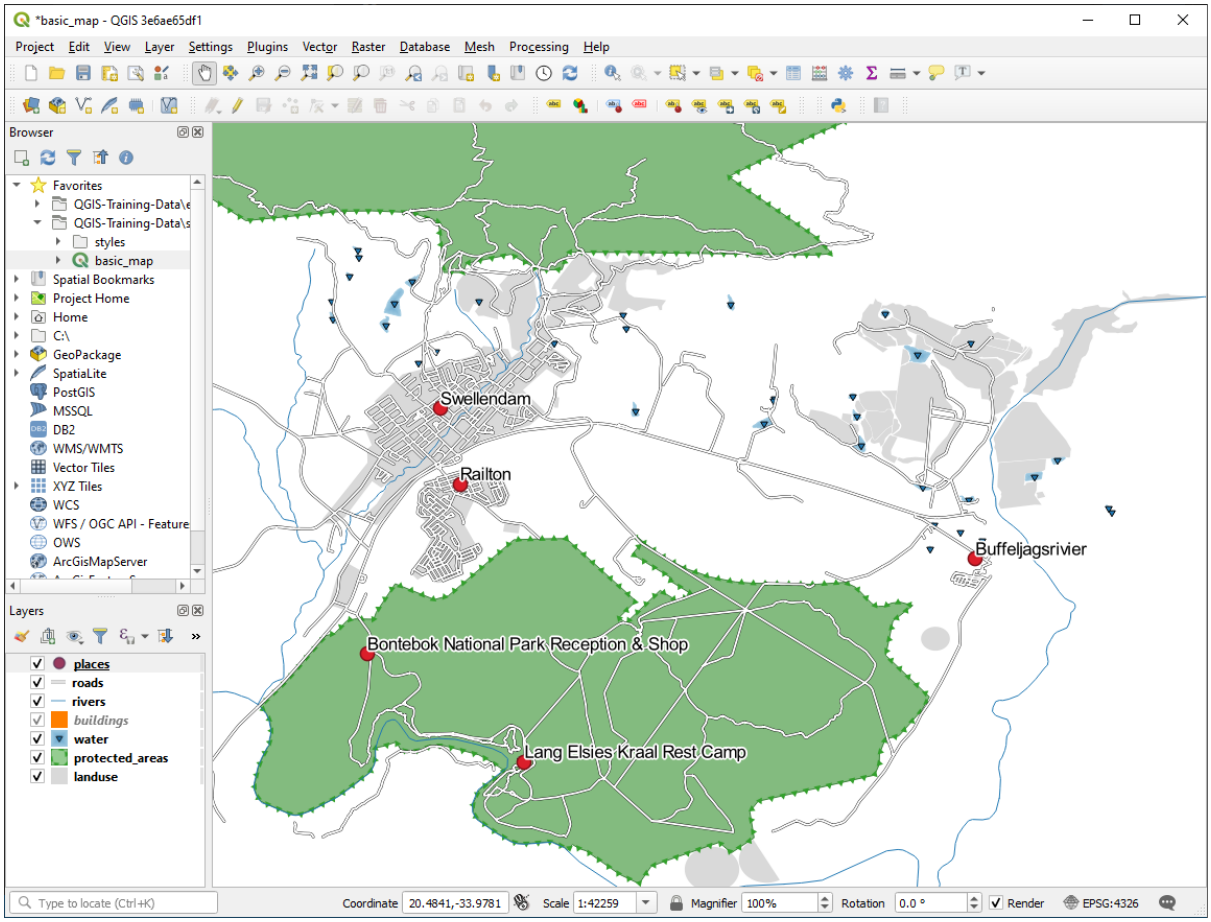
5. Выберите *Buffer* из списка параметров слева.

6. Установите флажок рядом с надписью Нарисовать текстовый буфер *Draw text buffer*, затем выберите параметры, соответствующие показанным здесь:



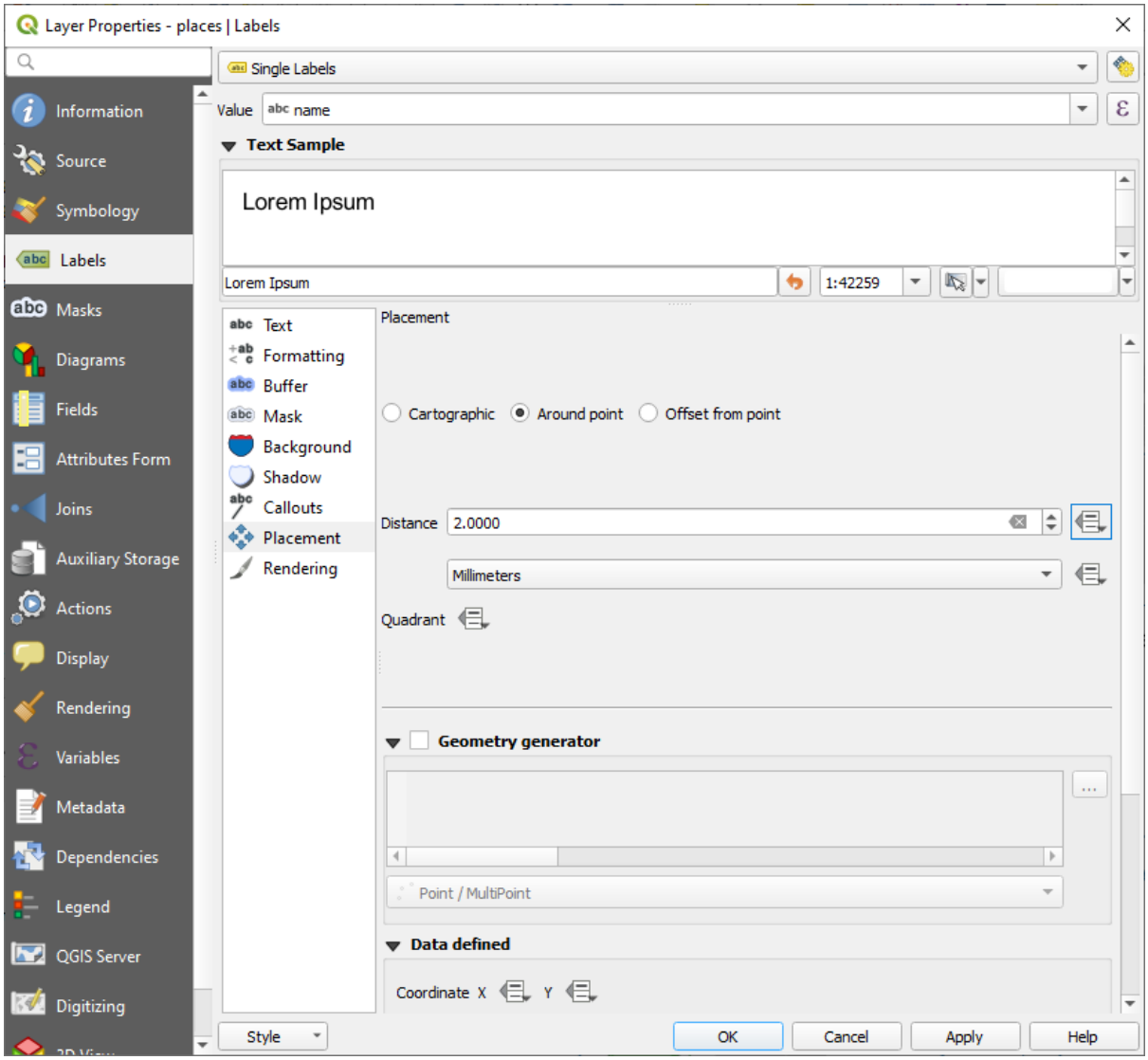
7. Кликните Применить *Apply*.

   Вы увидите, что появляется цветной буфер или граница к надписям мест, что упрощает их выбор на карте:



Теперь мы можем заняться позиционированием надписей относительно их точечных маркеров.

8. Выберите Разместить *Placement* из списка параметров слева.
9. Выберите Вокруг точки *Around point* и измените значение Расстояние *Distance* на `2.0` миллиметра:



10. Кликните Применить *Apply*.

    Вы увидите, что надписи/ метки больше не перекрывают свои точечные маркеры.

3.2.3 Идем дальше: Применяем надписи/метки вместо слоя символики

Во многих случаях расположение точки не обязательно должно быть очень конкретным. Например, большинство точек в слое `places` относятся к целым городам или пригородам, а конкретная точка, связанная с такими функциями, не является такой уж конкретной в больших масштабах. На самом деле, когда задается слишком конкретная точка, она часто сбивает с толку того, кто читает карту.

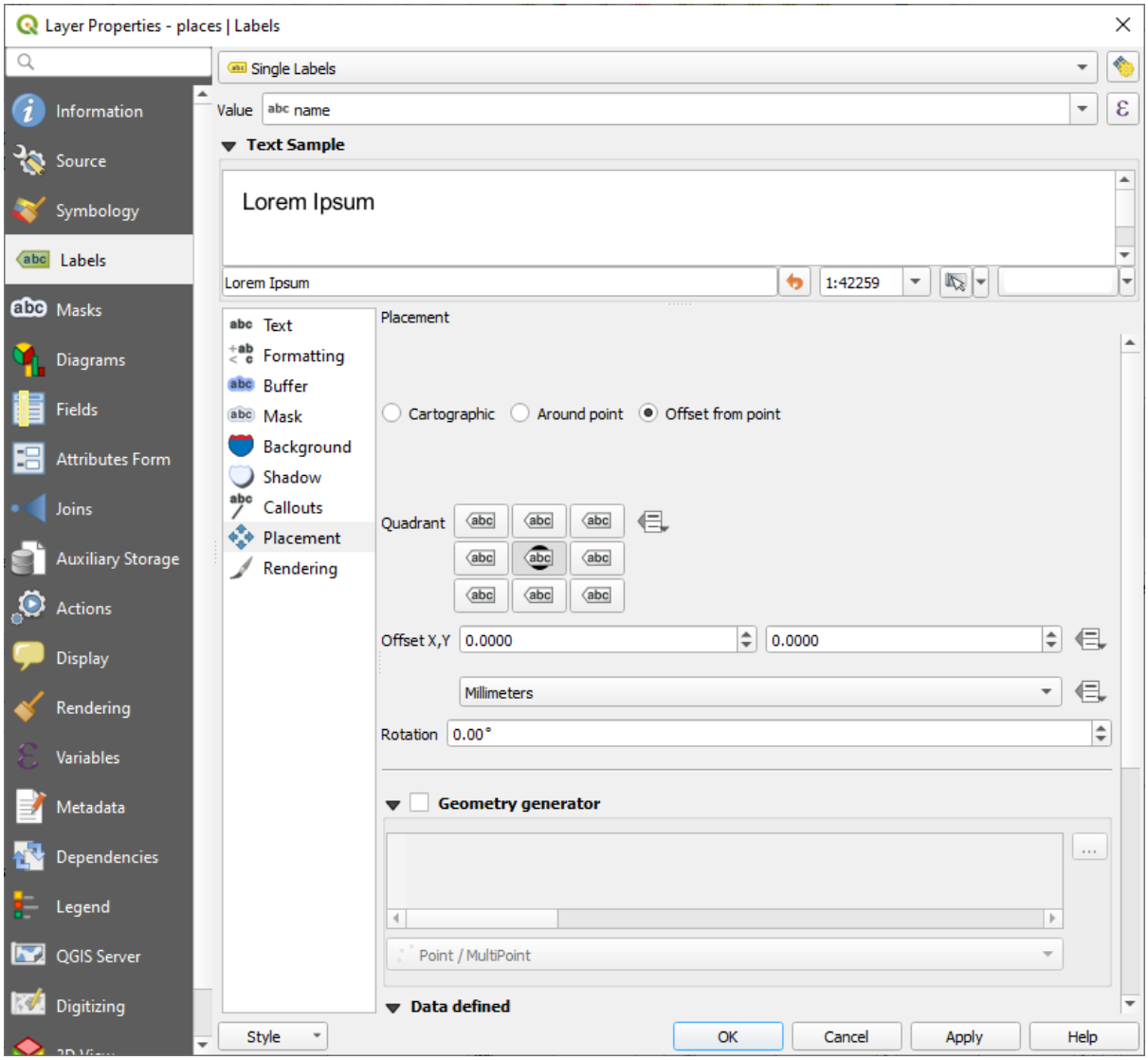
Например, на карте мира точка, заданная как Европейский Союз, может быть где-то в Польше. Человеку, который «читает» карту и видит точку с надписью Европейский Союз на Польше, может показаться, что столица *Европейского Союза*, таким образом, находится в Польше.

Поэтому, для предотвращения такого рода недоразумений, часто полезно деактивировать символы точек и полностью заменить их надписями.

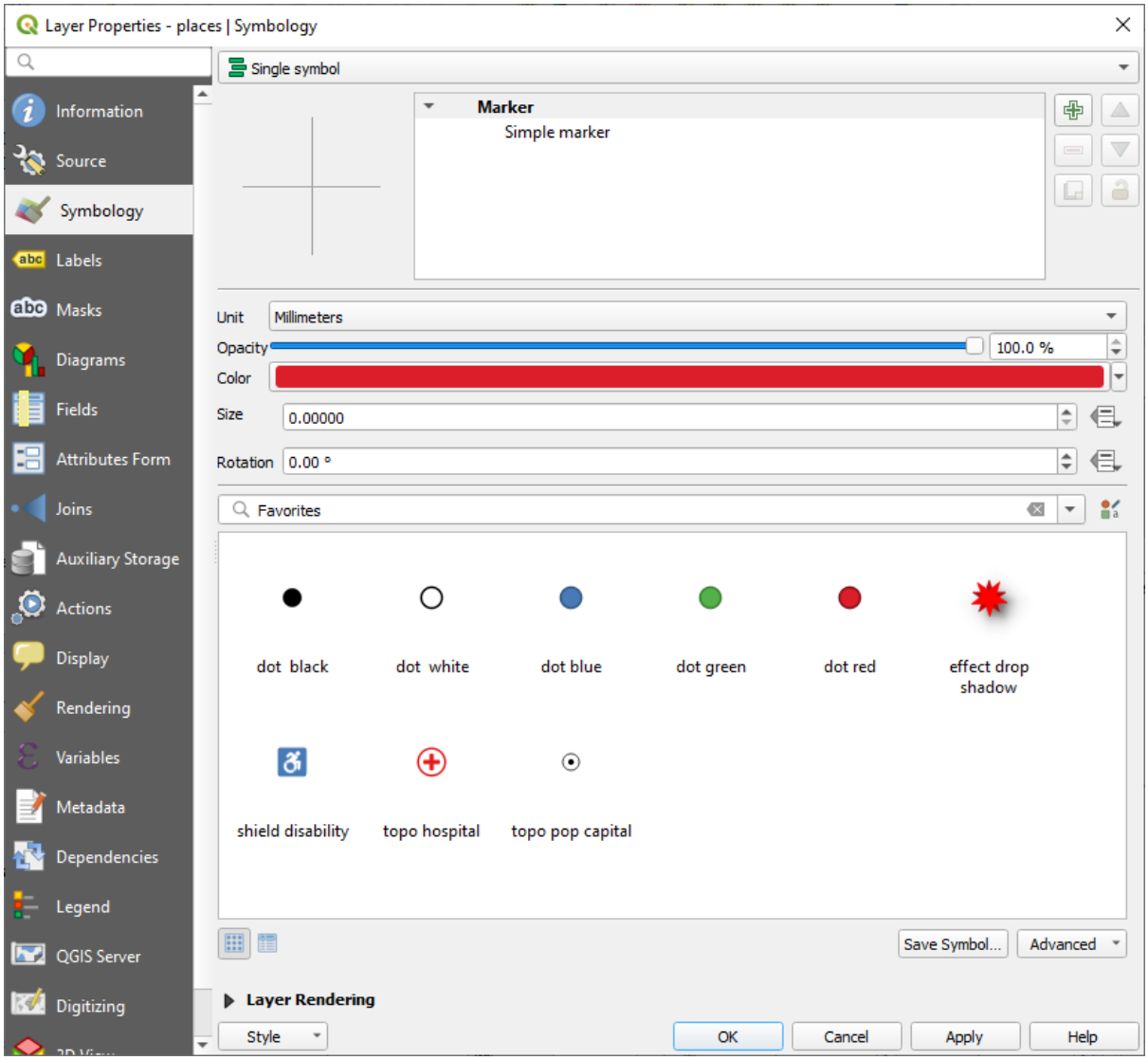
В QGIS это можно сделать, изменив положение надписей для их визуализации непосредственно над точками, на которые они ссылаются.

1. Откройте вкладку *Labels* в диалоговом окне *Layer Properties* в слое `places`.
2. Выберите *Placement* из списка параметров.
3. Кликните на кнопку Смещение от точки *Offset from point*.

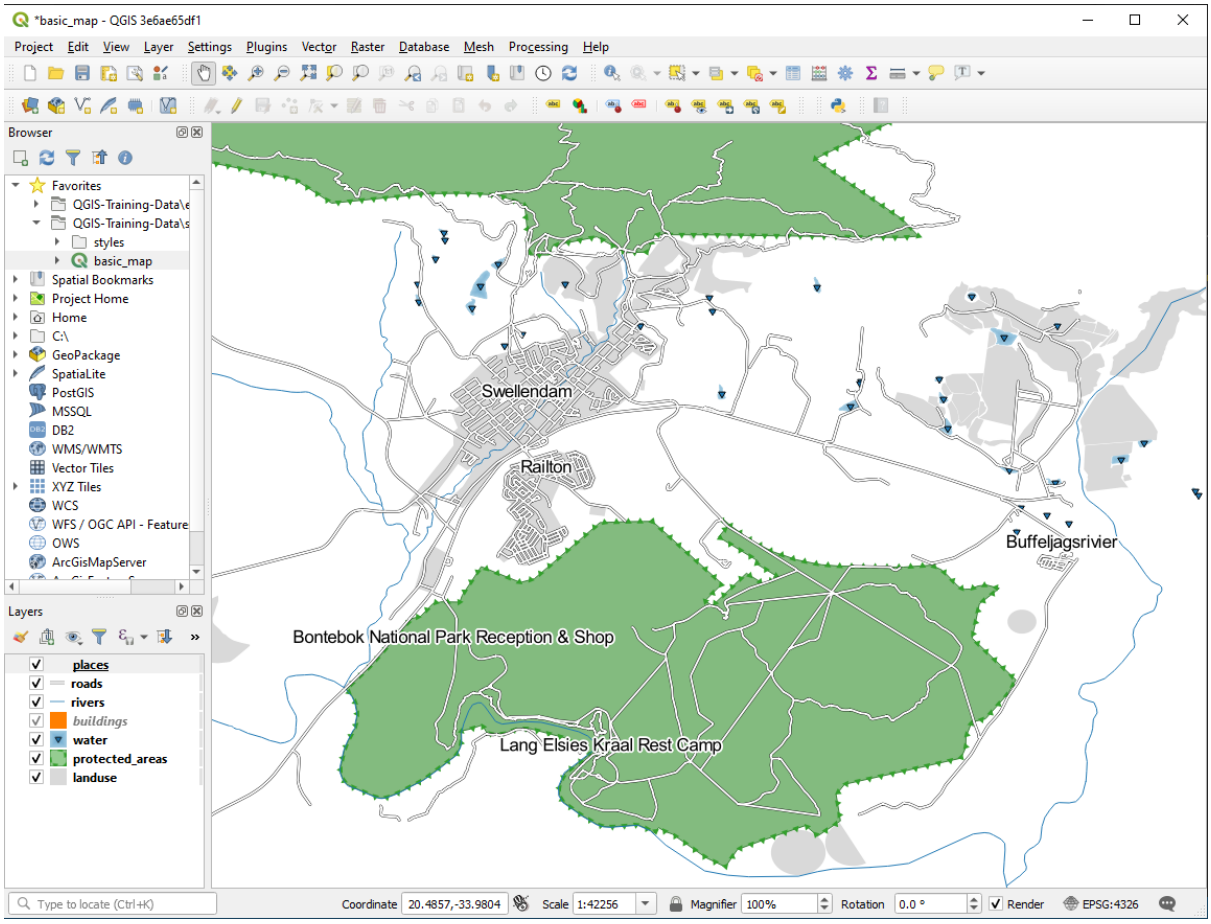
   Это откроет параметры Квадранта *Quadrant*, которые вы можете использовать для установки положения надписи/ метки относительно маркера точки. В данном случае мы хотим, чтобы надпись располагалась по центру точки, поэтому выберите центральный квадрант:



4. Скройте символы точки чрез редактирование слоя *Symbology* как обычно, и задайте размер Маркера *Marker* до `0.0`:



5. Кликните *Apply* и вы увидите следующий результат:



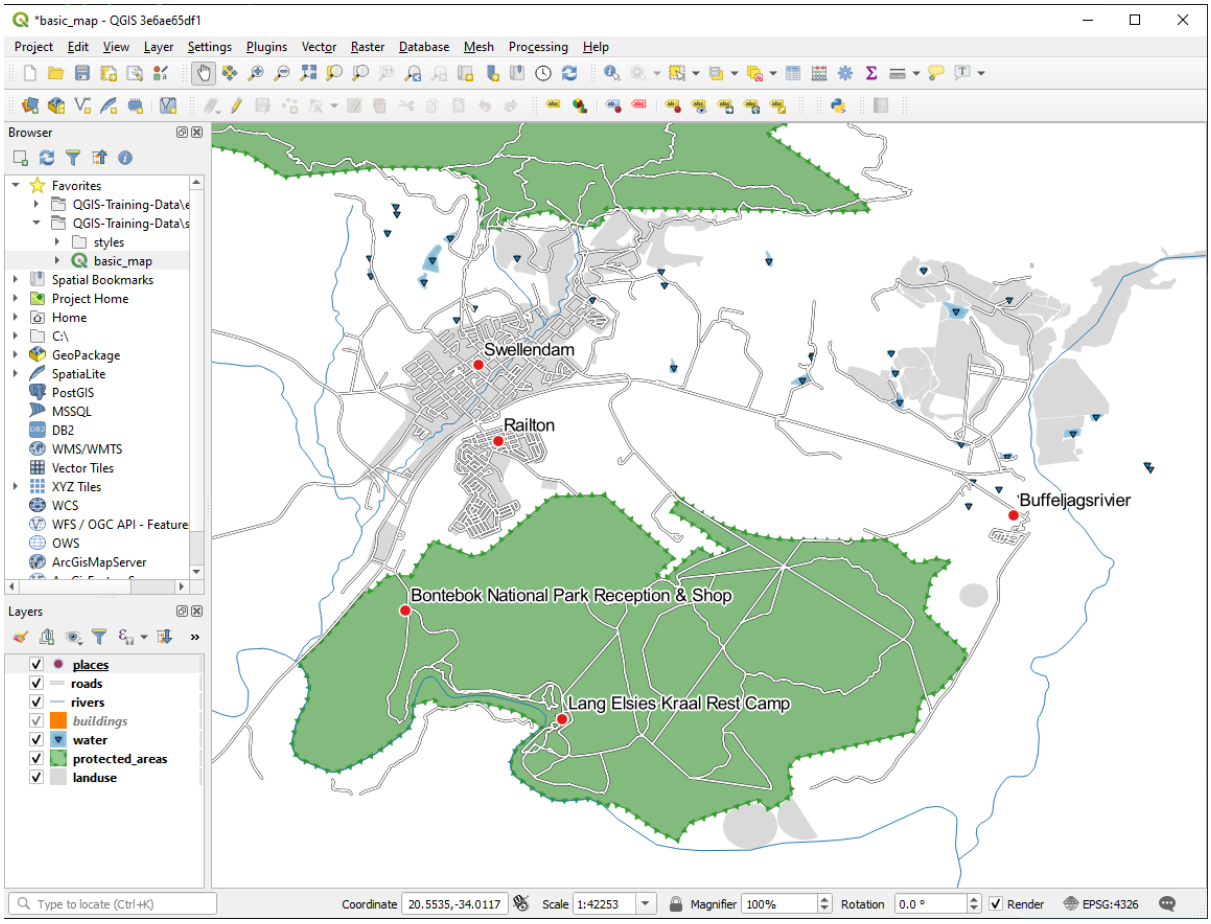
Если бы вы отдалили карту, то вы бы увидели, что некоторые надписи исчезают при большем масштабе, чтобы избежать перекрывания. Иногда это именно то, что вам нужно, когда вы имеете дело с наборами данных, которые содержат много точек, но в других случаях, вы потеряете полезную информацию. Есть еще одна возможность для работы с подобными случаями, о которых мы поговорим позже на этом занятии. А пока отдалите карту, кликните кнопку  на панели инструментов и посмотрите, что произойдет.

3.2.4  Попробуйте сами: Настройте надписи

- Сбросьте настройки надписи /метки и символа, чтобы получить точечный маркер и смещение надписи до `2.0` миллиметров.

Answer

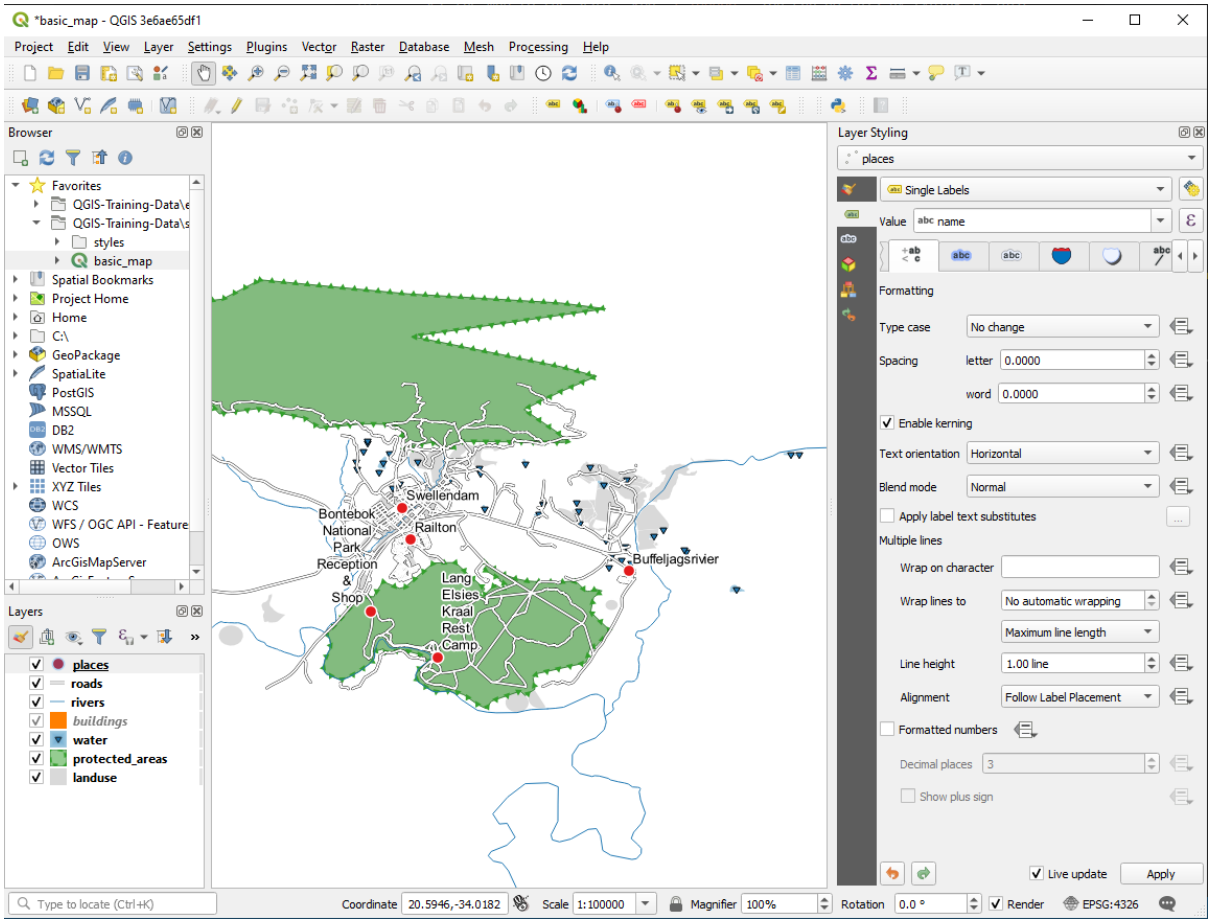
Your map should now show the marker points and the labels should be offset by 2mm. The style of the markers and labels should allow both to be clearly visible on the map:



- Установите масштаб карты `1:100000`. Вы можете сделать это через ввод в поле Масштаб *Scale* в *Status Bar*. Измените свои надписи, чтобы они подходили для просмотра в данном масштабе.

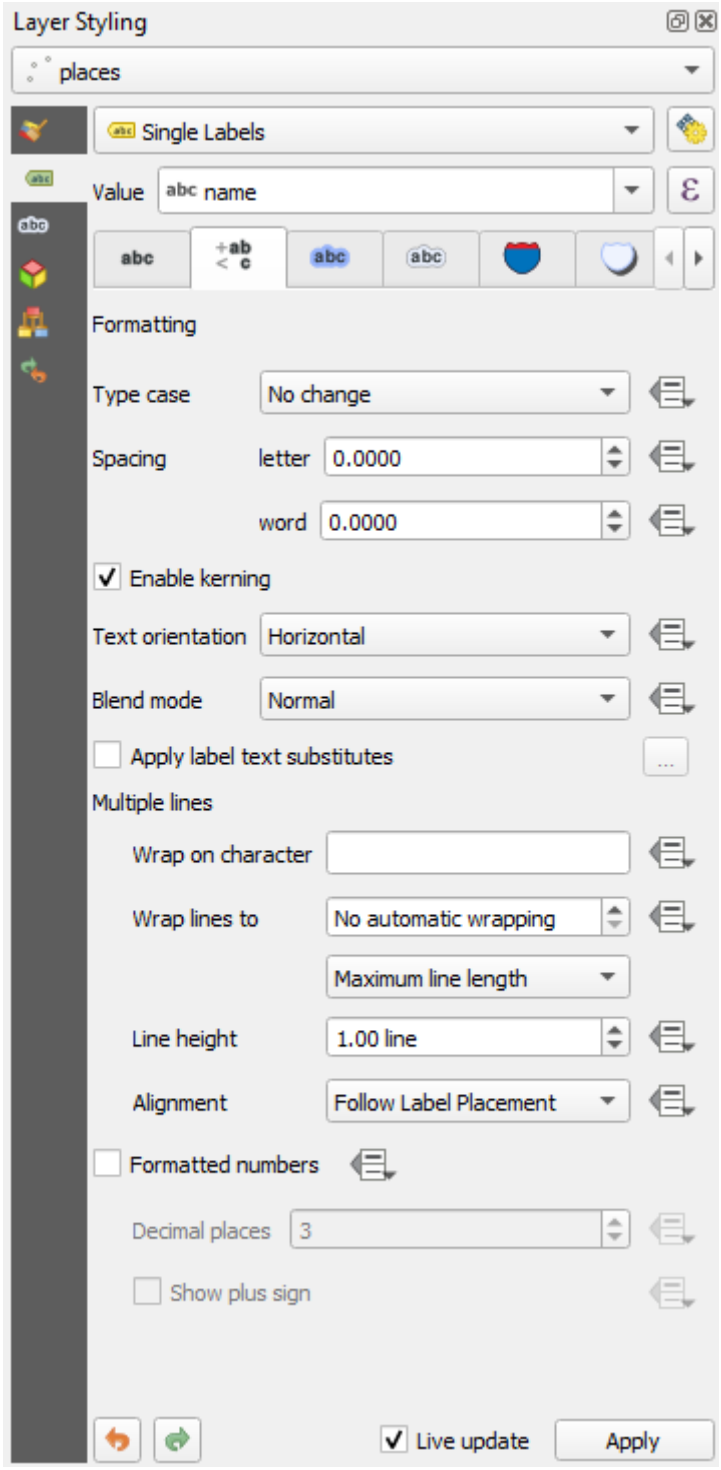
Answer

One possible solution has this final product:



To arrive at this result:

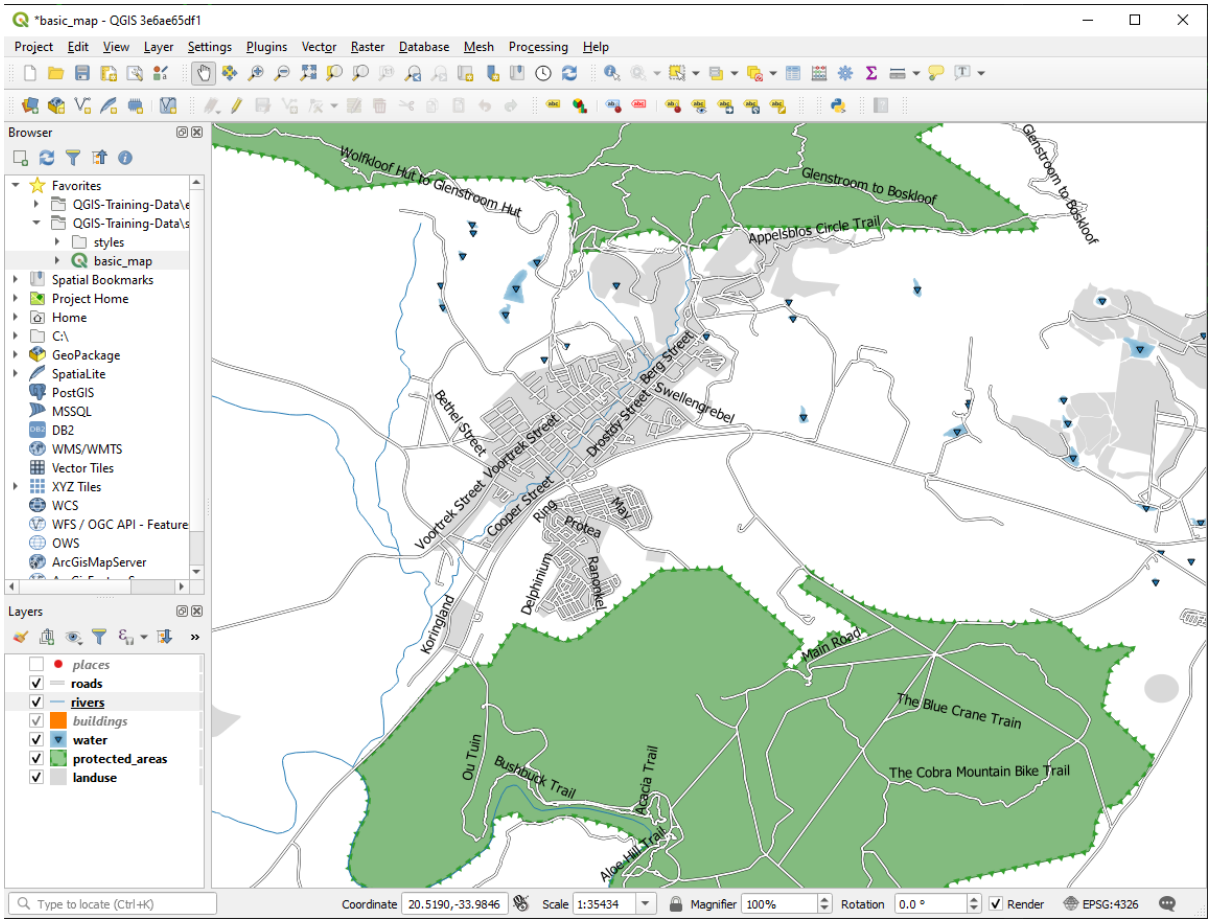
- Use a font size of `10`
- Use an around point placement distance of `1.5 mm`
- Use a marker size of `3.0 mm`
- In addition, this example uses the *Wrap on character* option:



- Enter a `space` in this field and click *Apply* to achieve the same effect. In our case, some of the place names are very long, resulting in names with multiple lines which is not very user friendly. You might find this setting to be more appropriate for your map.

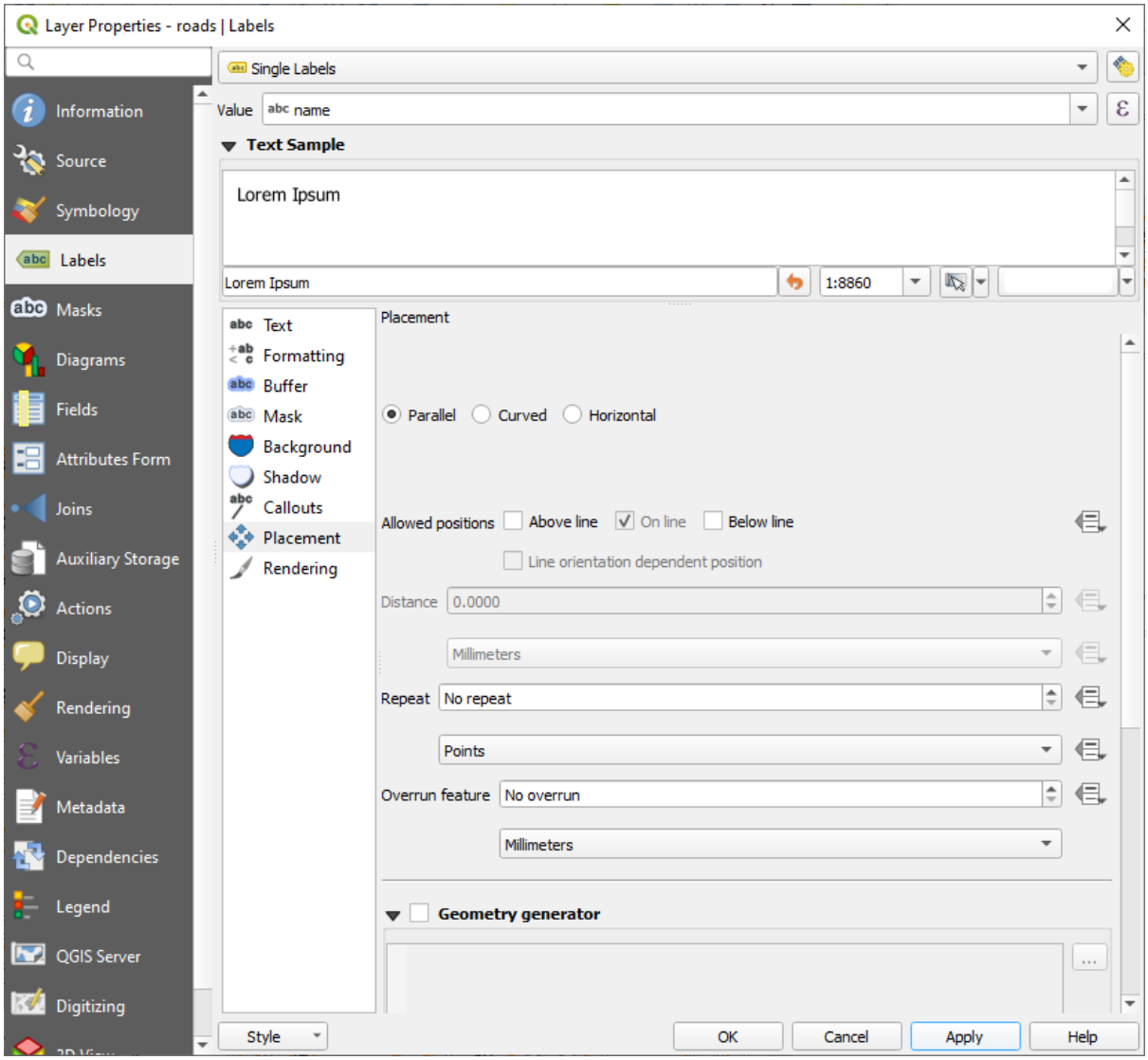
3.2.5 Идем дальше: Надписываем линии

Теперь, когда вы знаете, как работает надпись/ метка, возникает дополнительная проблема. Точки и полигоны легко надписать, но как насчет линий? Если вы надписываете их так же как точки, то ваши результаты будут выглядеть так:



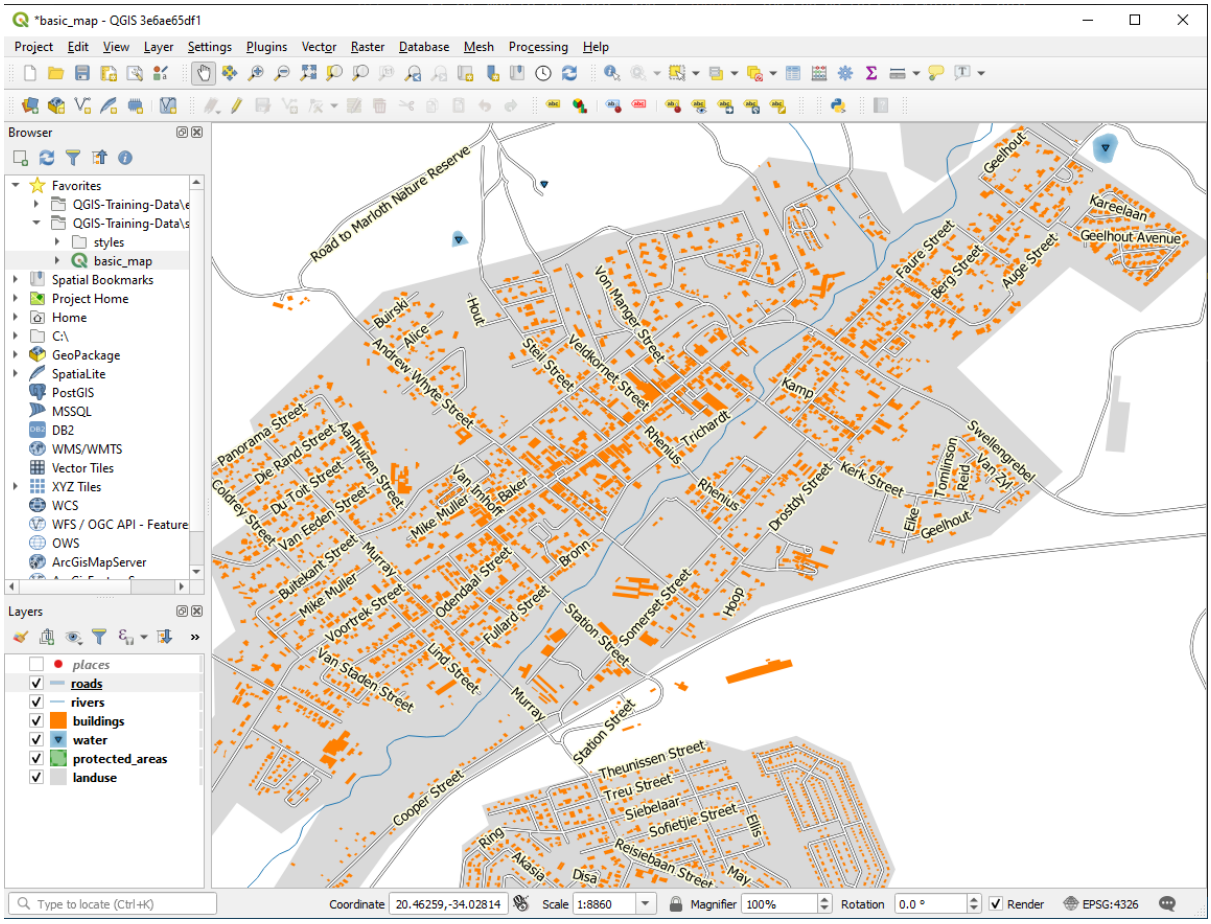
Теперь мы переформатируем надписи слоя `roads`, чтобы их было легче понять.

1. Скройте слой `places`, чтобы он вас не отвлекал.
2. Активируйте *Single Labels* для слоя `roads`, как вы делали выше для слоя Места `places`.
3. Установите Размер *Size* шрифта на `10`, чтобы вы могли видеть больше надписей.
4. Приблизить местность города Свеллендам Swellendam.
5. Во вкладке *Labels* есть вкладка Размещение *Placement*, далее выберите следующие настройки:



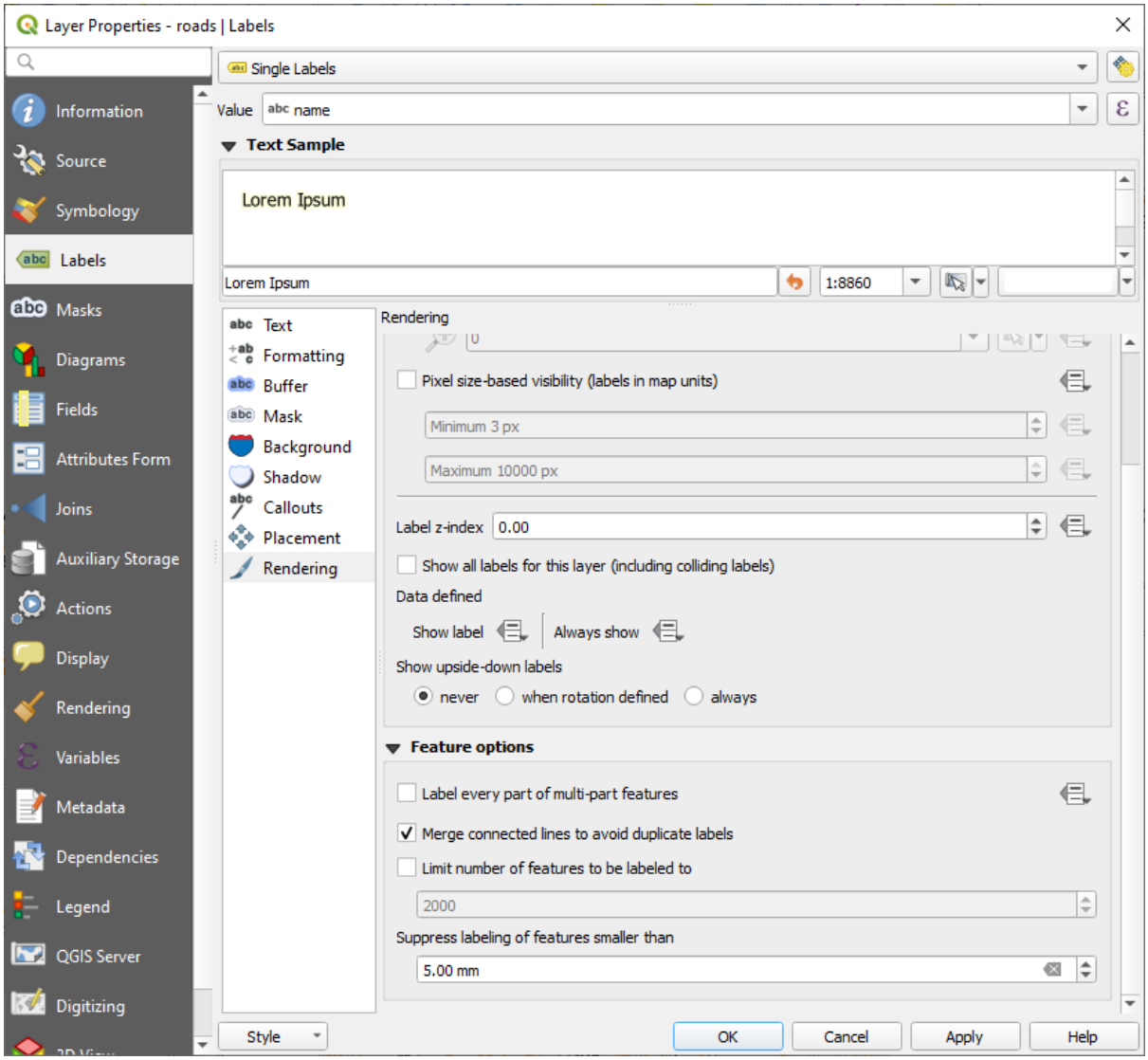
Вы, вероятно, обнаружите, что для стиля текста используются значения по умолчанию, и поэтому надписи очень трудно читать. Обновите *Text*, чтобы использовать темно-серый или черный цвет *Color*, и *Buffer*, чтобы использовать светло-желтый цвет *Color*.

Карта будет выглядеть примерно так, в зависимости от масштаба:



Вы увидите, что названия некоторых дорог появляются более одного раза, и это не всегда необходимо. Чтобы этого не произошло:

6. Во вкладке *Labels* есть диалоговое окно *Layer Properties*, выберите параметр Визуализация *Rendering* и выберите «Объединить соединенные линии, чтобы избежать дублирования надписей» *Merge connected lines to avoid duplicate labels*, как показано ниже:



7. Кликните *OK*.

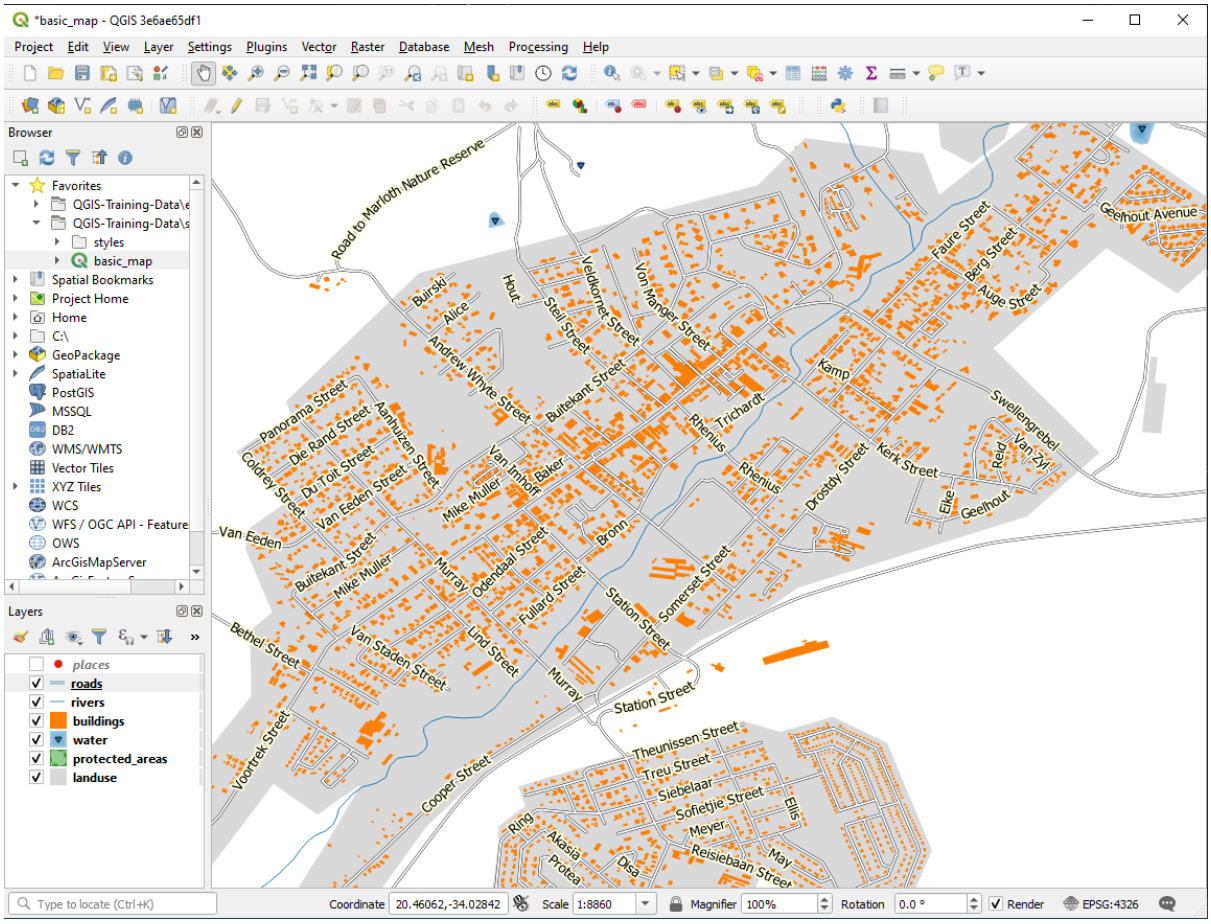
   Еще одна полезная функция –это то, что можно предотвратить рисование надписей для функций, слишком короткими, чтобы их можно было заметить.

8. На той же панели Визуализация *Rendering* установите значение Подавить маркировку функций менее чем… *Suppress labeling of features smaller than*… `5.00 mm` и обратите внимание на результаты, когда вы нажмете кнопку *Apply*.

   Попробуйте также другие настройки места *Placement*. Как мы видели ранее, вариант По горизонтали *Horizontal* в данном случае не лучший вариант, поэтому давайте попробуем вместо этого вариант Изогнутый *Curved*.

9. Выберите параметр Изогнутый *Curved* на панели Размещение *Placement* во вкладке Надписи *Labels*.

Вот результат:

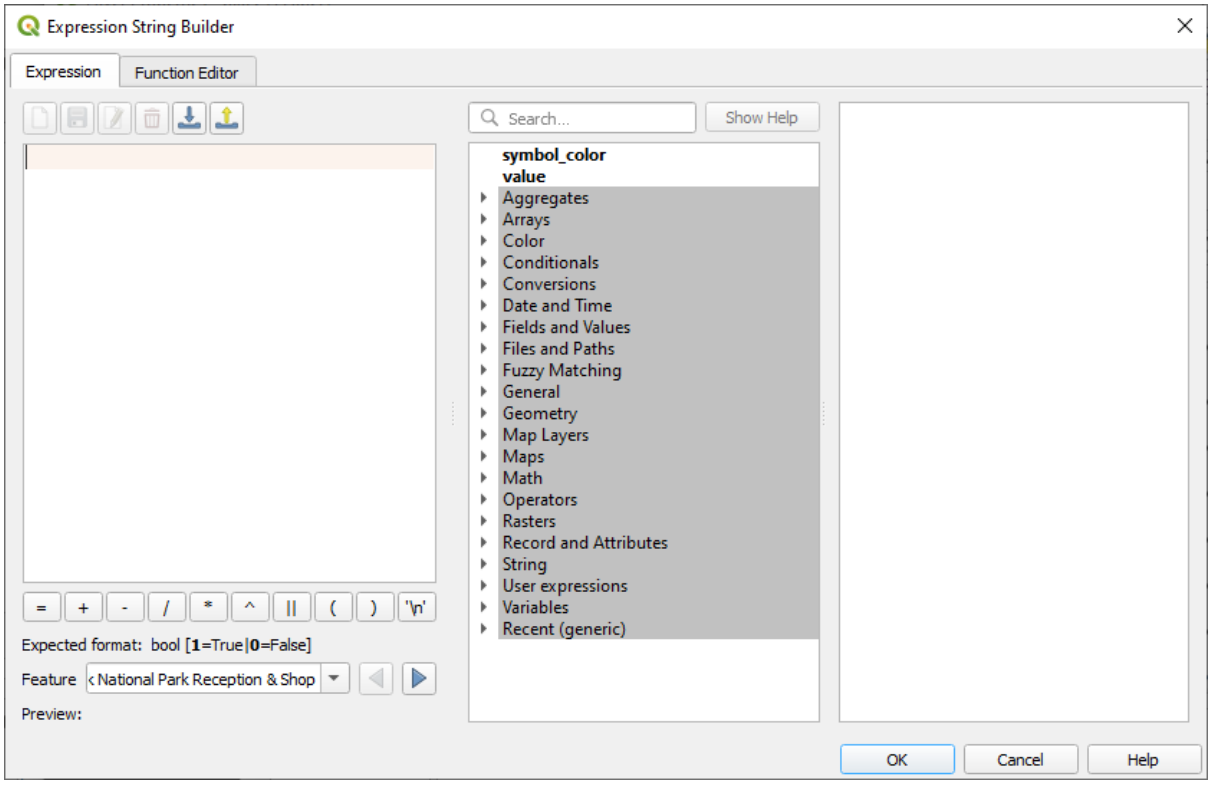


Как видите, такое действие скрывает некоторые надписи, которые были ранее видны, из-за сложности заставить некоторые из них следовать извилистым линиям улиц, оставаясь при этом читаемыми. Это делает другие надписи гораздо более полезными, поскольку они отслеживают дороги, а не плавают в пространстве между ними. Вы можете решить, какой из этих вариантов применить, в зависимости от того, что, по вашему мнению, кажется более полезным, или что выглядит лучше.

3.2.6 Идем дальше: Настраиваем параметры данных

1. Отключите надпись слоя `roads`.
2. Повторно активируйте надпись слоя `places`.
3. Откройте таблицу атрибутов `places` с помощью кнопки .

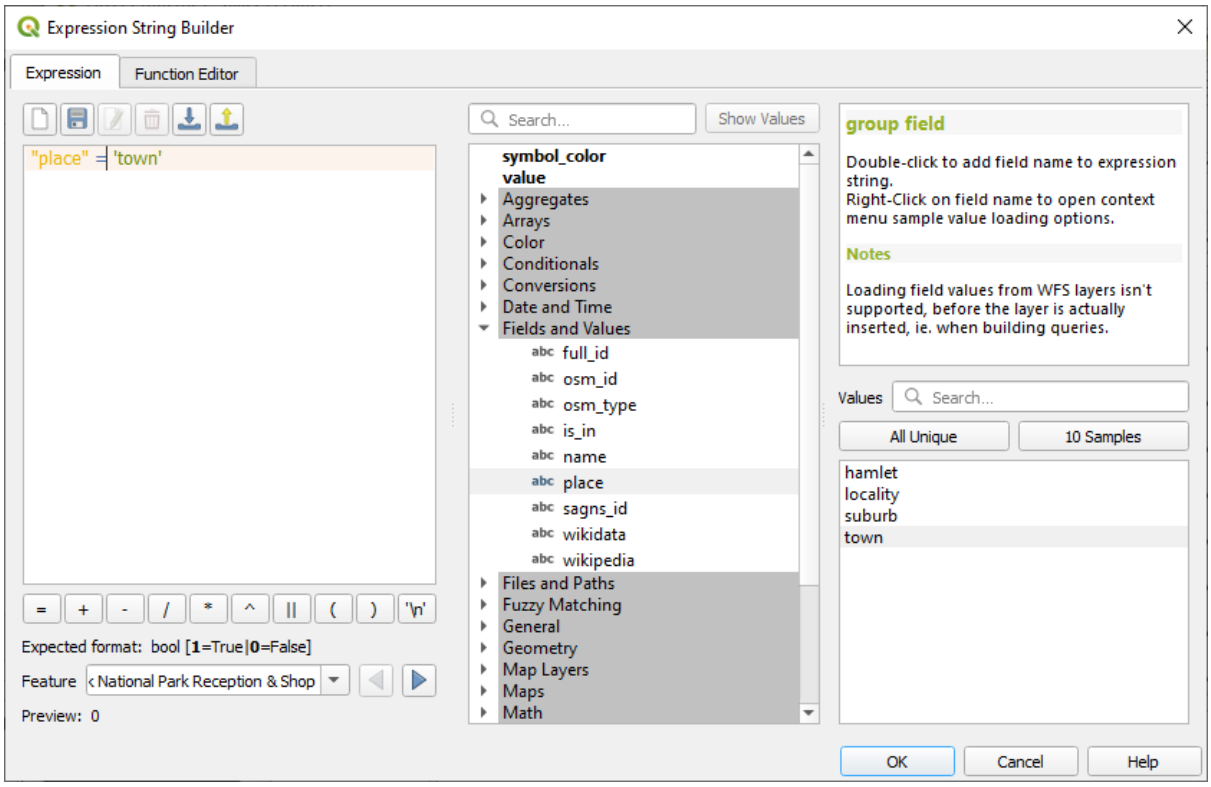
   В таблице есть одно *поле*, которое нас сейчас интересует: это Место `place`, оно определяет тип городской местности для каждой *записи*. Мы можем использовать эти данные, чтобы повлиять на стили надписей.
4. Перейдите на панель *Text* на панели `places` *Labels*.
5. Кликните на кнопку рядом с кнопкой курсивного текста под надписью Стиль *Style* и выберите Редактировать… *Edit…*, чтобы открыть Конструктор строк выражений *Expression String Builder*:



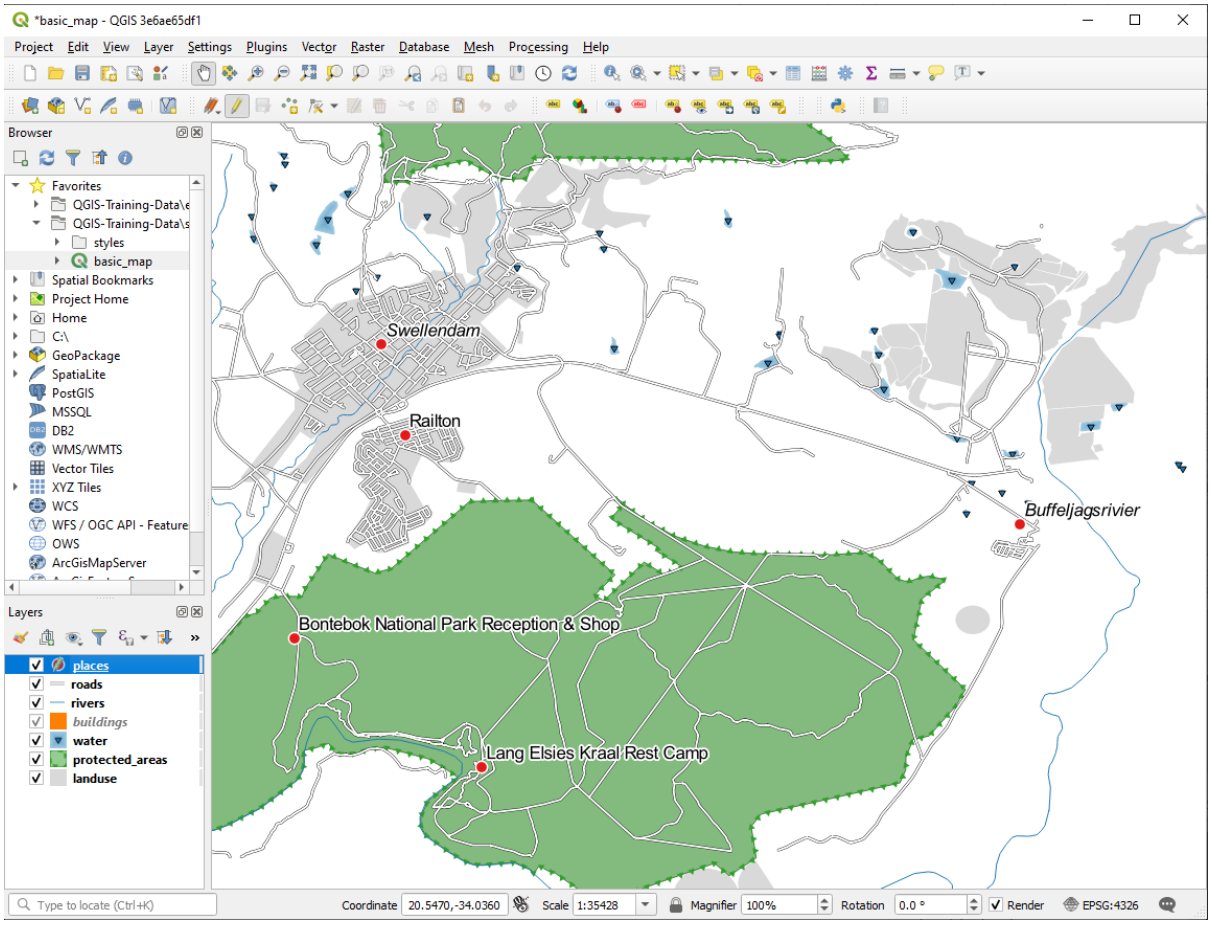
6. В поле *Fields and Values* дважды кликните на `place`, а затем кликните Все уникальные *All Unique*. Откроется список всех уникальных значений поля `place` данного слоя. Добавьте = в текстовом редакторе, а затем дважды кликните `town`.

   Кроме того, вы можете ввести: «место=город» / `"place" = 'town'` прямо в текстовом редакторе.

7. Кликните дважды *OK*:



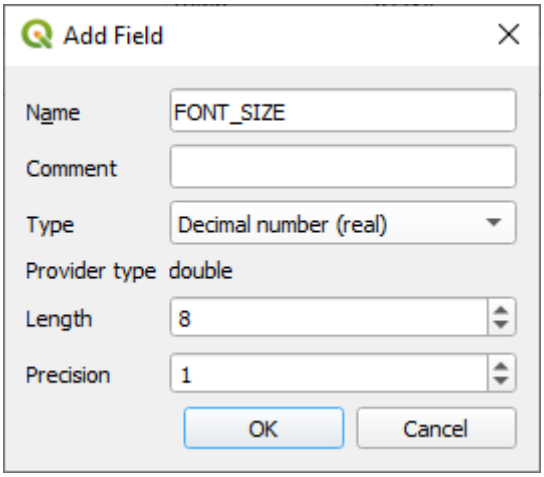
Обратите внимание, что надписи для всех мест, в которых поле `place` соответствует `town`, отображаются курсивом.



3.2.7 Попробуйте сами: Примените настройку параметров данных

Примечание: Мы немного забегаем вперед, чтобы продемонстрировать некоторые более продвинутые настройки надписей. Предполагается, что если вы на продвинутом уровне, вы знаете что нижеследующее означает. Если вы не сможете это сделать, пропускайте данный раздел и вернитесь позже, когда вы изучите необходимые материалы.

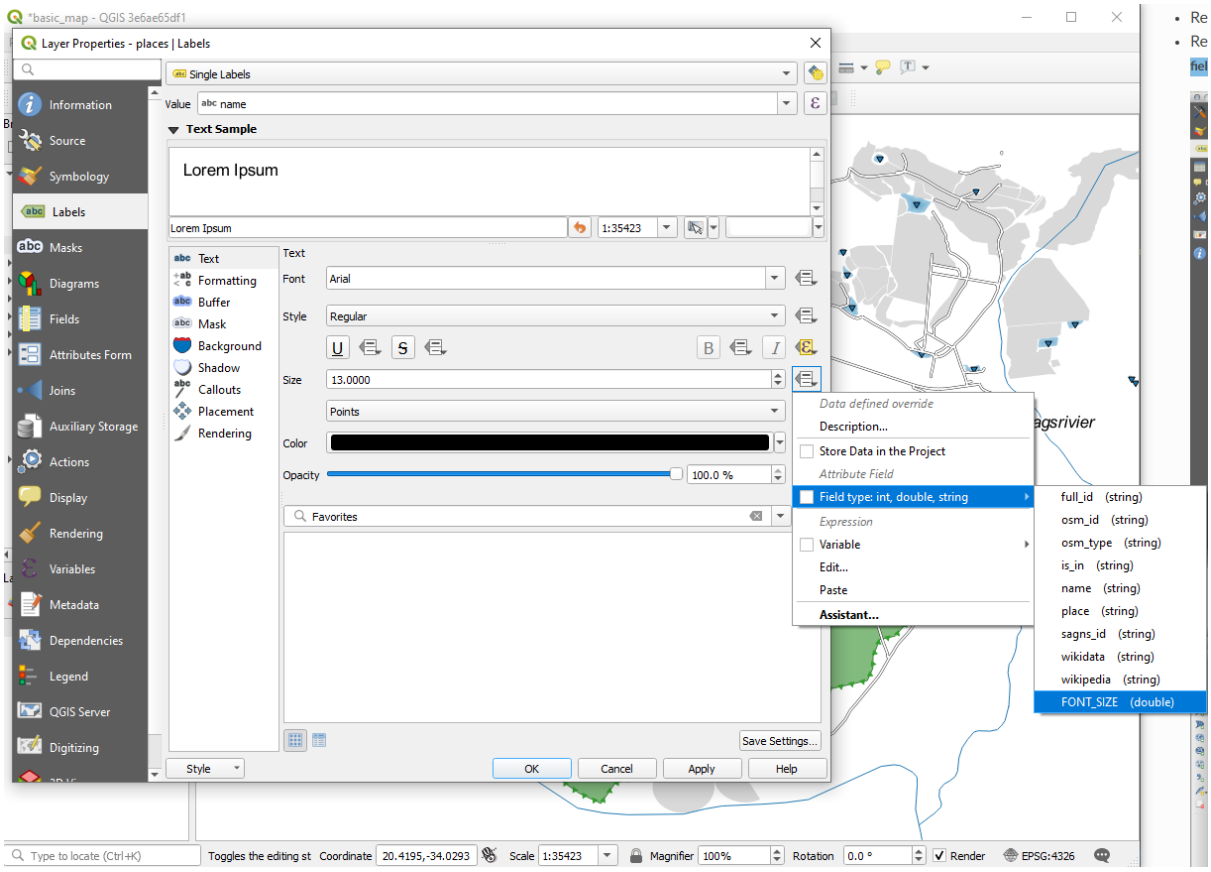
1. Откройте Таблицу атрибутов для `places`.
2. Войдите в режим редактирования, кликнув на кнопку .
3. Добавьте новый столбец с помощью кнопки .
4. Настройте его следующим образом:



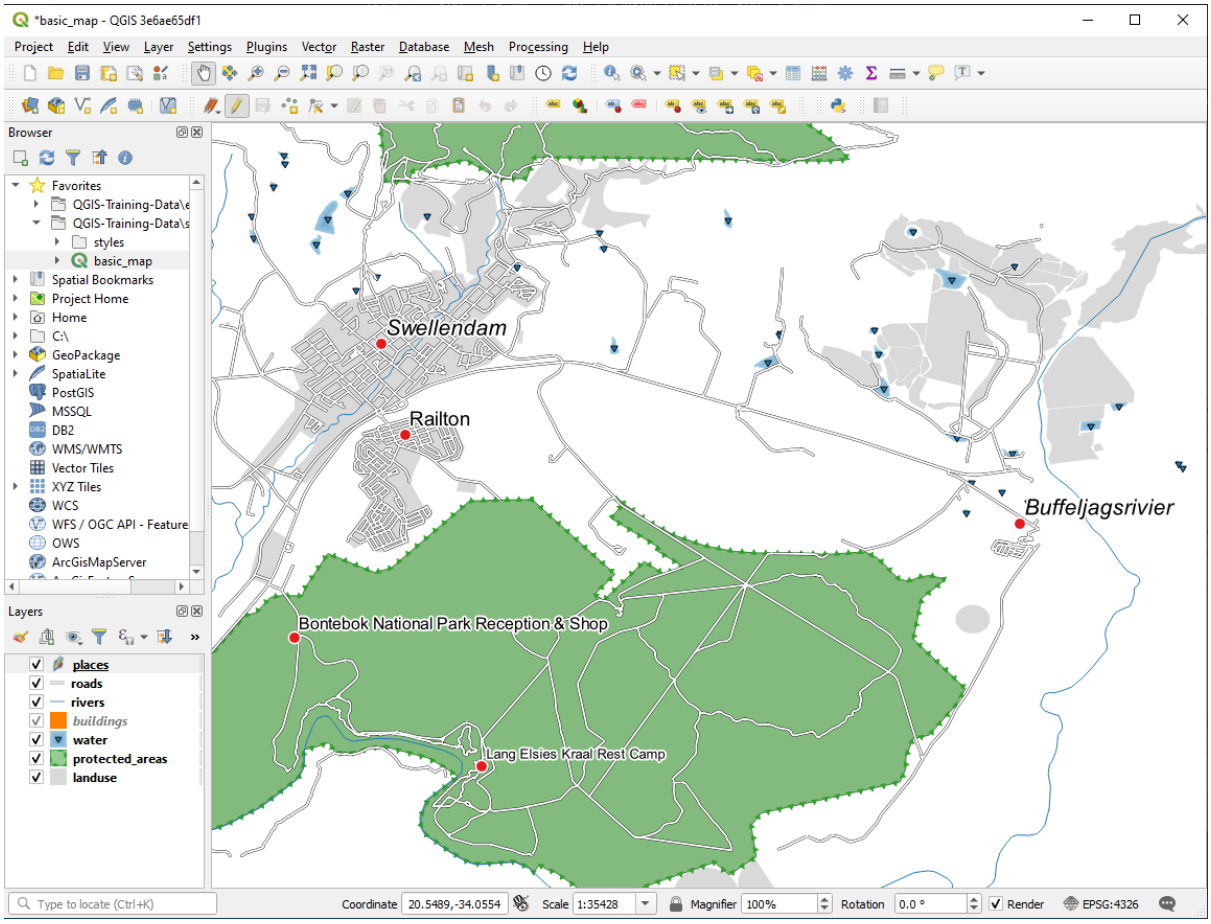
5. Используйте это, чтобы установить собственный размер шрифта для каждого типа места (каждый ключ в поле `place`).

Answer

1. Still in edit mode, set the `FONT_SIZE` values to whatever you prefer. The example uses `16` for towns, `14` for suburbs, `12` for localities, and `10` for hamlets.
2. Remember to save changes and exit edit mode
3. Return to the *Text* formatting options for the `places` layer and select `FONT_SIZE` in the *Attribute field* of the font size data defined override dropdown:



Your results, if using the above values, should be this:



3.2.8 Дополнительные возможности надписей

Мы не можем охватить все варианты в данном курсе, но имейте в виду, что во вкладке *Label* есть много других полезных функций. Вы можете установить визуализацию на основе масштаба, изменить приоритет визуализации для надписей в слое и установить каждый параметр надписи с помощью слоя атрибутов. Вы даже можете установить ротацию, положение XY и другие свойства надписи (если у вас есть поля атрибутов, выделенные для данной цели), а затем отредактировать эти свойства с помощью инструментов, расположенных рядом с основной кнопкой Параметры надписей слоя *Layer Labeling Options*:



(Данные инструменты будут активны, если существуют обязательные поля атрибутов и вы находитесь в режиме редактирования).

Не стесняйтесь исследовать больше возможностей системы надписей.

3.2.9 В заключении

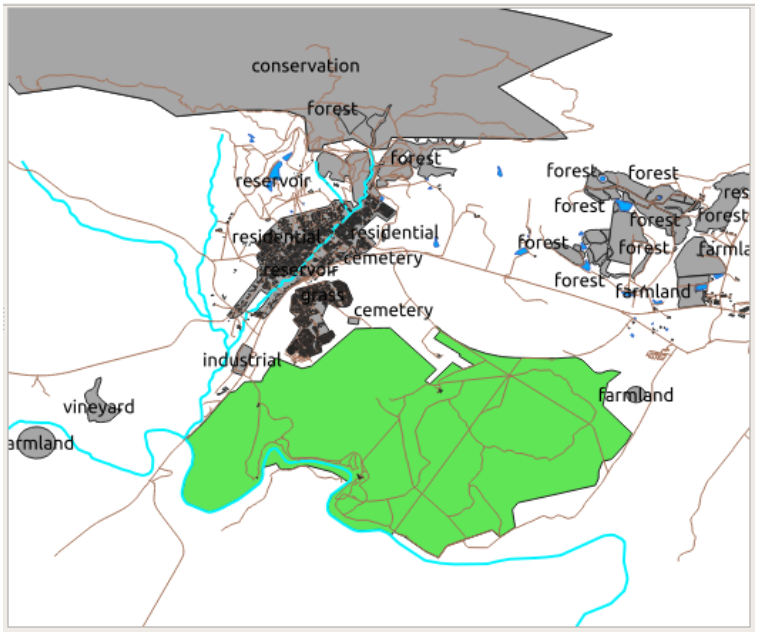
Вы узнали, как использовать атрибуты слоя для создания динамических надписей. Это может сделать вашу карту более информативной и стильной!

3.2.10 Что дальше?

Теперь, когда вы знаете, как атрибуты могут повлиять на вашу карту, как насчет их использования для изменения надписей самих объектов? Это тема следующего занятия!

3.3 Занятие: Классификация

Надписи - это хороший способ передать такую информацию, как названия отдельных мест, но их нельзя использовать для всего. Например, предположим, что кто-то хочет знать, для чего используется каждая местность Землепользования `landuse`. Используя надписи, вы получите следующее:

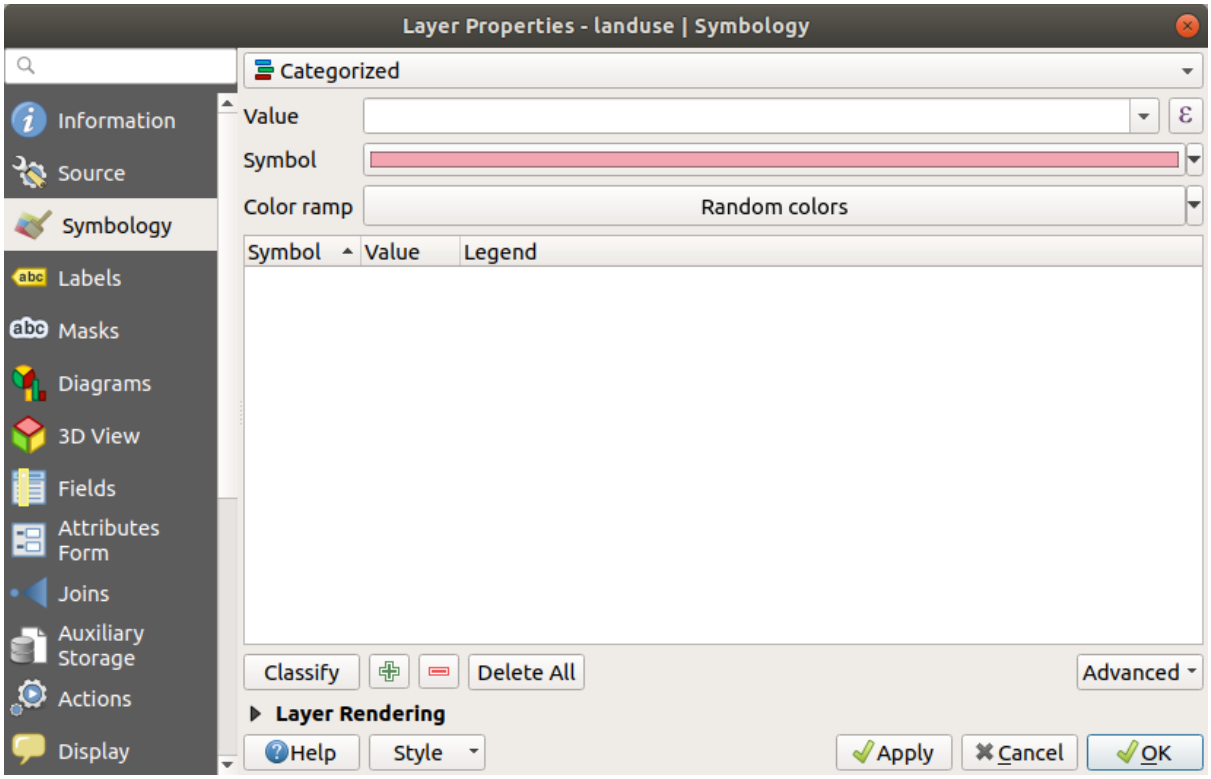


Это делает надписи на карте трудными для чтения и даже непреодолимыми, если на карте много различных участков землепользования.

Цель данного занятия: научиться эффективно классифицировать векторные данные.

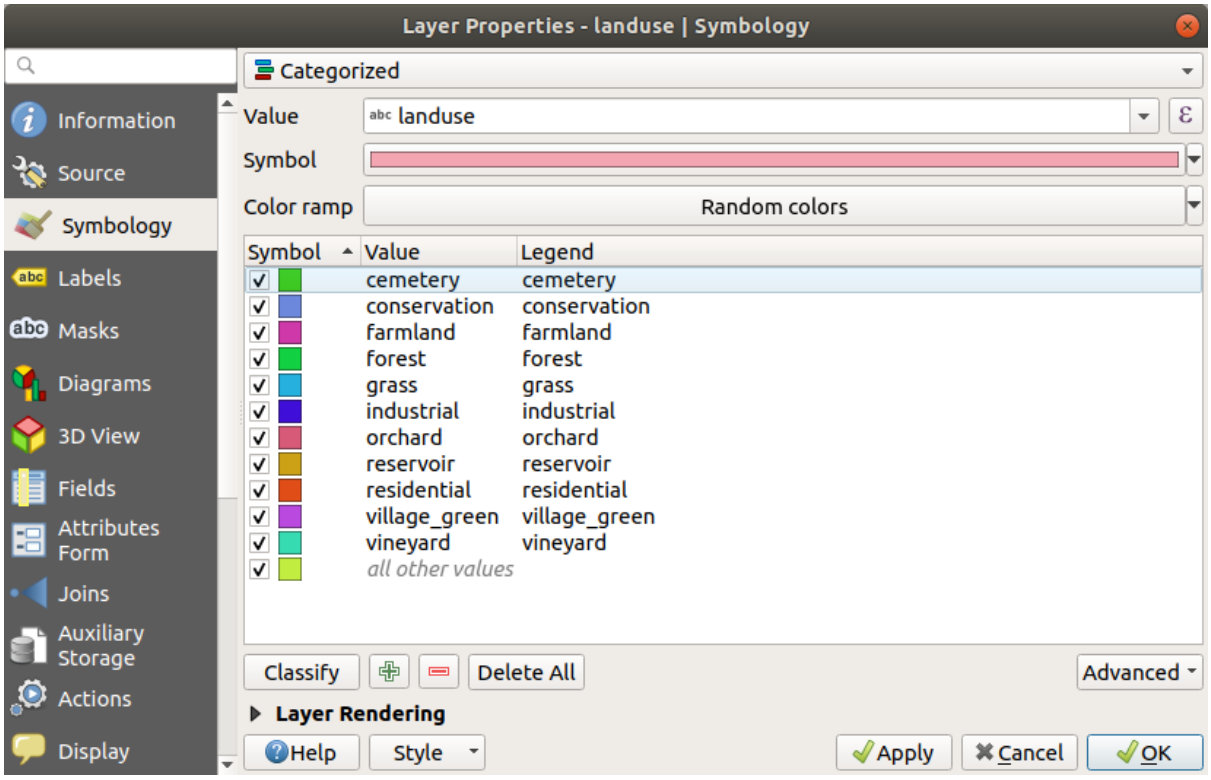
3.3.1 Идем дальше: Классифицируем номинальные данные

1. Откройте диалоговое окно Свойства слоя *Layer Properties* для слоя Землепользование `landuse`.
2. Перейдите во вкладку *Symbology*.
3. Кликните на раскрывающийся список с надписью *Single Symbol* и измените его на По категориям *Categorized*:



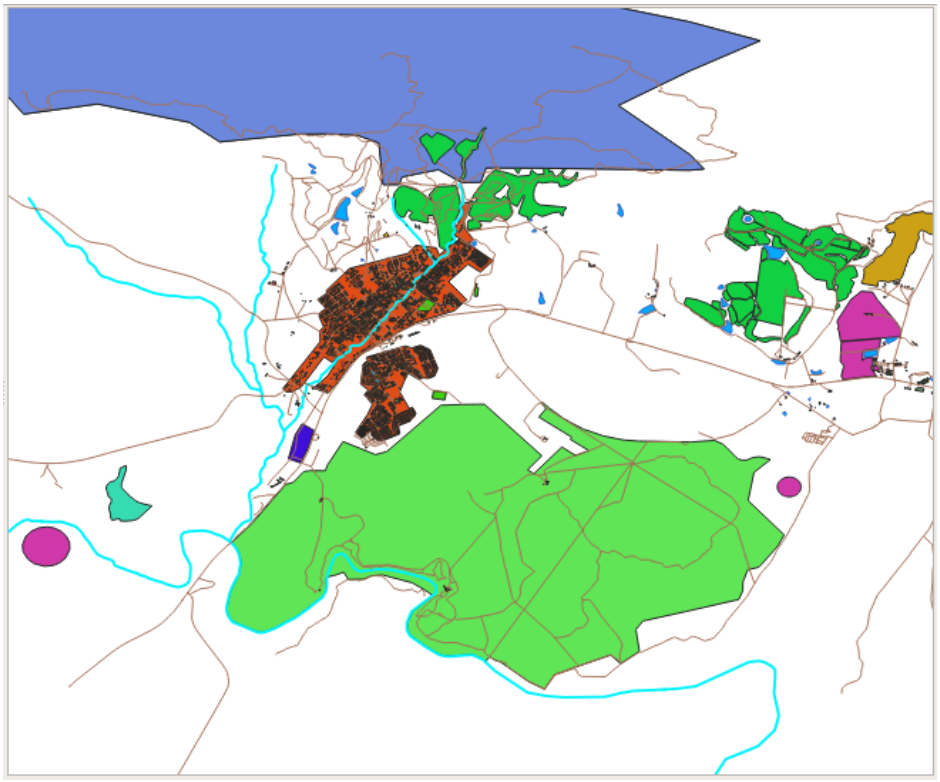
4. В новой панели измените *Value* на `landuse` и измените Цветовая шкала *Color ramp* на Случайный цвет *Random colors*.

5. Кликните на кнопку Классифицировать *Classify*.

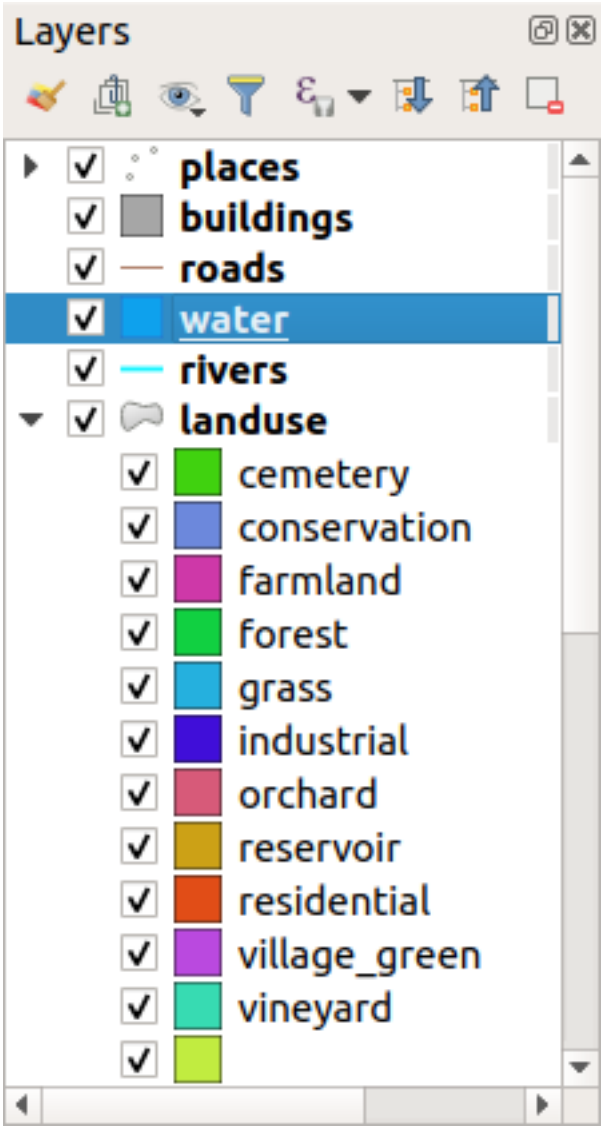


6. Кликните *OK*.

   Вы увидите примерно такое:

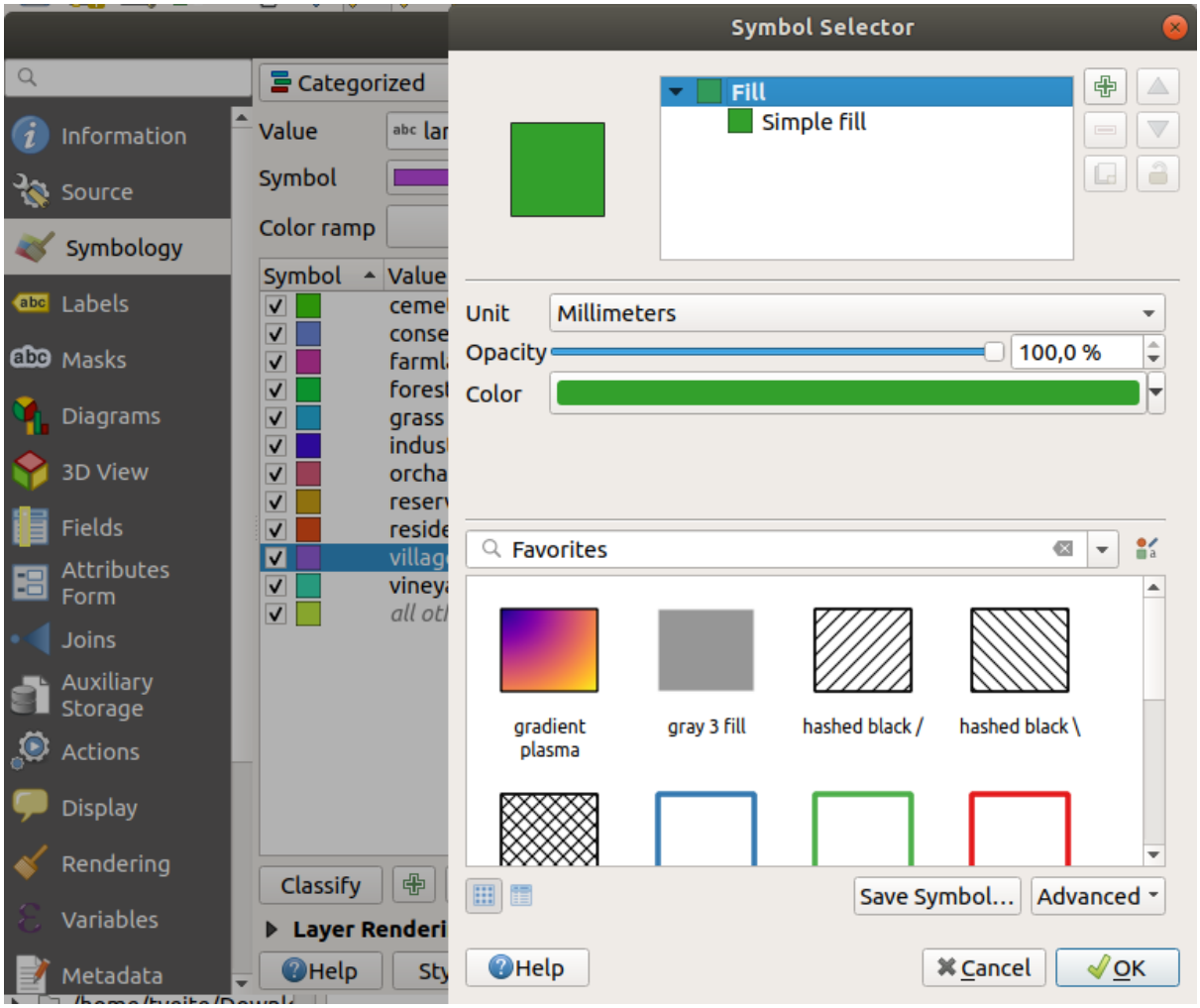


7. Кликните на стрелку (или знак плюса) рядом с `landuse` в панели *Layers*, вы увидите объяснение категорий:

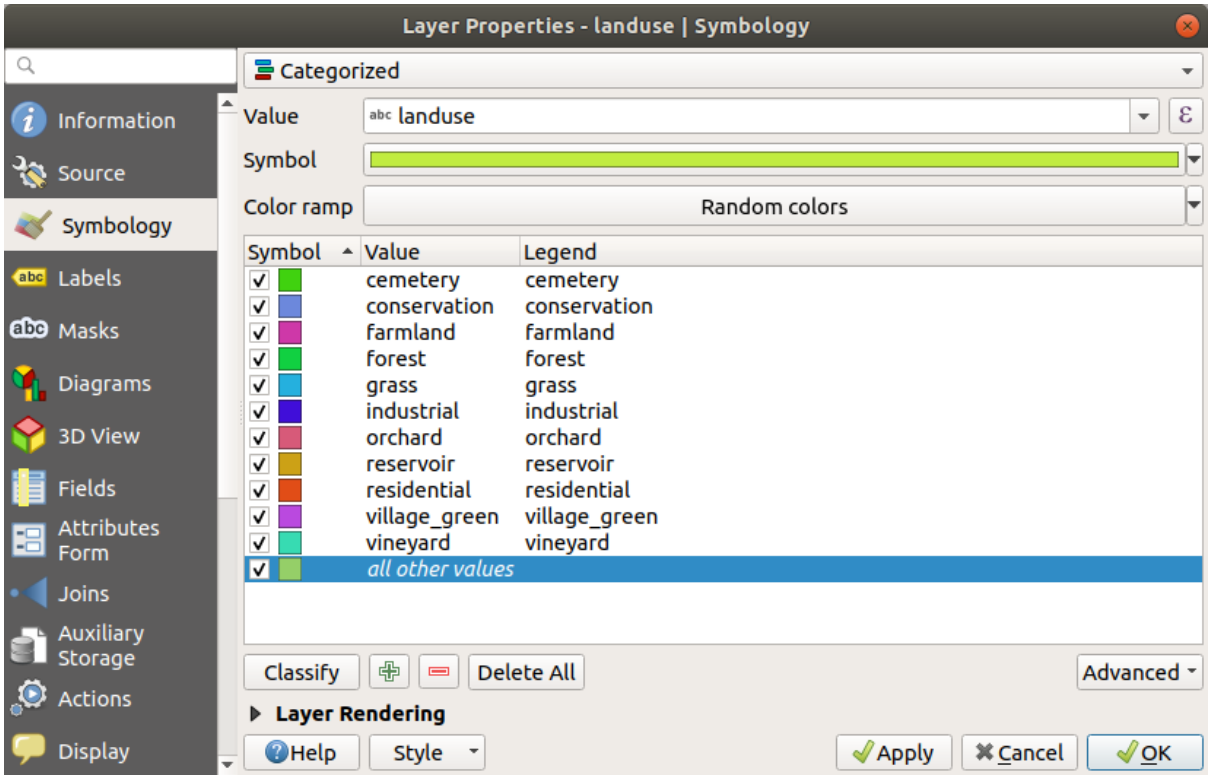


Теперь наши полигоны землепользования окрашены и классифицированы таким образом, что местности с одинаковым землепользованием имеют один цвет.

8. При желании вы можете изменить символ каждой местности землепользования, дважды кликнув на соответствующий цветной блок на панели *Layers* или в диалоговом окне Свойства слоя *Layer Properties*:



Обратите внимание, что имеется одна категория, которая пуста:



Эта пустая категория используется для окраски любых объектов, для которых не определено значение землепользования или которые имеют значение *NULL / НОЛЬ*. Может быть полезно оставить эту пустую категорию, чтобы местности со значением *NULL* все еще отображались на карте. Вы можете изменить цвет, чтобы он более четко представлял пустое значение или значение *NULL*.

Не забудьте сохранить карту сейчас, чтобы не потерять все внесенные тяжелым трудом изменения!

3.3.2 Попробуйте сами: больше Классификации

Примените полученные выше знания, чтобы классифицировать слой Здания `buildings`. Установите категоризацию на поле `building` и используйте цветовую шкалу Спектральная *Spectral*.

Примечание: Не забудьте приблизить местность города, чтобы увидеть результаты.

3.3.3 Идем дальше: Классификация соотношений

Существует четыре типа классификации: *номинальная, порядковая, интервальная и соотношение*.

В **номинальной** классификации категории, по которым классифицируются объекты, основаны на названиях; у них нет определенного порядка. Например, названия городов, коды районов и т.д. Символы, используемые для номинальных данных, не должны подразумевать порядок или величины.

- Для точек мы можем использовать символы различной формы.
- Для полигонов мы можем использовать различные типы штриховки или различные цвета (избегайте смешивания светлых и темных цветов).
- Для линий мы можем использовать различные пунктирные узоры, различные цвета (избегая смешивания светлых и темных цветов) и различные символы вдоль линий.

В **порядковой** классификации категории расположены в определённом порядке. Например, городам мира присваивается ранг в зависимости от их значимости для мировой торговли, путешествий, культуры и т. д. Символы, используемые для порядковых данных, должны подразумевать порядок, но не величину.

- Для точек мы можем использовать символы со светлыми и темными цветами.
- Для полигонов мы можем использовать градуированные цвета (от светлого до темного).
- Для линий мы можем использовать градуированные цвета (от светлого до темного).

В **интервальной** классификации, числа находятся в шкале с положительными, отрицательными и нулевыми значениями. Например, высота выше/ниже уровня моря, температура в градусах по Цельсию. Символы, используемые для данных интервала, должны подразумевать порядок и величину.

- Для точек мы можем использовать символы с различным размером (от малого до большого).
- Для полигонов мы можем использовать градуированные цвета (от светлого до темного) или добавлять диаграммы различного размера.
- Для линий можно использовать толщину (от тонкой до толстой/широкой).

В классификации **соотношения** числа имеют шкалу только с положительными и нулевыми значениями. Например, температура выше абсолютного нуля (0 градусов по Кельвину), расстояние от точки, средний трафик по данной улице в месяц и т. д. Символы, используемые для данных «соотношения», должны подразумевать порядок и величину.

- Для точек мы можем использовать символы с различным размером (от малого до большого).
- Для полигонов мы можем использовать градуированные цвета (от светлого до темного) или добавлять диаграммы различного размера.
- Для линий можно использовать толщину (от тонкой до толстой/широкой).

В приведенном выше примере мы использовали номинальную классификацию для окрашивания каждой записи в слое `landuse` на основе ее атрибута `landuse`. Теперь мы будем применять классификацию соотношения, чтобы классифицировать записи по местностям.

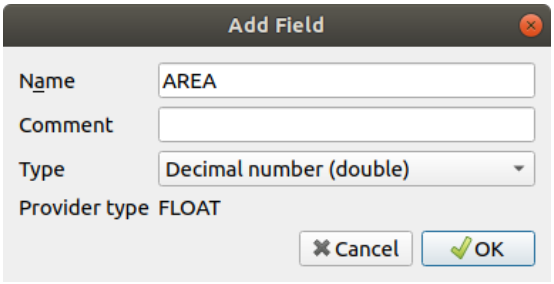
Мы собираемся переклассифицировать слой таким образом, что существующие классы будут утеряны, если их не сохранить. Чтобы сохранить текущую классификацию необходимо:

1. Открыть диалоговое окно Свойств слоя.
2. Кликнуть на кнопку Сохранить стиль… *Save Style …* в раскрывающемся меню Стиль *Style*.
3. Выбрать Переименовать текущий… *Rename Current…*, войти в `land usage` и кликнуть *OK*.

   Категории и их символы теперь сохранены в свойствах слоя.
4. Теперь необходимо кликнуть на кнопку Добавить… *Add…* в раскрывающемся меню *Style* и создать новый стиль с именем Соотношение `ratio`. Таким образом будет сохранена новая классификация.
5. Закрыть диалоговое окно Свойства слоя *Layer Properties*.

Мы хотим классифицировать местности землепользования по размеру, но есть проблема: у них нет поля размера, поэтому придется создать таковое.

1. Откройте таблицу Атрибуты для слоя `landuse`.
2. Войдите в режим редактирования, кликнув на кнопку Переключить редактирование  Toggle editing.
3. Добавьте новый столбец десятичного типа под названием Местность `AREA` с помощью кнопки Новое поле  New field:



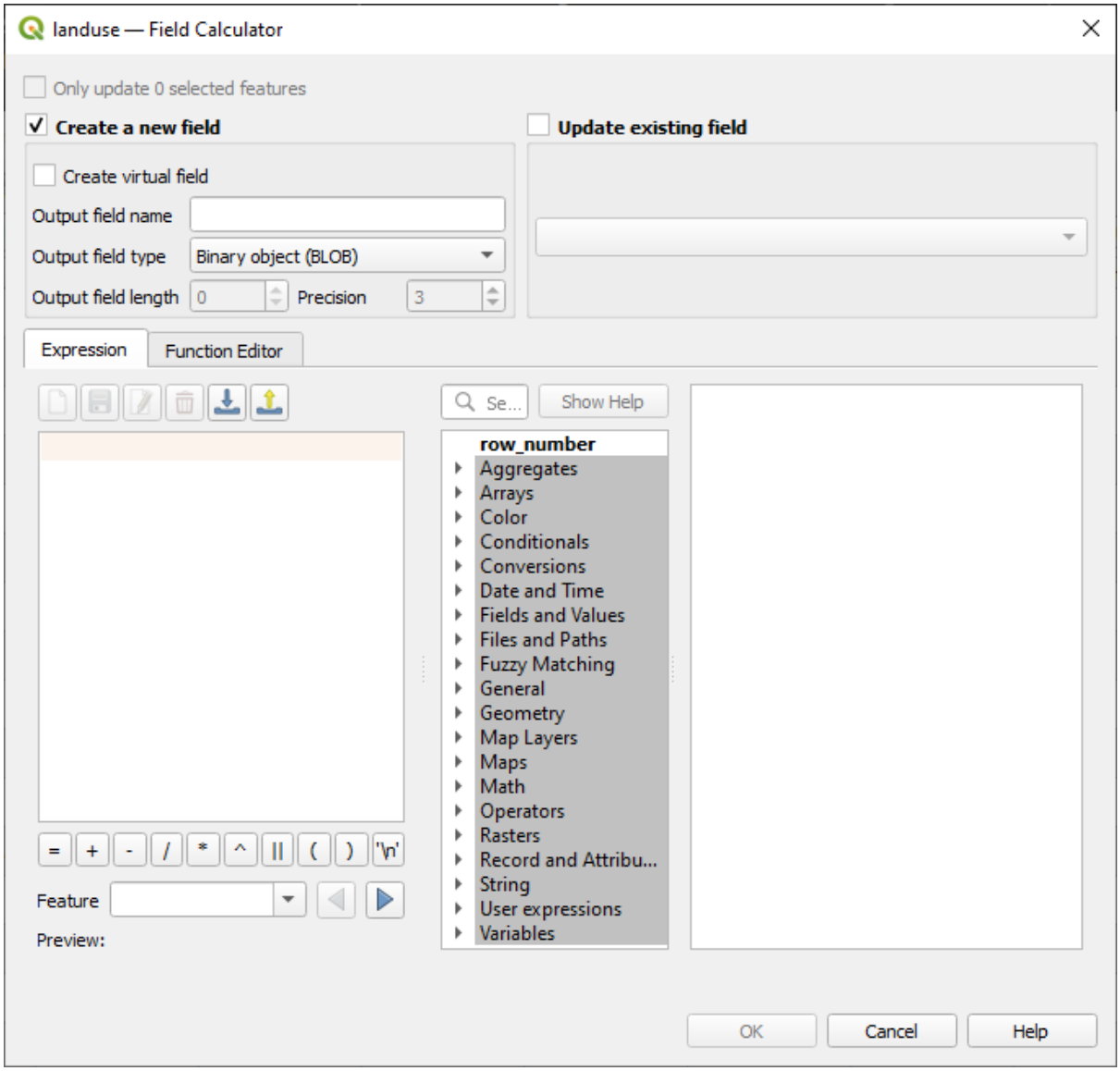
4. Кликните *OK*.

   Будет добавлено новое поле (в крайнем правом углу таблицы, вам может потребоваться прокрутить по горизонтали, чтобы его увидеть). Однако на данный момент оно не заполнено, в нем просто много значений *NULL*.

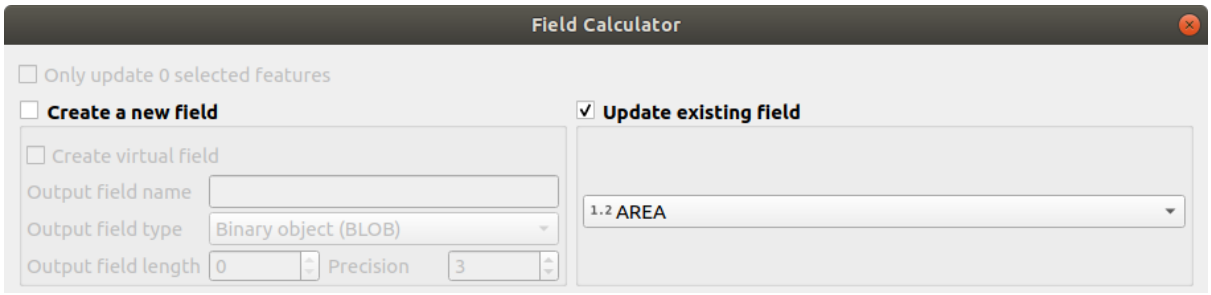
   Чтобы решить эту проблему, нам потребуется рассчитать местности.

   1. Откройте калькулятор поля с кнопкой .

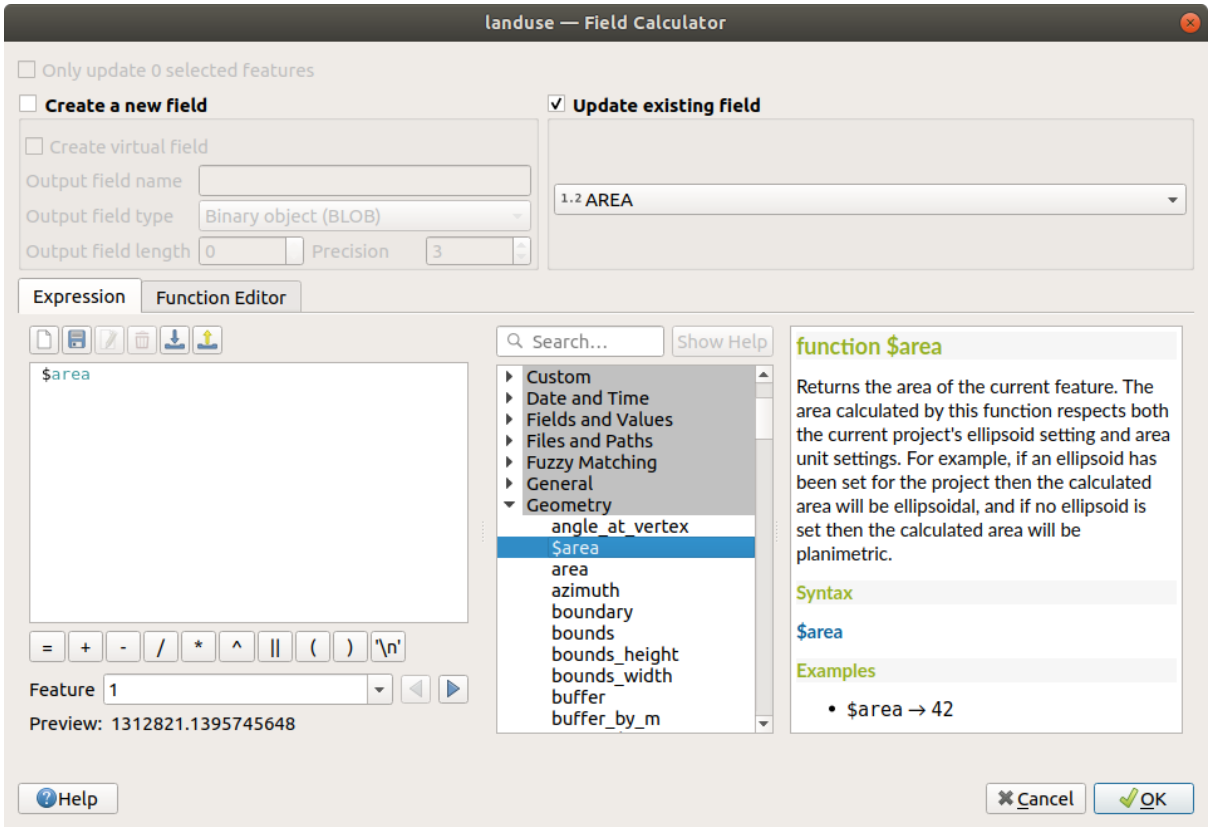
      Вы получите такое диалоговое окно:



2. Установите флажок *Update existing fields* Обновить существующие поля.
3. Выберите *AREA* в полях раскрывающегося меню.



4. Во вкладке Выражение *Expression* разверните список с группой функций *Geometry* и найдите *$area*.
5. Дважды кликните на нее, чтобы она появилась в поле *Expression*.



6. Кликните *OK*.
7. Прокрутите до поля `AREA` в таблице атрибутов и вы заметите, что оно заполнено значениями (возможно, вам потребуется кликнуть на заголовок столбца, чтобы обновить данные).

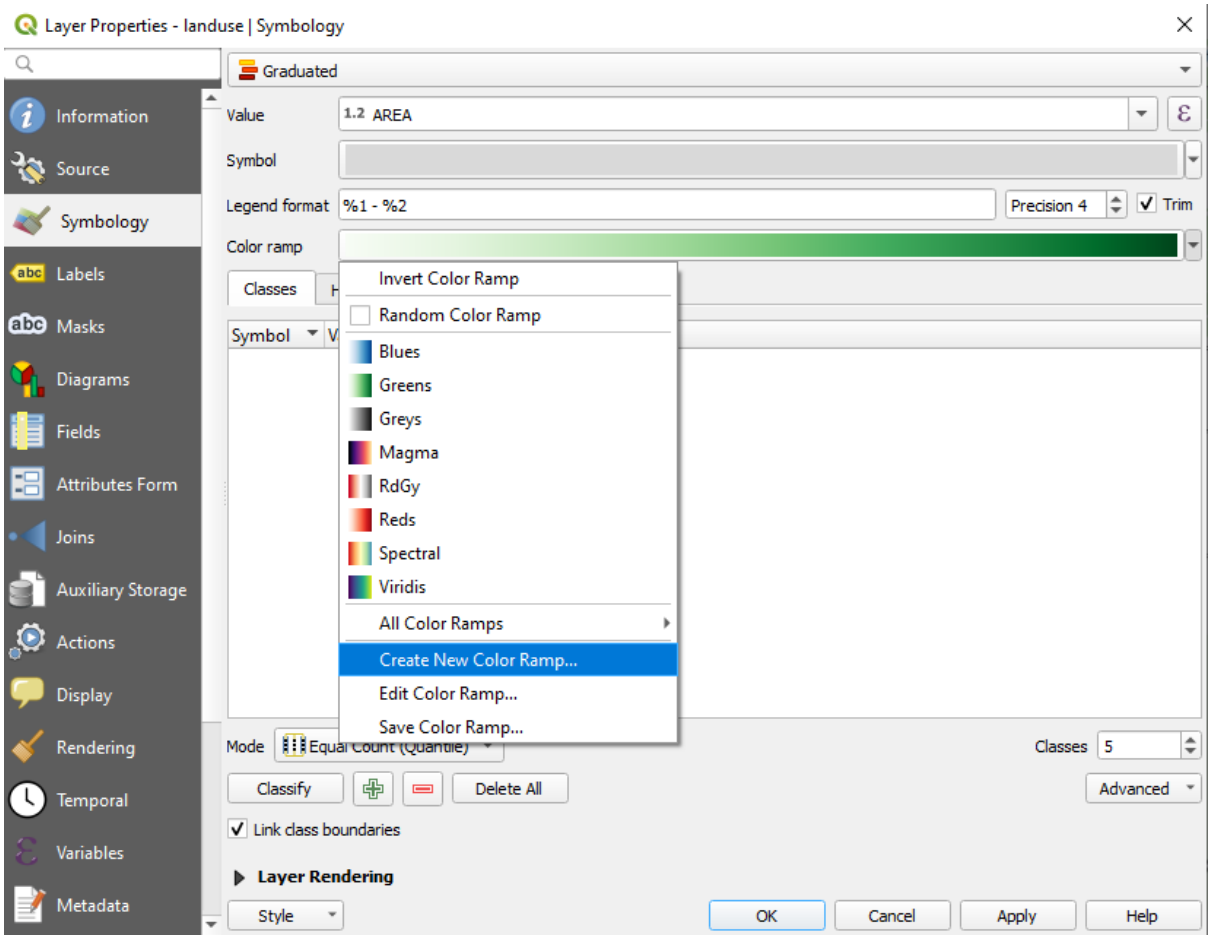
Примечание: Данные местности соответствуют настройкам единицы местности проекта, поэтому они могут быть в квадратных метрах или квадратных градусах.

---

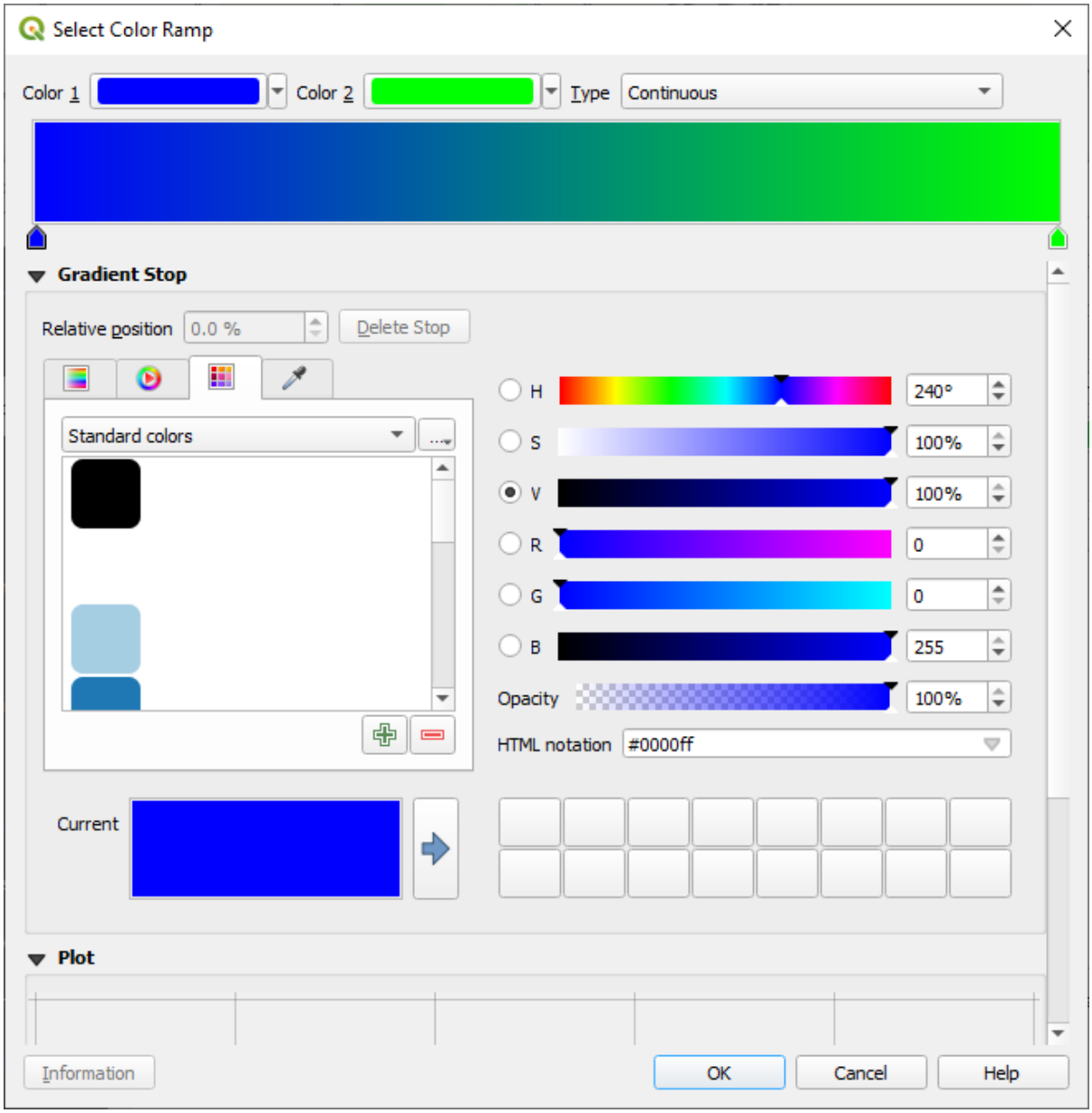
5. Кликните , чтобы сохранить изменения и выйти из режима редактирования с помощью кнопки Переключить редактирование  Toggle editing.
6. Закройте список атрибутов.

Теперь, когда у нас есть данные, давайте воспользуемся ими для визуализации слоя `landuse`.

1. Откройте вкладку *Layer properties* в диалоговом окне *Symbology* для слоя `landuse`.
2. Измените стиль классификации с По категориям *Categorized* на Величина *Graduated*
3. Измените *Value* на `AREA`.
4. В разделе Цветовая шкала *Color ramp`выберите параметр Создать новую цветовую шкалу… :guilabel:`Create New Color Ramp…*:



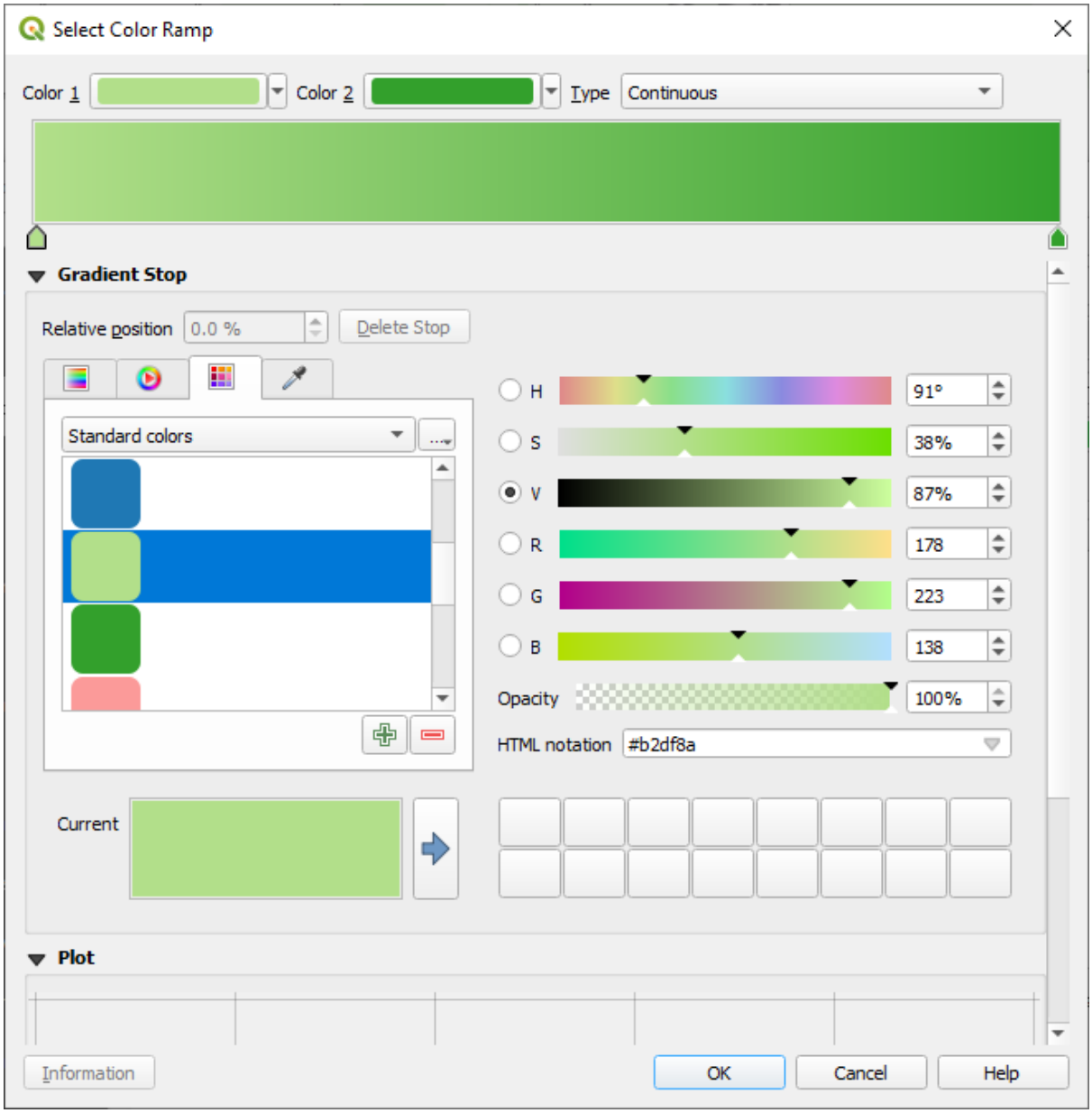
5. Выберите *Gradient* (если он еще не выбран) и кликните *OK*. Вы увидите следующее:



Вы будете использовать это для обозначения местности, небольшие территории как *Color 1*, а большие территории как *Color 2*.

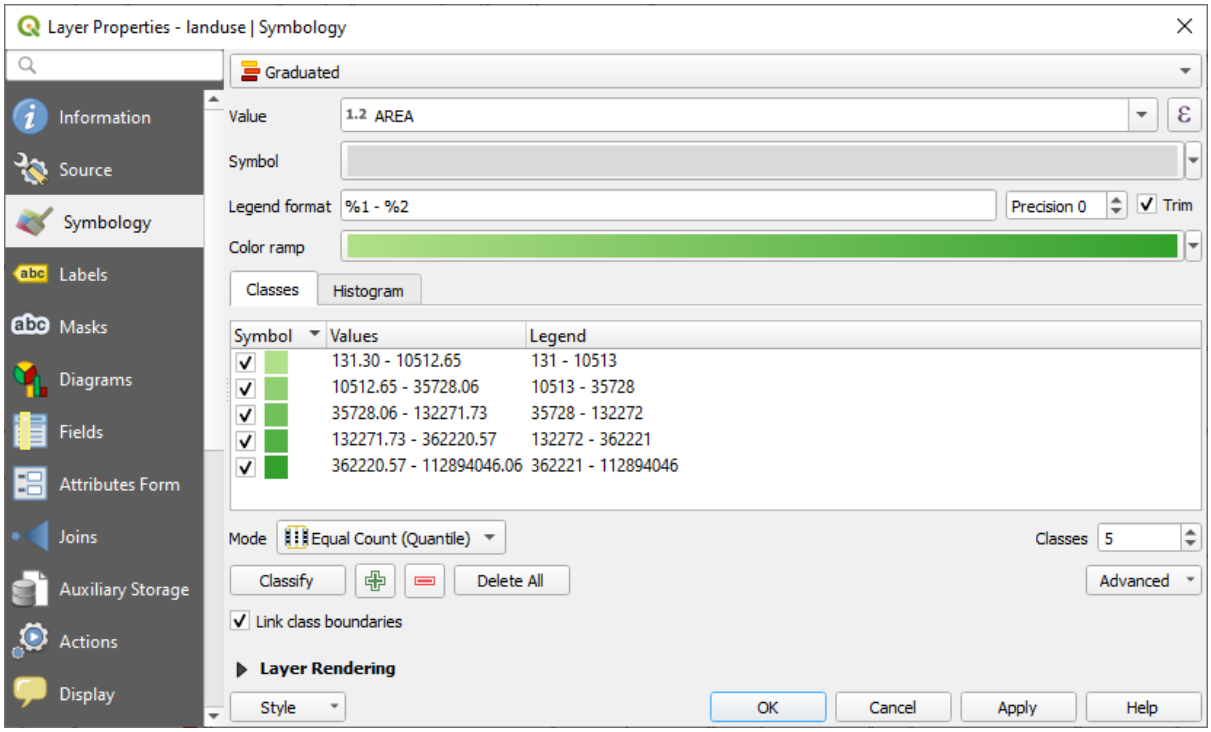
6. Выберите соответствующие цвета.

   В примере ниже, результат выглядит следующим образом:



7. Кликните *OK*.
8. Вы можете сохранить цветовую шкалу выбрав кнопку Сохранить цветовую шкалу *Save Color Ramp...* во вкладке *Color ramp*. Выберите подходящее имя для цветовой шкалы и кликните на *Save*. Теперь вы сможете легко выбрать ту же самую цветовую шкалу выбрав *All Color Ramps*.
9. Кликните на *Classify*

   Теперь у вас будет что-то подобное:



Оставьте все остальное как есть.

10. Кликните *OK*:

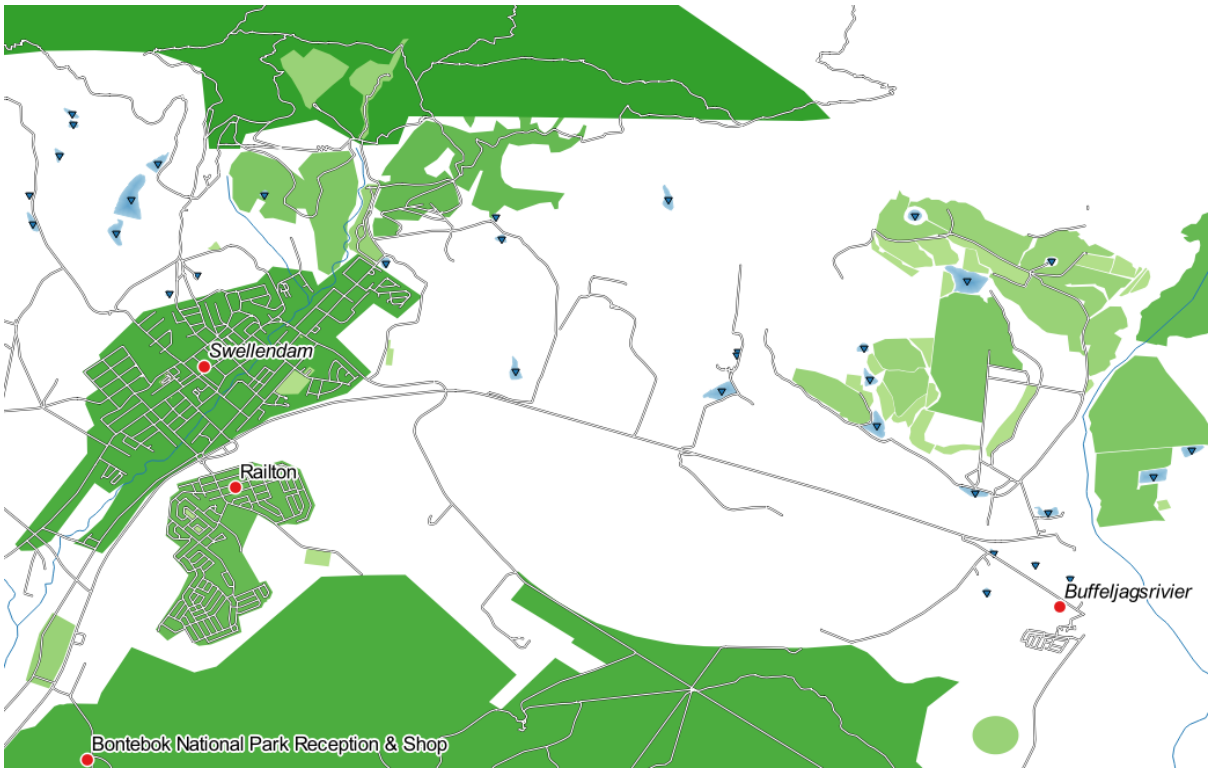


3.3.4  Попробуйте сами: Точно определить Классификацию

- Измените значения Режимы *Mode* и Классы *Classes*, пока не получите логичную классификацию.

Answer

The settings you used might not be the same, but with the values *Classes* = `6` and *Mode* = *Natural Breaks (Jenks)* (and using the same colors, of course), the map will look like this:



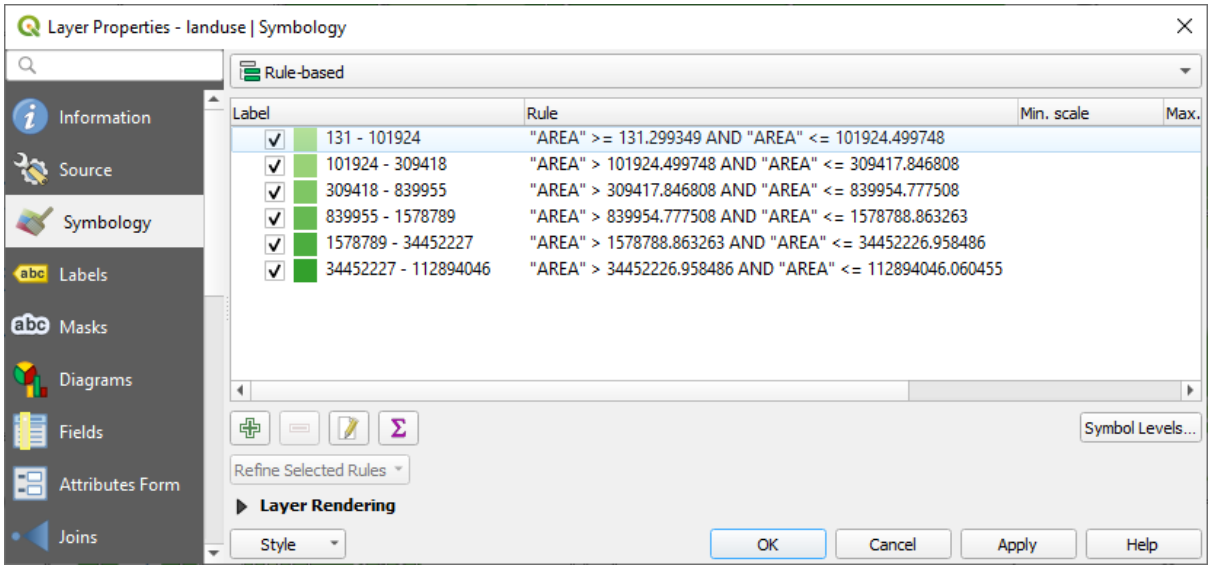
3.3.5  Идем дальше: Классификация на основе правил

Часто бывает полезно комбинировать несколько критериев для классификации, но, к сожалению, обычная классификация учитывает только один атрибут. Вот здесь и пригодится классификация на основе правил.

На этом занятии мы представим слой `landuse` таким образом, чтобы можно было легко отличить город **|Свеллендам|** от других жилых территорий и других типов землепользования (на основе их территории).

1. Откройте диалоговое окно Свойства слоя *Layer Properties* для слоя Землепользование `landuse`.
2. Перейдите во вкладку *Symbology*.
3. Переключите стиль классификации на *Rule-based*.

   QGIS автоматически покажет правила, которые представляют текущую классификацию, выполненные для данного слоя. Например, после выполнения вышеупомянутого упражнения вы должны увидеть что-то вроде этого:

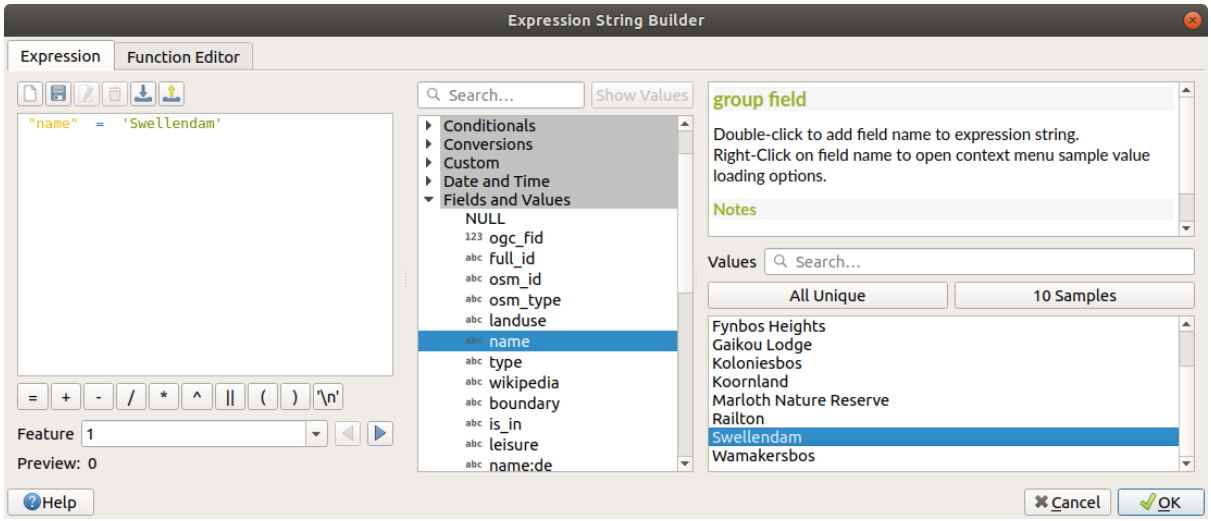


4. Кликните и перетащите мышку, чтобы выбрать все правила.

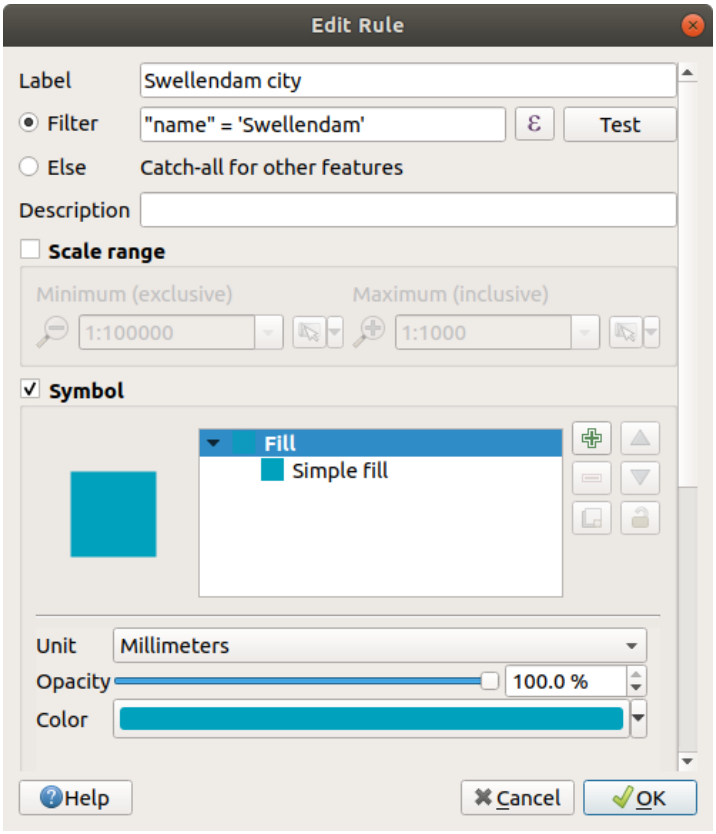
5. Используйте кнопку Удалить выбранные правила Remove selected rules, чтобы удалить все существующие правила.

Давайте теперь добавим наши собственные правила.

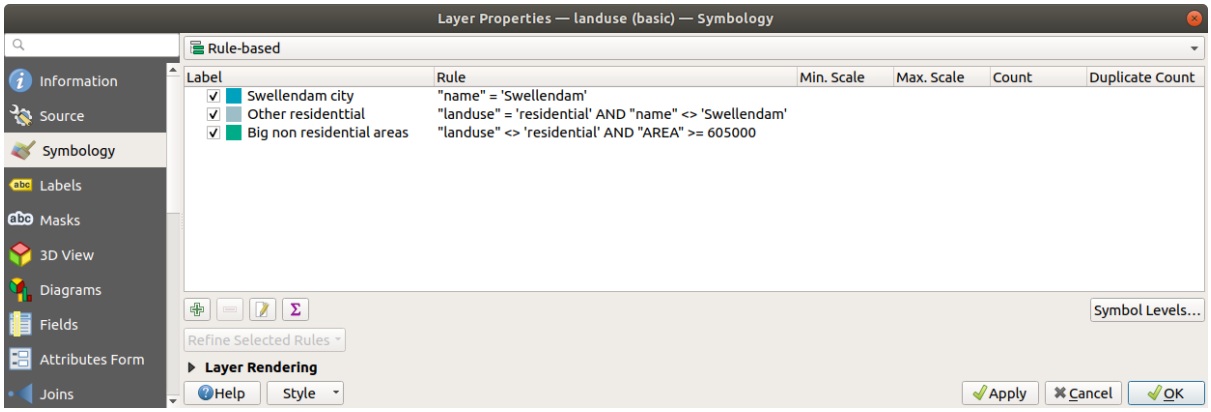
1. Кликните на кнопку Добавить правило Add rule.
2. Появится диалоговое окно Редактировать правило *Edit rule*.
3. Введите Город Свеллендам `Swellendam city` как *Label*.
4. Кликните кнопку ε рядом с текстовой областью Фильтр *Filter*, чтобы открыть Конструктор строки выражений *Expression String Builder*.
5. Введите критерий `"name" = 'Swellendam'` и подтвердите.



6. Back to the *Edit rule* dialog, assign it a darker grey-blue color in order to indicate the town's importance in the region and remove the border

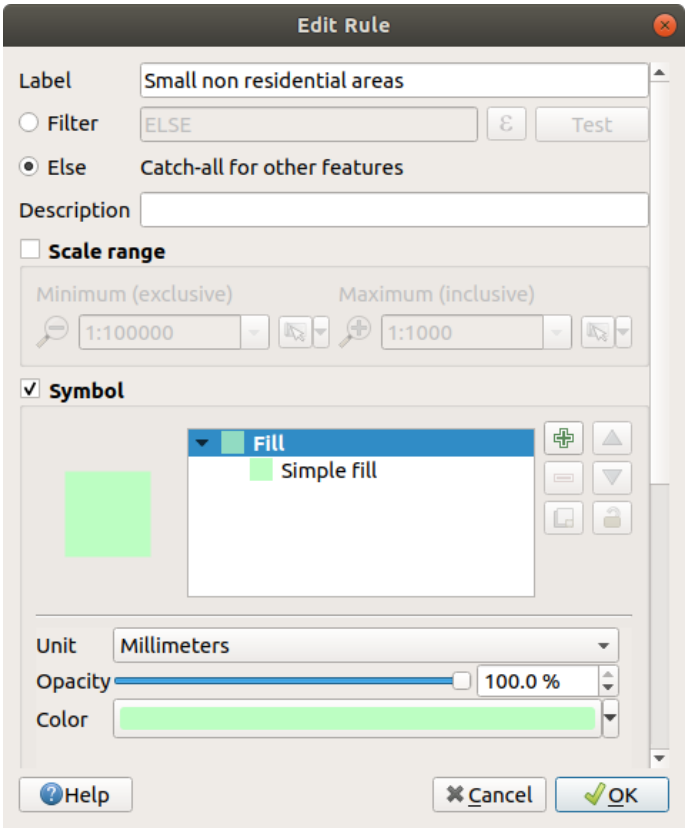


7. Кликните *OK*.
8. Повторите вышеуказанные шаги, чтобы добавить следующие правила:
    1. Надпись **Другие жилые** с критерием `"landuse" = 'residential' AND "name" <> 'Swellendam'` (или `"landuse" = 'residential' AND "name" != 'Swellendam'`). Выберите бледный сине-серый цвет заливки в *Fill color*.
    2. Надпись **Большие нежилые местности** с критерием `"landuse" <> 'residential' AND "AREA" >= 605000`. Выберите средне-зеленый цвет.

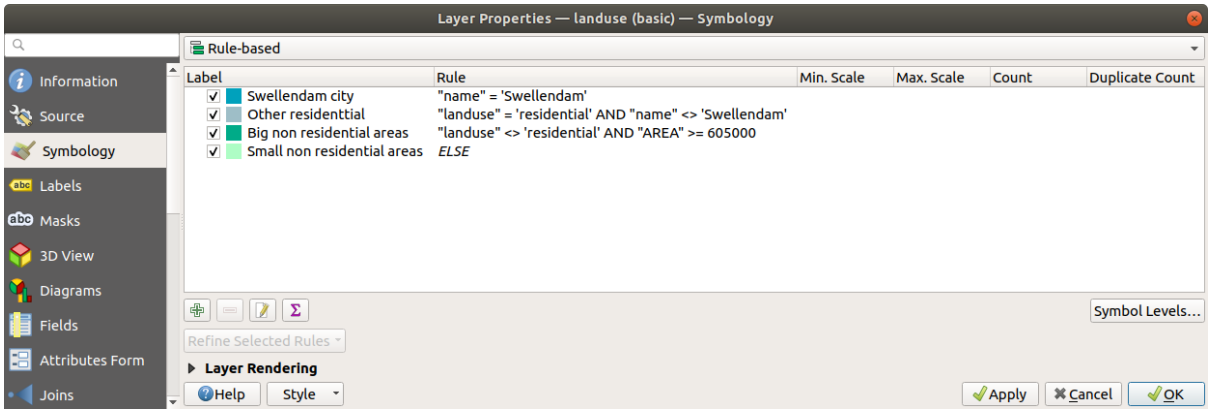


Данные фильтры являются эксклюзивными, поскольку они исключают местности на карте (нежилые районы размером менее 605000 (квадратных метров) не включены ни в одно из правил).

3. Мы «догоним» оставшиеся функции, используя новое правило с надписью **Небольшие нежилые местности**. Вместо выражения фильтра проверьте *Else*. Придайте этой категории подходящий бледно-зеленый цвет.



Ваши правила теперь выглядят следующим образом:



9. Примените данную символику.

Ваша карта будет выглядеть примерно так:



Теперь у вас есть карта, на которой Свеллендам Swellendam – самая известная жилая местность, а другие нежилые местности, окрашены в соответствии с их размером.

3.3.6 В заключении

Символика позволяет нам представить атрибуты слоя в удобной для чтения форме. Это позволяет нам, а также читателю карты понять значение функций, используя любой выбранный нами соответствующий атрибут. В зависимости от проблем, с которыми вы сталкиваетесь, вы будете применять различные методы классификации для их решения.

3.3.7 Что дальше?

Теперь у нас есть красивая карта, но как мы собираемся вывести ее из QGIS в формат, который мы можем распечатать или преобразовать в изображение или PDF? Это тема следующего занятия!

ГЛАВА 4

Модуль: Составление базовой карты

В данном модуле, вы узнаете, как применять макет для печати QGIS для создания качественных карт со всеми необходимыми компонентами.

4.1 Занятие: Создание динамического макета печати

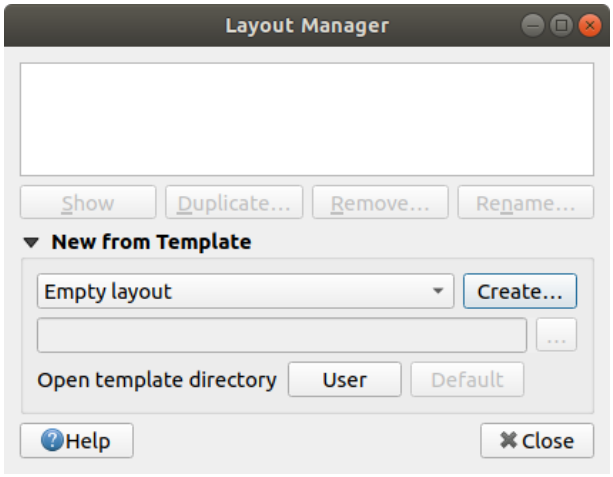
Теперь, когда у вас есть карта, у вас должна быть возможность распечатать ее или выгрузить в документ. Причина заключается в том, что файл карты GIS не является изображением. Скорее, он сохраняет состояние программы GIS, со ссылками на все слои, их надписи, цвета и т.д. Таким образом, для тех, у кого нет данных или одной и той же программы GIS (например, QGIS), файл карты будет бесполезен. К счастью, QGIS может выгрузить файл карты в формат, который может прочитать любой компьютер, а также распечатать карту, если у вас есть подключенный принтер. Как выгрузка, так и печать выполняются с помощью *Макета для печати*.

****Цель данного занятия: ** применить *Макет для печати* QGIS для создания базовой карты со всеми необходимыми настройками.

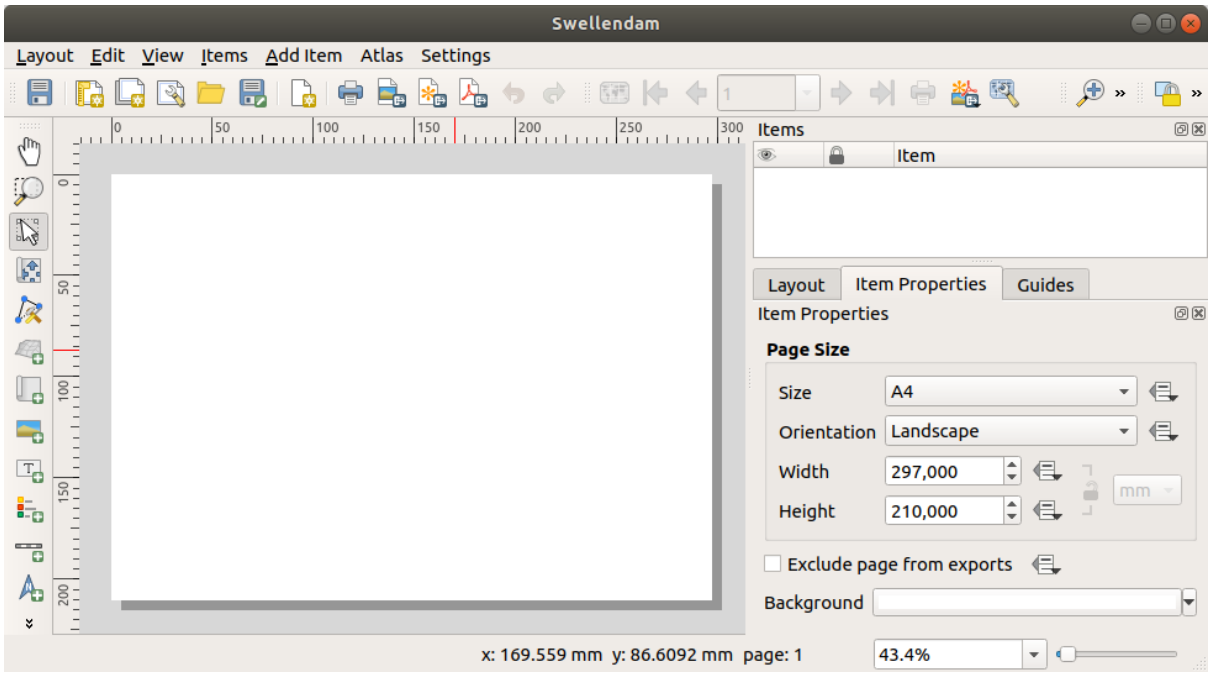
4.1.1  Идем дальше: Диспетчер макетов

QGIS позволяет вам создать множество карт, используя один и тот же файл карты. Для этого имеется инструмент под названием *Диспетчер макетов*.

1. Кликните на *Project* ► *Layout Manager…* , чтобы открыть этот инструмент. Появится пустое диалоговое окно Диспетчер макетов *Layout manager*.

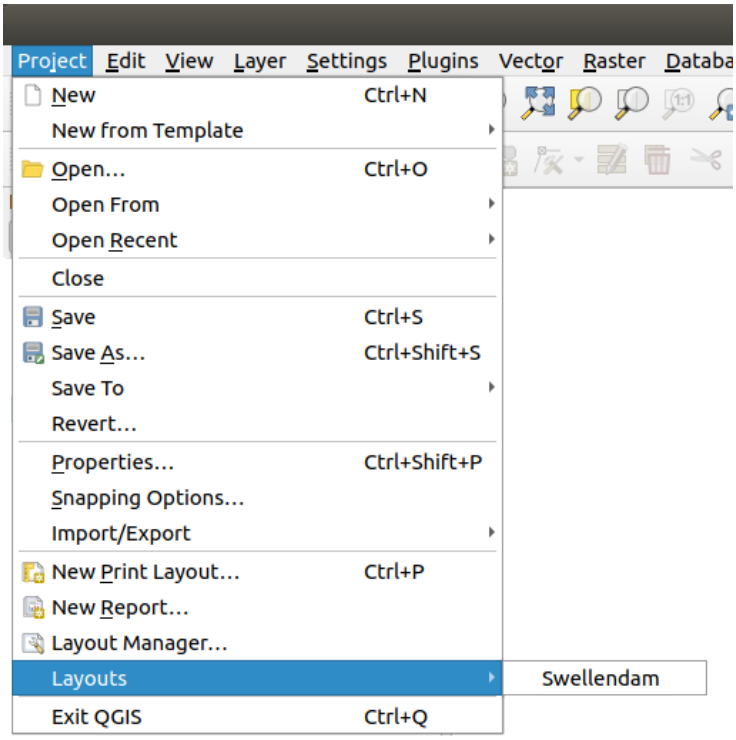


2. В разделе Новый из шаблона *New from Template* выберите Пустой макет *Empty layout* и кликните на кнопку *Create…*.
3. Присвойте новому макету имя Свеллендам Swellendam и кликните *OK*.
4. Появится окно *Макет для печати*:



Вы также можете создать новый формат с помощью команды меню *Project* ► *New Print Layout…*.

Какой бы маршрут вы не выбрали, новый макет для печати теперь доступен из меню *Project* ► *Layouts* ►, как показано на рисунке ниже.



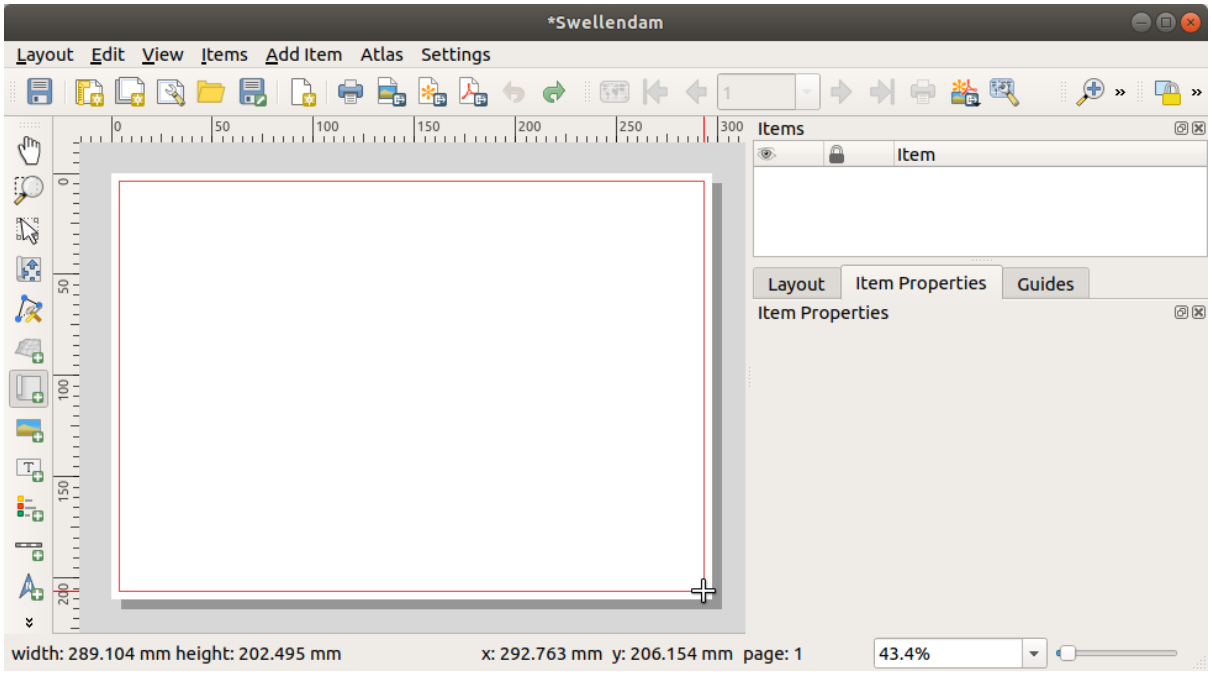
4.1.2 Идем дальше: Составляем базовую карту

В данном примере композиция карты уже была такой, какой мы ее хотели видеть. Убедитесь, что ваша тоже.

1. Кликните правой кнопкой мыши на лист в центральной части макета окна и выберите Свойства страницы… *Page properties…* в контекстном меню.
2. Убедитесь, что во вкладке Свойства элемента *Item Properties* установлены следующие значения:
   - *Size*: `A4`
   - *Orientation*: `Landscape` Ориентация: Ландшафт

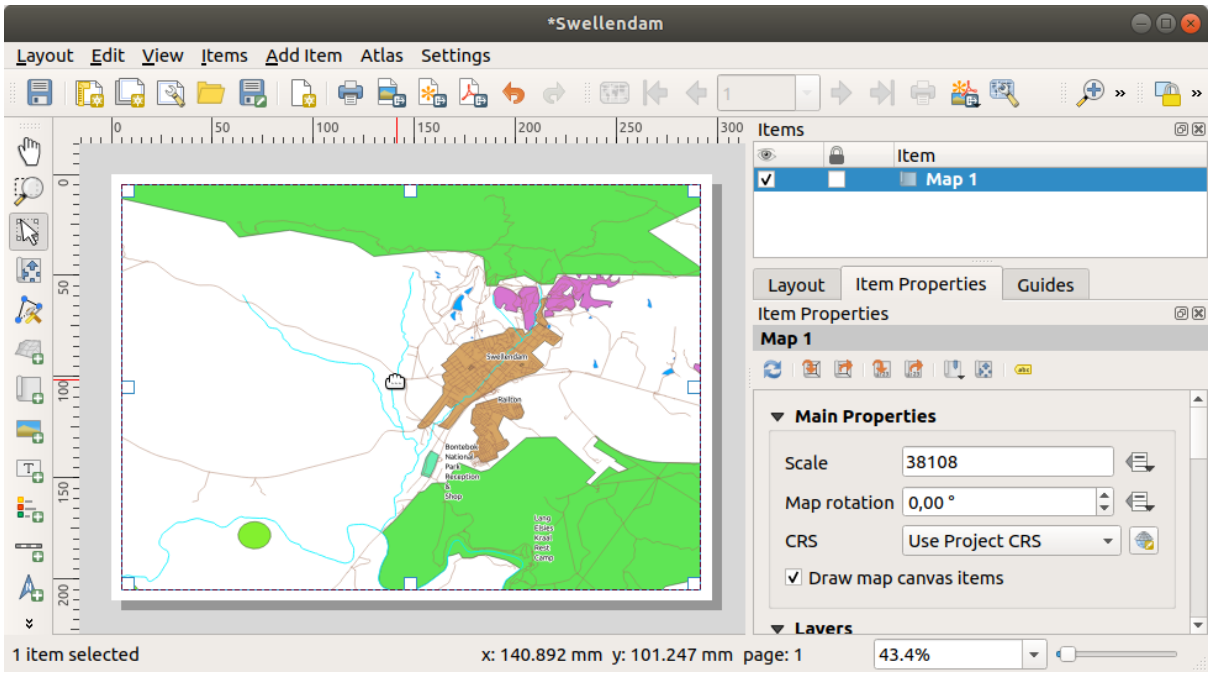
   Теперь у вас есть макет страницы такой, как вы хотели, но эта страница все еще пуста. Карты явно не хватает. Давайте исправим это!
3. Кликните на кнопку Добавить карту  Add Map.

   Активировав данный инструмент, вы сможете разместить карту на странице.
4. Кликните и перетащите рамку на пустую страницу:

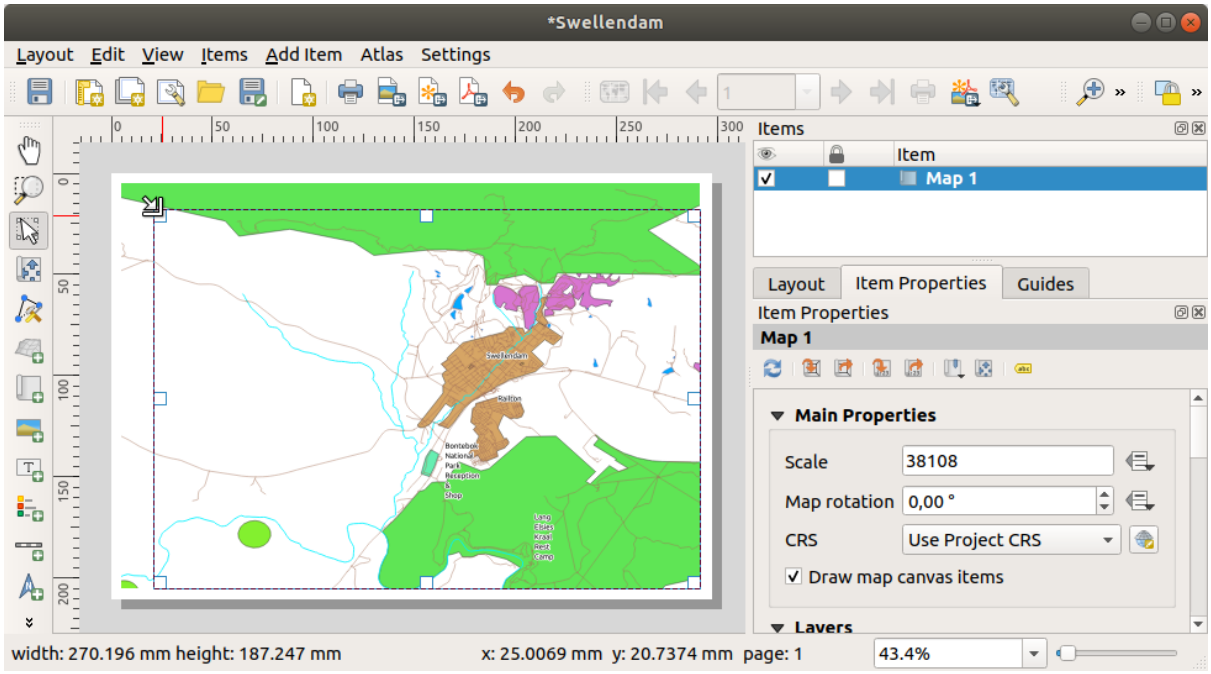


На странице появится карта.

5. Подвигайте карту, кликая и перетаскивая ее:



6. Измените ее размер, кликая и перетаскивая рамку по краям:



Примечание: Конечно, ваша карта может выглядеть совсем иначе! Это зависит от того, как настроен ваш собственный проект. Но не волнуйтесь! Данные инструкции являются основными, поэтому они будут работать одинаково независимо от того, как выглядит сама карта.

7. Обязательно оставьте границы по краям и пространство вверху для заголовка.
8. Приближайте и отдаляйте страницу (но не карту!) с помощью следующих кнопок:

9. Масштабируйте и прокручивайте карту в главном окне QGIS. Карту также можно прокрутить с помощью инструмента Переместить содержимое элемента  Move item content.

   Вид карты обновляется при приближении и отдалении.

10. Если по какой-то причине вид карты не обновляется правильно, вы можете принудительно обновить карту, кликнув на кнопку Обновить вид  Refresh view.

    Помните, что размер и положение, которые вы задали на карте, не обязательно должны быть окончательными. Если вы не удовлетворены, то вы всегда можете вернуться и изменить их позже. На данный момент вам нужно убедиться, что вы сохранили свою работу на этой карте. Поскольку Макет для печати в QGIS является частью основного файла карты, вы должны сохранить свой проект.

11. Перейдите в *Layout* ►  *Save Project*. Это удобный ярлык для перехода на тот, который находится в главном диалоговом окне.

4.1.3 Идем дальше: Добавляем заголовок

Теперь ваша карта выглядит на странице хорошо, но читающим ее пользователям она еще не говорит, что на ней есть. Им нужен какой-то контекст, который вы им предоставите, добавив элементы карты. Сначала добавим заголовок:

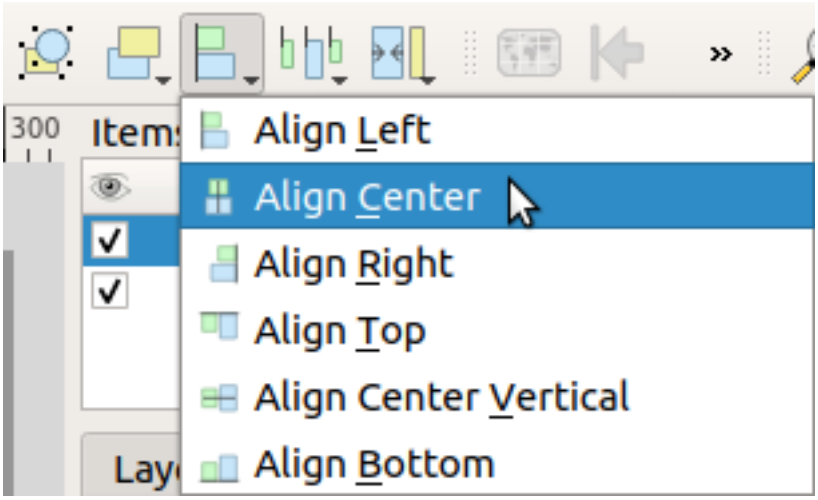
1. Кликните на кнопку Добавить надпись Add Label.
2. Кликните на страницу над картой, примите предложенные значения в диалоговом окне Свойства нового элемента *New Item Properties*, и в верхней части карты появится надпись.
3. Измените размер и поместите его в центр вверху страницы. Можно менять размер и перемещать его так же, как вы изменяли размер и перемещали карту.

   Когда вы переместите заголовок, вы заметите появление рекомендаций, которые помогут вам разместить заголовок в центре страницы.

   Однако на панели инструментов Действия есть инструмент, который помогает расположить заголовок относительно карты (а не страницы):

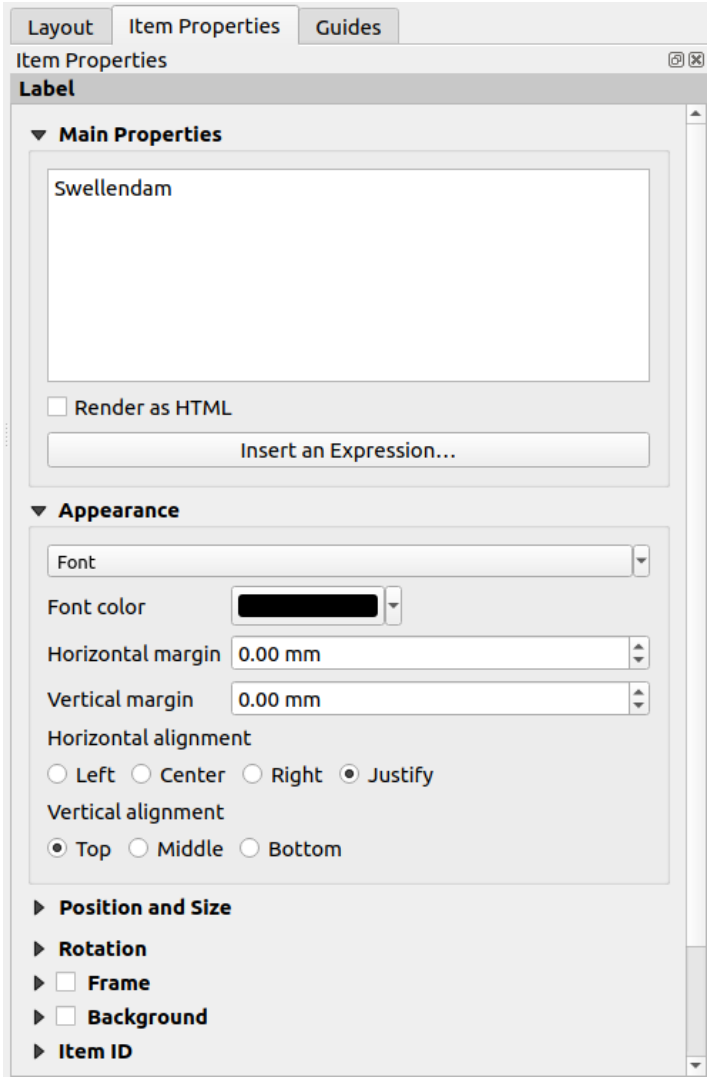
   

4. Кликните на карту, чтобы выбрать его.
5. Удерживая `Shift` на клавиатуре, кликните на надпись, чтобы выделить и карту, и надпись.
6. Найдите кнопку Выровнять выбранные элементы влево Align selected items left и кликните на стрелку раскрывающегося списка рядом с ней, чтобы отобразить параметры позиционирования, и кликните Выровнять по центру *Align center*:



   Теперь рамка надписи расположена по центру на карте, но не содержимое. Чтобы разместить содержимое надписи по центру необходимо:

   1. Выбрать надпись, кликнув на нее.
   2. Кликнуть на вкладку Свойства предмета *Item Properties* на боковой панели окна макета.
   3. Изменить текст надписи на «Свеллендам» «Swellendam»:



4. Используйте данный интерфейс для настройки шрифта и параметров выравнивания в разделе *Appearance*:

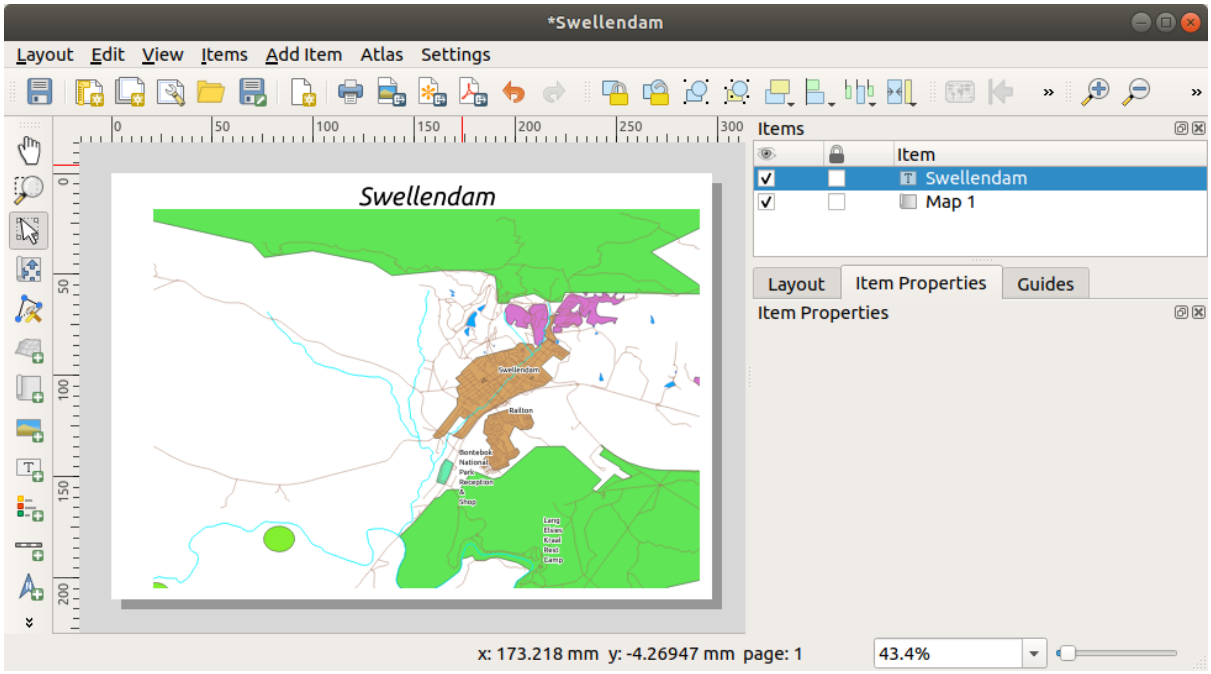
    1. Выберите крупный, в разумных пределах, шрифт (в данном примере будет использоваться шрифт по умолчанию размером `36`).
    2. Установите Горизонтальное выравнивание *Horizontal Alignment* по Центру *Center*.

    Вы также можете изменить цвет шрифта, но, скорее всего, лучше оставить его черным по умолчанию.

5. Настройки по умолчанию не могут добавить рамку в текстовое поле заголовка. Однако, если вы хотите добавить рамку, вы можете сделать это следующим образом:

    1. На вкладке *Item Properties* прокрутите вниз, пока не увидите параметр Рамка *Frame*.
    2. Нажмите на *Frame*, чтобы включить рамку. Вы также можете изменить цвет и ширину рамки.

В этом примере мы не включаем рамку, поэтому наша страница пока выглядит таким образом:



Вы можете зафиксировать элементы на месте, чтобы случайно не переместить их после выравнивания:

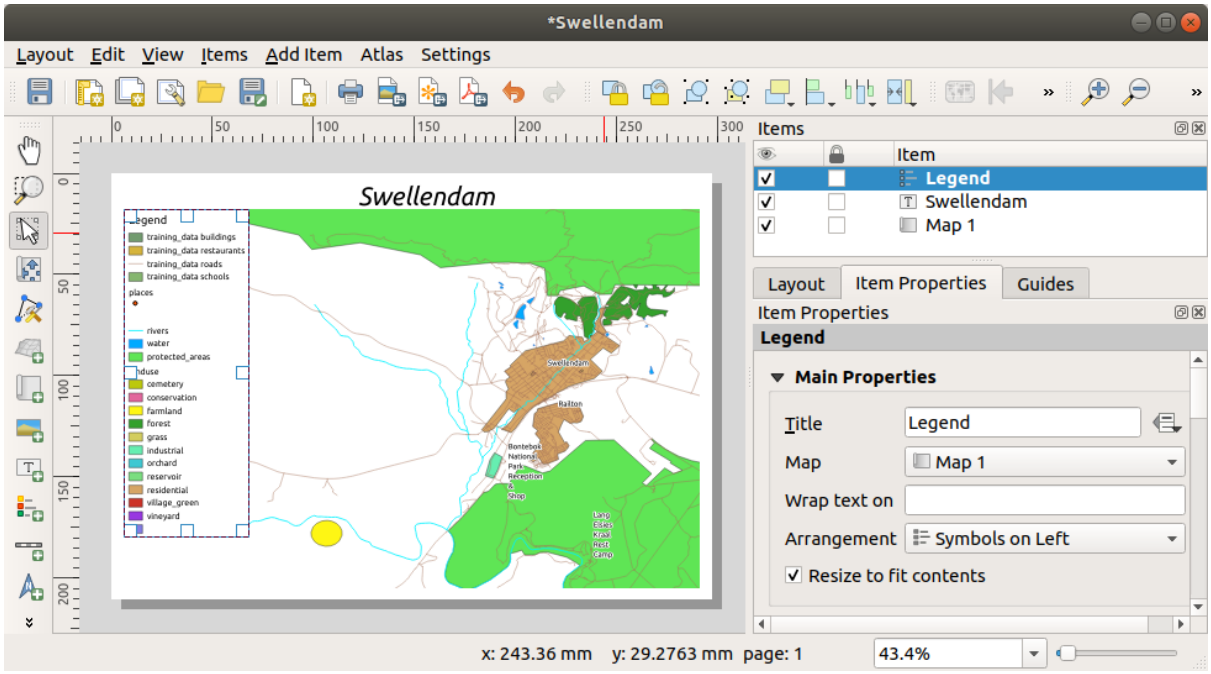
1. Выберите надпись и элементы карты.
2. Кликните на кнопку Заблокировать выбранные элементы Lock Selected Items на панели инструментов *Действия*.

Примечание: Кликните на Разблокировать все элементы Unlock All Items на панели инструментов *Действия* чтобы вновь отредактировать элементы.

4.1.4 Идем дальше: Добавляем легенды

Читающий карту также должен иметь возможность видеть что на самом деле означают различные объекты на карте. В некоторых случаях, как и в названиях мест, это совершенно очевидно. В других случаях угадать сложнее, например, цвета леса. Добавим новую легенду.

1. Кликните на кнопку Добавить легенду Add Legend.
2. Кликните на страницу, чтобы разместить легенду, примите предложенные значения в диалоговом окне Свойства нового элемента *New Item Properties*.
3. Легенда добавлена на страницу макета, она отображает символы слоев, заданные в основном диалоговом окне.
4. Как обычно, вы можете кликнуть и переместить элемент в нужное место:



4.1.5 Идем дальше: Настраиваем элементы легенд

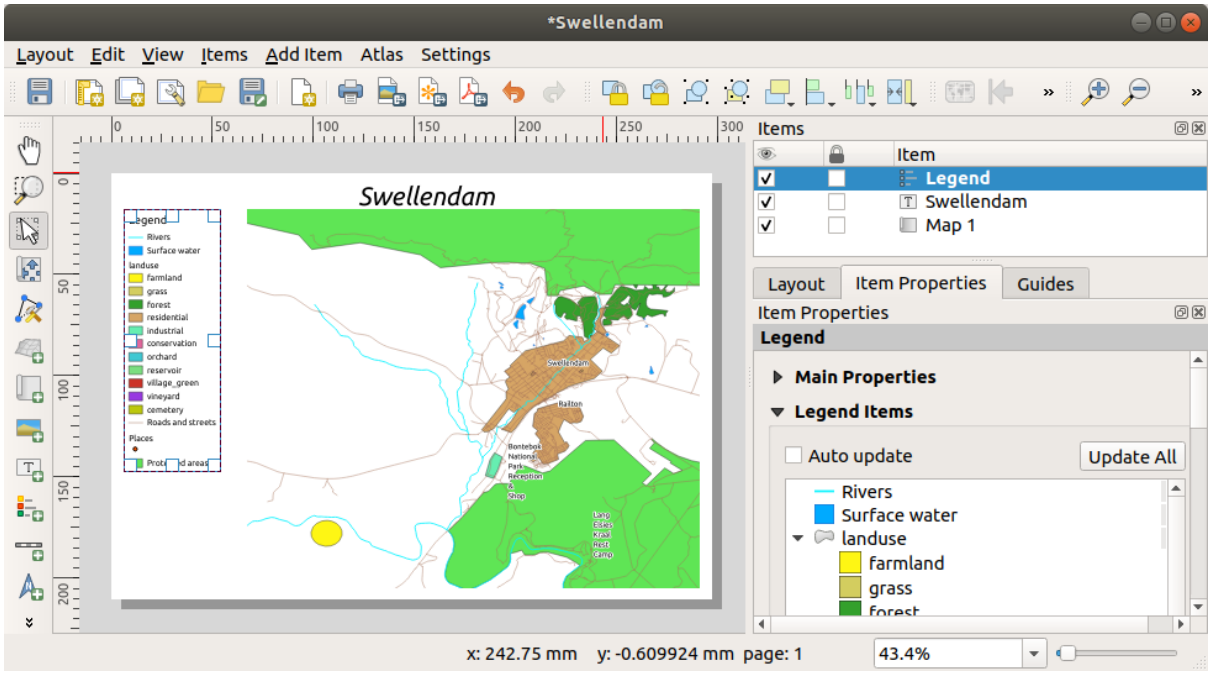
Не все в легенде, является обязательным, поэтому давайте удалим некоторые ненужные элементы.

1. Во вкладке *Item Properties* вы найдете группу Элементы легенд *Legend items*.
2. Снимите флажок с Автообновления *Auto update*, чтобы напрямую изменять элементы легенды.
3. Выберите *buildings*.
4. Удалите его из легенды, кликнув на кнопку .

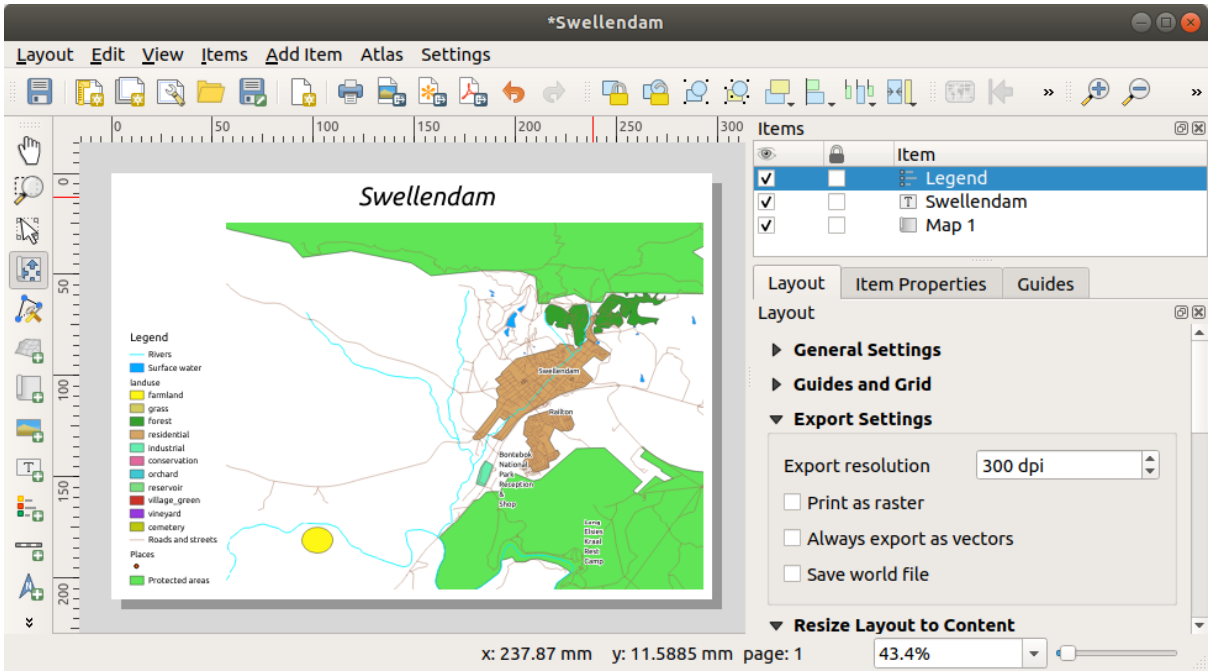
Вы также можете переименовывать элементы.

1. Выберите слой из того же списка.
2. Кликните на кнопку Редактировать свойства выбранного элемента Edit selected item properties.
3. Переименуйте слои `Places`, `Roads and Streets`, `Surface Water`, and `Rivers`.

Вы также можете изменить порядок элементов:



Поскольку легенда, вероятно, будет расширена новыми именами слоев, вы можете переместить и изменить размер легенды и / или карты. Вот результат:



4.1.6  Идем дальше: Экспортируем (выгружаем) карту

Примечание: Вы часто вспоминаете о сохранении вашей работы?

Наконец-то карта готова к выгрузке! Вы увидите кнопку экспортировать в верхнем левом углу окна макета:

- Печать макета  Print Layout: взаимодействует с принтером. Поскольку параметры принтера различаются в зависимости от модели принтера, с которой вы работаете, вероятно, лучше обратиться к руководству для принтера или к общему руководству по печати для получения дополнительной информации по данной теме.

  Другие кнопки позволяют экспортировать страницу карты в файл.

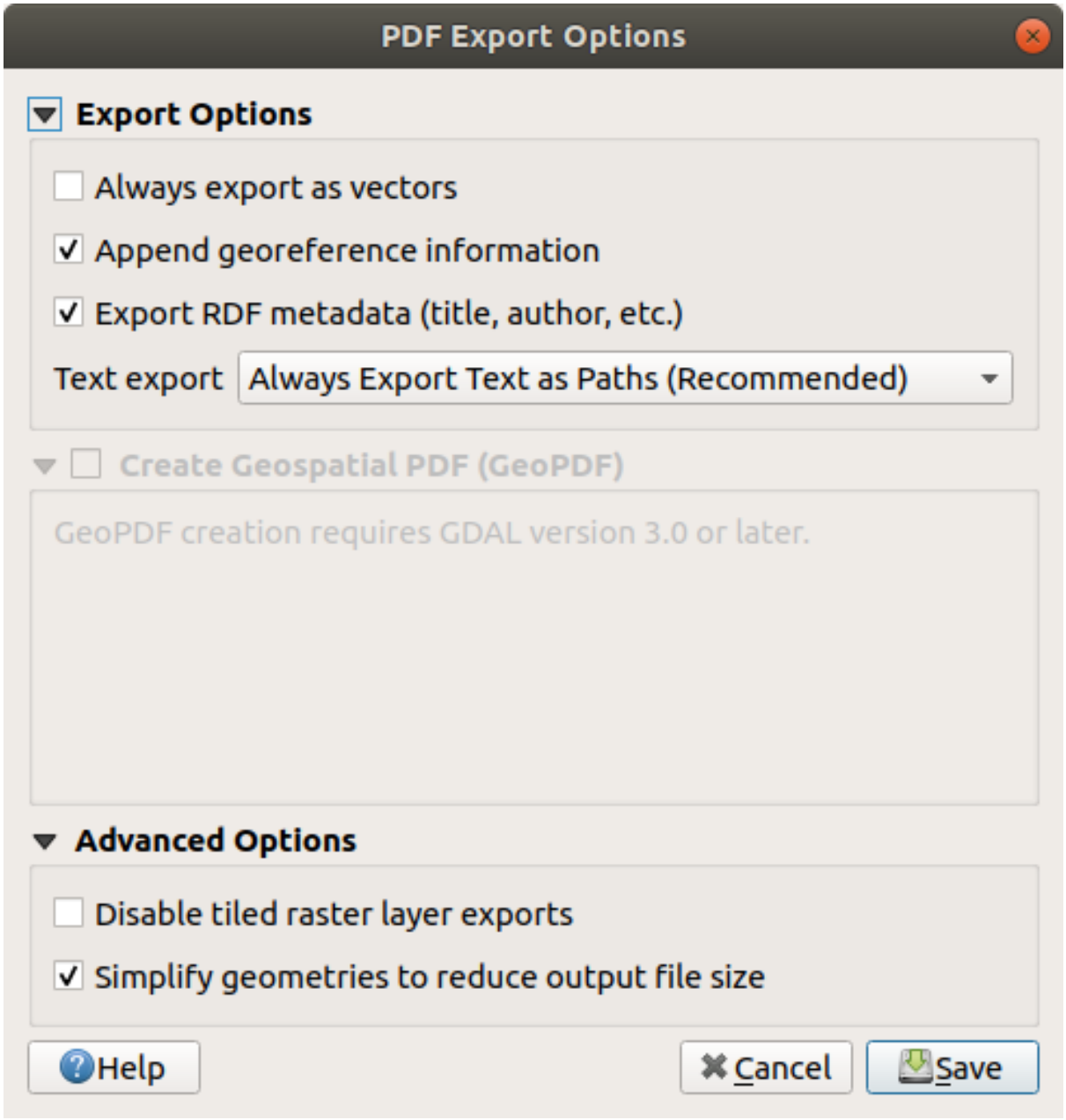
- Экспортировать как изображение  Export as Image:: дает вам на выбор различные распространенные форматы изображений. Это, наверное, самый простой вариант, но изображение, которое он создает, «мертвое» и его трудно редактировать.

- Экспортировать как SVG  Export as SVG: если вы отправляете карту картографу (которому может потребоваться отредактировать карту для публикации), лучше всего экспортировать ее в формате SVG. SVG означает «Масштабируемая векторная графика», также ее можно импортировать в такие программы, как `Inkscape `_ или другое программное обеспечение для редактирования векторных изображений.

- Экспортировать как PDF  Export as PDF: Если вам нужно отправить карту клиенту, чаще всего используется PDF-файл, поскольку для него проще настроить параметры печати. Некоторые картографы также могут предпочесть PDF, если у них есть программа, позволяющая импортировать и редактировать этот формат.

Для наших целей мы будем использовать PDF.

1. Кликните на Экспорт в PDF  Export as PDF.
2. Как обычно, выберите место для сохранения и имя файла. Появится следующее диалоговое окно.



3. Теперь вы можете безопасно использовать значения по умолчанию и кликнуть *Save*.

   QGIS продолжит экспорт карты. Кликните на сообщение в верхней части диалогового окна печати макета, как только экспорт завершится.

4. Кликните на гиперссылку в сообщении, чтобы открыть папку, в которой был сохранен PDF-файл, в файловом менеджере вашей системы.

5. Откройте его и посмотрите, как выглядит ваш макет.

   Все нормально? Поздравляем с первым завершенным проектом карты QGIS!

6. Вас что-то не удовлетворяет? Вернитесь в окно QGIS, внесите соответствующие изменения и снова выполните экспорт.

7. Не забудьте сохранить файл проекта.

4.1.7 В заключении

Теперь вы знаете, как создать базовый макет статичной карты. Мы можем пойти дальше и создать макет карты, который динамически адаптируется, с большим количеством элементов макета.

4.2 Занятие: Создание динамического макета печати

Теперь, когда вы научились создавать базовый макет карты, мы сделаем еще один шаг и создадим макет карты, который динамически адаптируется к размеру карты и свойствам страницы, например, при изменении размера страницы. Также дата создания будет динамически адаптироваться.

4.2.1  Идем дальше: Создаем динамическое Полотно Карты

1. Загрузите наборы данных формата ESRI Shapefile `protected_areas.shp`, `places.shp`, `rivers.shp` and `water.shp` в Полотно Карты и настройте их свойства для вашего удобства.
2. После того, как все будет отображено и обозначено символами на ваш вкус, кликните на иконку Новая печать макета  New Print Layout на панели инструментов или выберите *File* ► *New Print Layout*. Вам будет предложено выбрать заголовок для нового макета для печати.
3. Мы хотим создать макет карты, состоящий из заголовка и карты региона возле Свеллендама, Южная Африка. У макета должны быть края 7,5 мм и высота заголовка - 36 мм.
4. Создайте элемент карты называемый `main map` в полотне карты и перейдите на панель *Layout*. Прокрутите вниз до раздела Переменные *Variables* и найдите раздел *Layout*. Здесь мы установим некоторые переменные, которые вы можете использовать на всем динамическом макете для печати. Перейдите на панель *Layout* и прокрутите вниз до раздела *Variables*. Первая переменная будет определять края. Кликните на кнопку  и введите в строке имени `sw_layout_margin`. Установите значение `7.5`. Снова кликните на  и введите имя `sw_layout_height_header`. Установите значение `36`.
5. Теперь вы готовы автоматически создать положение и размер полотна карты с помощью переменных. Убедитесь, что ваш элемент карты выбран, перейдите на панель *Item Properties* , прокрутите вниз и откройте раздел Положение и размер *Position and Size*. Кликните на  Data defined override для *X* и из *Variables* выберите `@sw_layout_margin`.
6. Кликните на  Data defined override для *Y*, и выберите *Edit...* и введите формулу:

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

7. Вы можете задать размер элемента карты, используя переменные для ширины и высоты *Width* и *Height*. Клик на  Data defined override для *Width* и снова выберите *Edit* .... Заполните формулу:

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

Кликните на  Data defined override для *Height* и снова выберите *Edit* .... Заполните формулу:

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

8. Мы также создадим сетку, в которой содержатся координаты всей основной карты. Снова перейдите в *Item Properties* и выберите раздел *Grids*. Вставьте сетку, кликнув на . Кликните на *Modify grid* ... и установите *Interval* для *X*, *Y* и *Offset* в соответствии с масштабом карты, который вы выбрали на основном полотне QGIS. Перекрестный тип сетки *Grid type Cross* очень хорошо подходит для наших целей.

4.2.2  Идем дальше: Создаем динамический заголовок

1. Вставьте прямоугольник, в котором будет заголовок, с помощью кнопки  Add Shape. Введите имя заголовка `header` на панели *Items*.

2. Снова перейдите к *Item Properties* и откройте раздел *Position and Size*. Используя  Data defined override, выберите переменную `sw_layout_margin` как для *X*, так и для *Y*. Ширина *Width* должна определяться следующим выражением:

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

   а Высота *Height* переменной `sw_layout_height_header`.

3. Мы вставим горизонтальную линию и две вертикальные линии, чтобы разделить заголовок на разные секции при помощи Добавить элемент узла  Add Node Item. Создайте горизонтальную линию две вертикальные линии назовите их `Horizontal line`, `Vertical line 1` `Vertical line 2`.

   1. Для горизонтальной линии:

      1. Задайте *X* для переменной `sw_layout_margin`.
      2. Задайте выражение для *Y*:

```
@sw_layout_margin + 8
```

      3. Задайте выражение для *Width*:

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

   2. Для первой вертикальной линии:

      1. Задайте выражение для *X*:

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

      2. Задайте *Y* для переменной `sw_layout_margin`.
      3. Высота должна совпадать с созданным нами заголовком, поэтому задайте Высоту *Height* для переменной `sw_layout_height_header`.

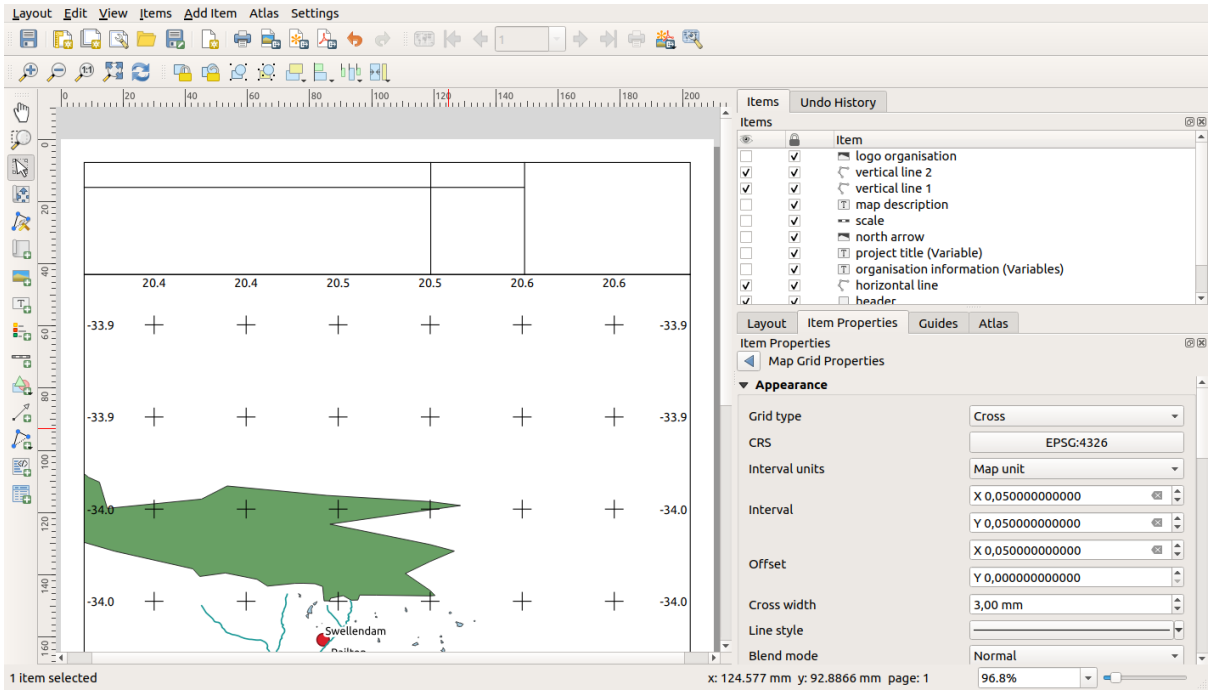
   3. Вторая вертикальная линия, установлена слева от первой.

      1. Задайте выражение для *X*:

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

      2. Задайте *Y* для переменной `sw_layout_margin`.
      3. Высота должна совпадать с другой вертикальной линией, поэтому задайте *Height* для переменной `sw_layout_height_header`.

   На рисунке ниже показана структура нашего динамического макета. Мы заполним некоторыми элементами те области, которые созданы линиями.



4.2.3 Идем дальше: Создаем надписи для динамического заголовка

1. Название вашего проекта QGIS может включаться автоматически. Название заголовка задается в Свойствах проекта *Project Properties*. Вставьте надпись с помощью кнопки Add Label и введите имя `project title (variable)`. В Основных свойствах *Main Properties* на панели Свойств элементов *Items Properties* введите следующее выражение:

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

Задайте положение надписи:

1. Для *X*, используйте выражение:

```
@sw_layout_margin + 3
```

2. Для *Y*, используйте выражение:

```
@sw_layout_margin + 0.25
```

3. Для *Width*, используйте выражение:

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

4. Введите `11.25` для *Height*.

В разделе Внешний вид *Appearance* установите размер шрифта 16 pt.

2. Вторая надпись содержит описание созданной вами карты. Повторно введите надпись и назовите ее `map description`. Описание карты. В *Main Properties* введите текст `map description`. Мы также включим в *Main Properties*:

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

Здесь мы использовали две функции `Date and Time` functions (`now` и `format_date`).

Задайте положение надписи:

1. Для *X*, используйте выражение:

```
@sw_layout_margin + 3
```

2. Для *Y*, используйте выражение:

```
@sw_layout_margin + 11.5
```

3. Третья надпись включает информацию о вашей организации. Сначала мы создадим несколько переменных в меню *Variables* из *Item Properties*. Перейдите в меню *Layout* каждый раз кликайте на кнопку  и вводите названия `o_department`, `o_name` , `o_adress` и `o_postcode`. Во второй строке введите информацию о вашей организации. Мы будем применять эти переменные в разделе *Main Properties*.

   В разделе *Main Properties* введите следующее:

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

   Задайте положение надписи:

   1. Для *X*, используйте выражение:

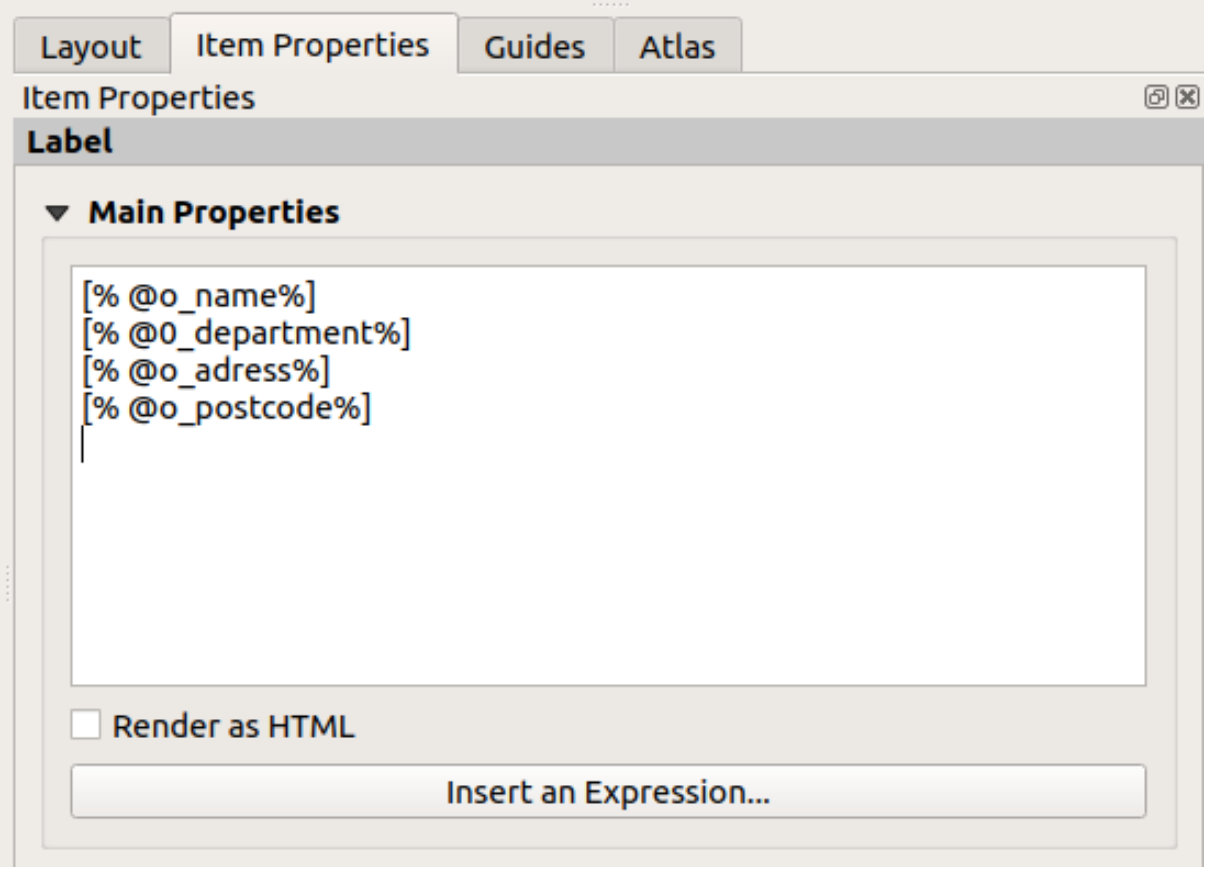
```
@layout_pagewidth - @sw_layout_margin - 49.5
```

   2. Для *Y*, используйте выражение:

```
@sw_layout_margin + 15.5
```

   3. Для *Width* используйте `49.00`.
   4. Для *Height* используйте выражение:

```
@sw_layout_height_header - 15.5
```



4.2.4 Идем дальше: Добавляем изображения к динамическому заголовку

1. Используйте кнопку Add Picture, чтобы разместить картинку над вашей надписью `organisation information`. После ввода названия `organisation logo`, определите положение и размер логотипа:

   1. Для *X*, используйте выражение:

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

   2. Для *Y*, используйте выражение:

      ```
      @sw_layout_margin + 3.5
      ```

   3. Для *Width* используйте `39.292`.

   4. Для *Height* используйте `9.583`.

   Чтобы добавить логотип вашей организации, вы должны сохранить его в своем домашнем каталоге и ввести путь в *Main Properties* ► *Image Source*.

2. Нашему макету по-прежнему нужна стрелка, указывающая на север. Стрелку можно вставить также с помощью Добавить стрелку на север Add North Arrow. Мы будем использовать стрелку на север по умолчанию. Определите положение:

   1. Для *X*, используйте выражение:

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

2. Для *Y*, используйте выражение:

```
@sw_layout_margin + 9
```

3. Для *Width* используйте `21.027`.
4. Для *Height* используйте `21.157`.

4.2.5 Идем дальше: Создаем шкалу динамического заголовка

1. Чтобы вставить шкалу в заголовок, кликните на кнопку Add Scale Bar и поместите ее в прямоугольник над стрелкой на север. В *Map* в разделе Основные свойства *Main Properties* выберите `main map(Map 1)`. Это означает, что масштаб изменяется автоматически согласно размеру протяженности, которую вы выбрали на основном полотне QGIS. Выберите *Style Numeric*. Это означает, что мы вводим простой масштаб без шкалы. Так как для шкалы нужно положение и размер.
   1. Для *X*, используйте выражение:

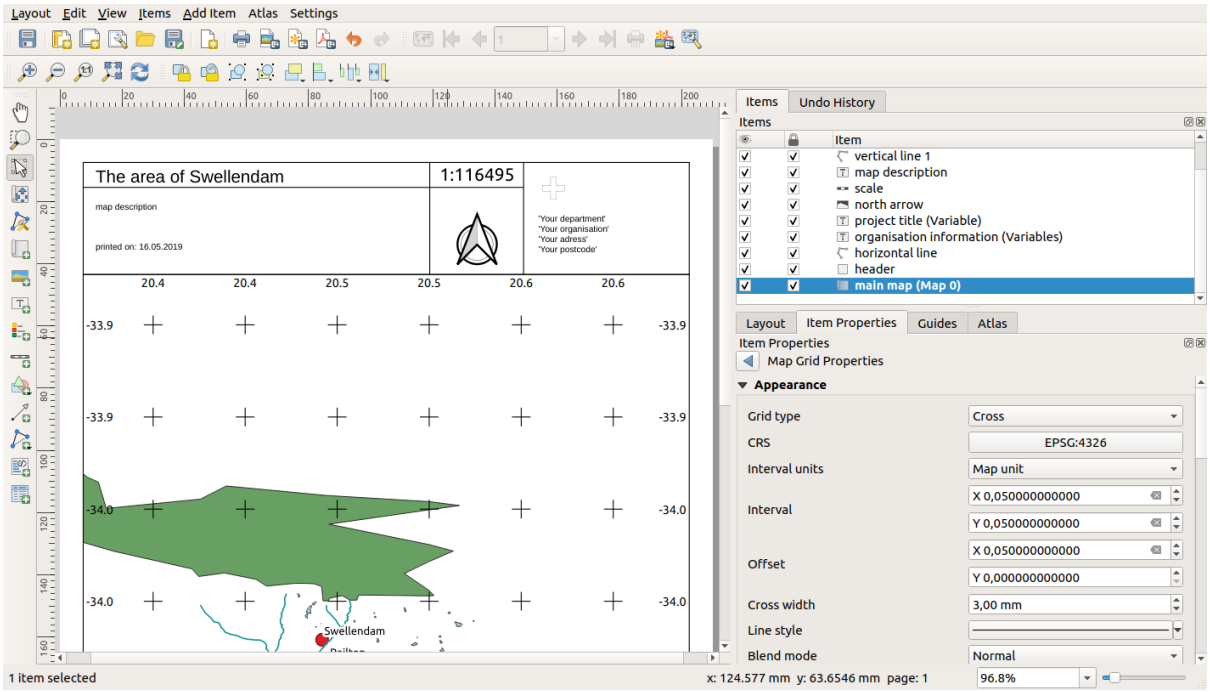
```
@layout_pagewidth - @sw_layout_margin * 2 - 78
```

   2. Для *Y*, используйте выражение:

```
@sw_layout_margin + 1
```

   3. Для *Width* используйте `25`.
   4. Для *Height* используйте `8`.
   5. Разместите `Reference point` по центру.

Поздравляю! Вы создали свой первый динамический макет карты. Взгляните на макет и проверьте, все ли выглядит так, как вы хотели! Макет динамической карты автоматически реагирует на изменение Свойств страницы *page properties*. Например, если вы измените размер страницы с DIN A4 на DIN A3, кликните на Обновить просмотр Refresh view, и дизайн страницы будет адаптирован.



4.2.6 Что дальше?

На следующей странице вам будет предложено выполнить задание. Это позволит вам практиковать методы, которые вы уже изучили.

4.3 Задание 1

Откройте ваш существующий проект карты и тщательно его отредактируйте. Если вы заметили мелкие ошибки или вещи, которые хотели исправить ранее, сделайте это сейчас.

При настройке карты продолжайте задавать себе вопросы. Легко ли читать и понимать эту карту тем, кто не знаком с данными? Если бы я увидел эту карту в Интернете, на плакате или в журнале, привлекла бы она мое внимание? Хотел бы я прочитать эту карту, если бы она не была моей?

Если вы изучаете данный курс на  базовом или  среднем уровне, ознакомьтесь с методами из более «продвинутых» разделов. Если вы видите на карте что-то, что вы хотели бы сделать, почему бы не попробовать реализовать это?

Если вам преподают данный курс, то преподаватель курса может потребовать, чтобы вы отправили окончательную версию вашей карты, экспортированную в PDF, для оценки. Если вы проходите данный курс самостоятельно, рекомендуется оценить собственную карту по тем же критериям. Ваша карта будет оцениваться по общему виду и символике самой карты, а также по внешнему виду и макету страницы и элементам карты. Помните, что при оценке внешнего вида карт упор всегда делается на *простоту использования*. Чем лучше карта выглядит и чем легче ее понять с первого взгляда, тем она лучше.

Удачной настройки!

4.3.1 В заключении

В первые четырех модулях вы научились создавать и стилизовать векторную карту. В следующих четырех модулях вы узнаете, как использовать QGIS для проведения полного анализа GIS. Сюда входят создание и редактирование векторных данных; анализ векторных данных; использование и анализ растровых данных; и использование GIS для решения проблемы от начала до конца, используя как растровые, так и векторные источники данных.

ГЛАВА 5

Module: Модуль: Создание векторных данных

Создание карт при помощи существующих данных – это только начало. В этом модуле вы узнаете, как модифицировать существующие векторные данные и как полностью создавать новые наборы данных.

5.1 Занятие: Создание нового набора векторных данных

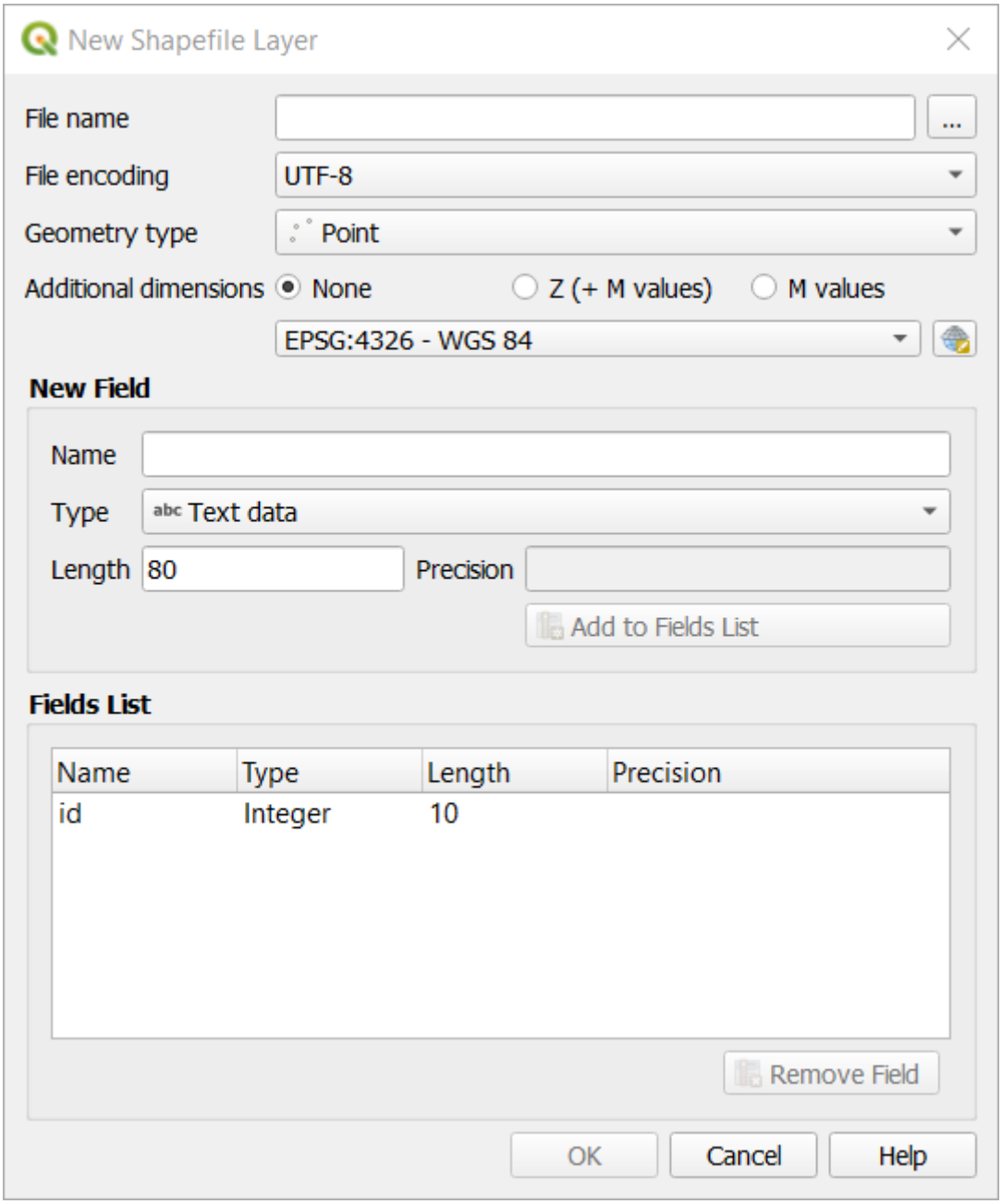
Данные, которые вы используете, должны откуда-то поступать. Данные уже существуют для наиболее распространенных приложений; но чем более конкретен и специализирован проект, тем меньше вероятность того, что данные уже будут доступны. В таких случаях вам нужно будет создать свои собственные новые данные.

Цель данного занятия: создать новый набор векторных данных.

5.1.1  Идем дальше: Диалоговое окно для создания слоя

Прежде чем вы сможете добавить новые векторные данные, вам понадобится набор векторных данных, для того чтобы их добавить. В нашем случае вы начнете с создания абсолютно новых данных, а не с редактирования существующего набора данных. Следовательно, вам сначала необходимо определить свой собственный новый набор данных.

1. Откройте QGIS и создайте новый пустой проект.
2. Перейдите и кликните меню, войдите в *Layer* ► *Create Layer* ► *New Shapefile Layer* и кликните его. Вам будет представлено диалоговое окно Новый слой шейп-файла *New Shapefile Layer*, что позволит вам определить новый слой.

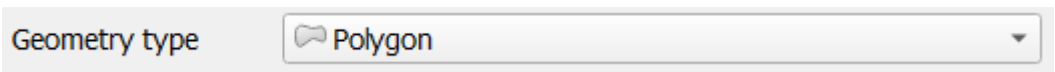


3. Кликните на … в поле *File name*. Появится диалоговое окно сохранения.
4. Перейдите в каталог `exercise_data`.
5. Сохраните новый слой как `school_property.shp`.

   На этом этапе важно решить, какой тип набора данных вам нужен. Каждый векторный тип слоя «строится по-своему» в фоновом режиме, поэтому после создания слоя вы не можете изменить его тип.

   В следующем упражнении мы собираемся создать новые функции, описывающие местности. Вам потребуется создать набор данных полигона для таких функций.

6. В поле Тип геометрии *Geometry Type*, выберите Полигон *Polygon* в раскрывающемся меню:

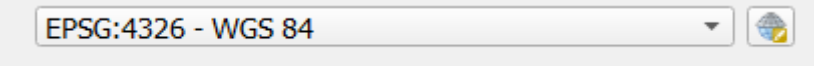


Это не влияет на остальную часть диалогового окна, но приведет к использованию правильного типа

геометрии при создании набора векторных данных.

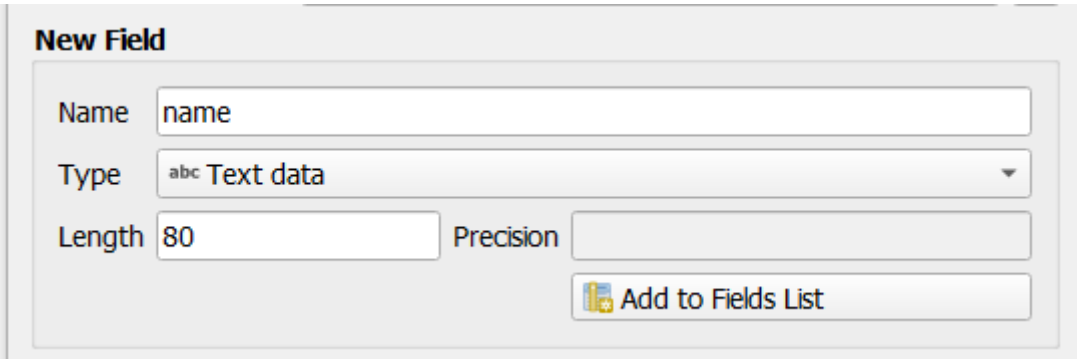
В следующем поле можно указать Систему координат (СК-CRS). ССК - это метод связывания числовых координат с положением на поверхности Земли. Чтобы узнать больше, см. Руководство пользователя по работе с прогнозами - Working with Projections.

В данном примере мы будем использовать, связанный с этим проектом, ССК по умолчанию, то есть WGS84.

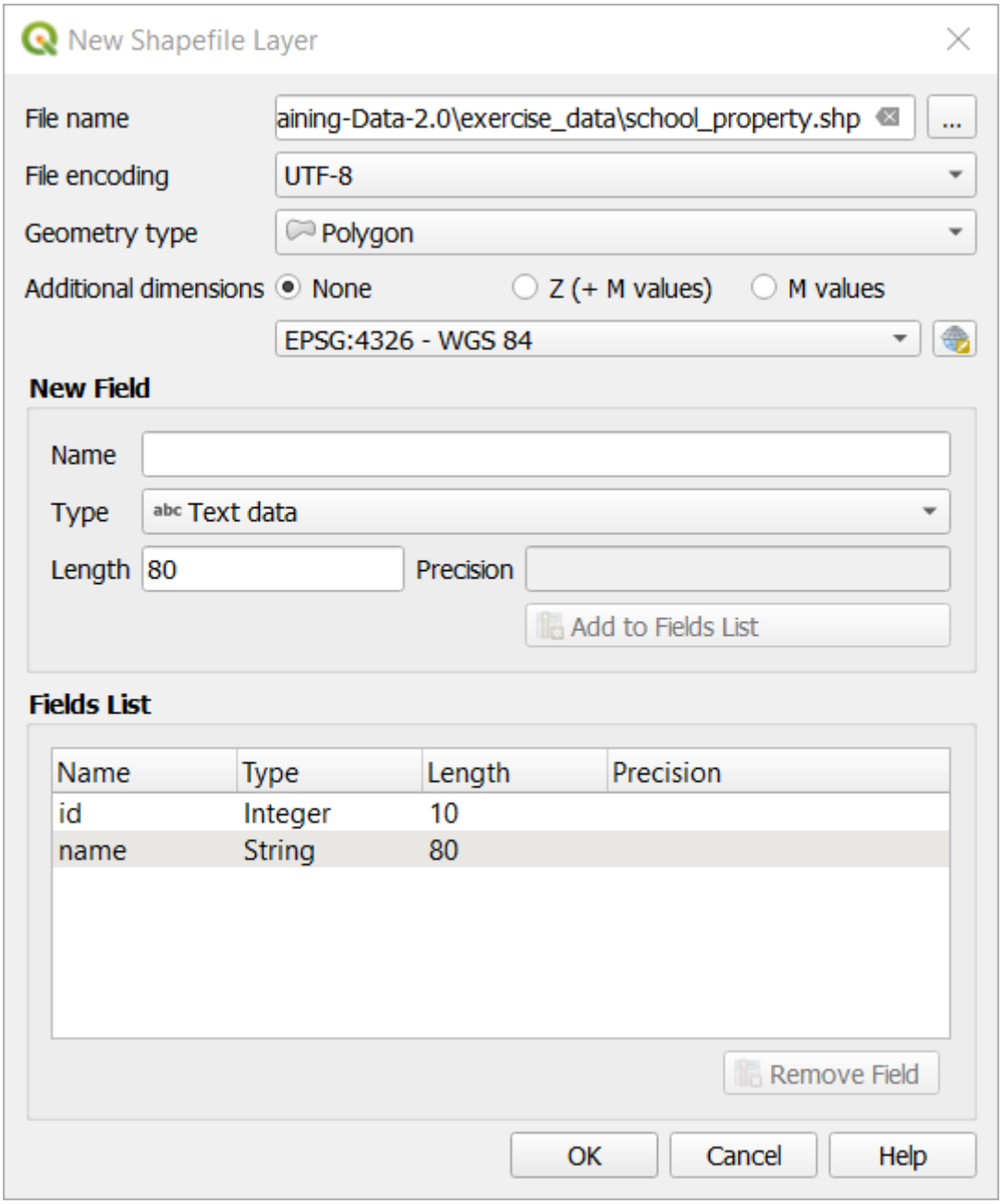


Далее идет набор полей, сгруппированных в Новом поле *New Field*. По умолчанию, новый слой имеет только один атрибут – поле `id` (которое вы должны увидеть в списке полей *Fields list*) ниже. Однако для того, чтобы создаваемые вами данные были полезными, вам действительно нужно что-то описать о функциях, которые вы будете создавать в данном новом слое. Для наших текущих целей будет достаточно добавить одно поле, которое называется `name` , которое будет содержать `Text data` и будет ограничено длиной текста в `80` символов.

7. Повторите настройку ниже, затем кликните кнопку Добавить в список полей *Add to Fields List*:



8. Убедитесь, что ваше диалоговое окно выглядит следующим образом:



9. Кликните *OK*

Должен появиться новый слой на вашей панели *Layers*.

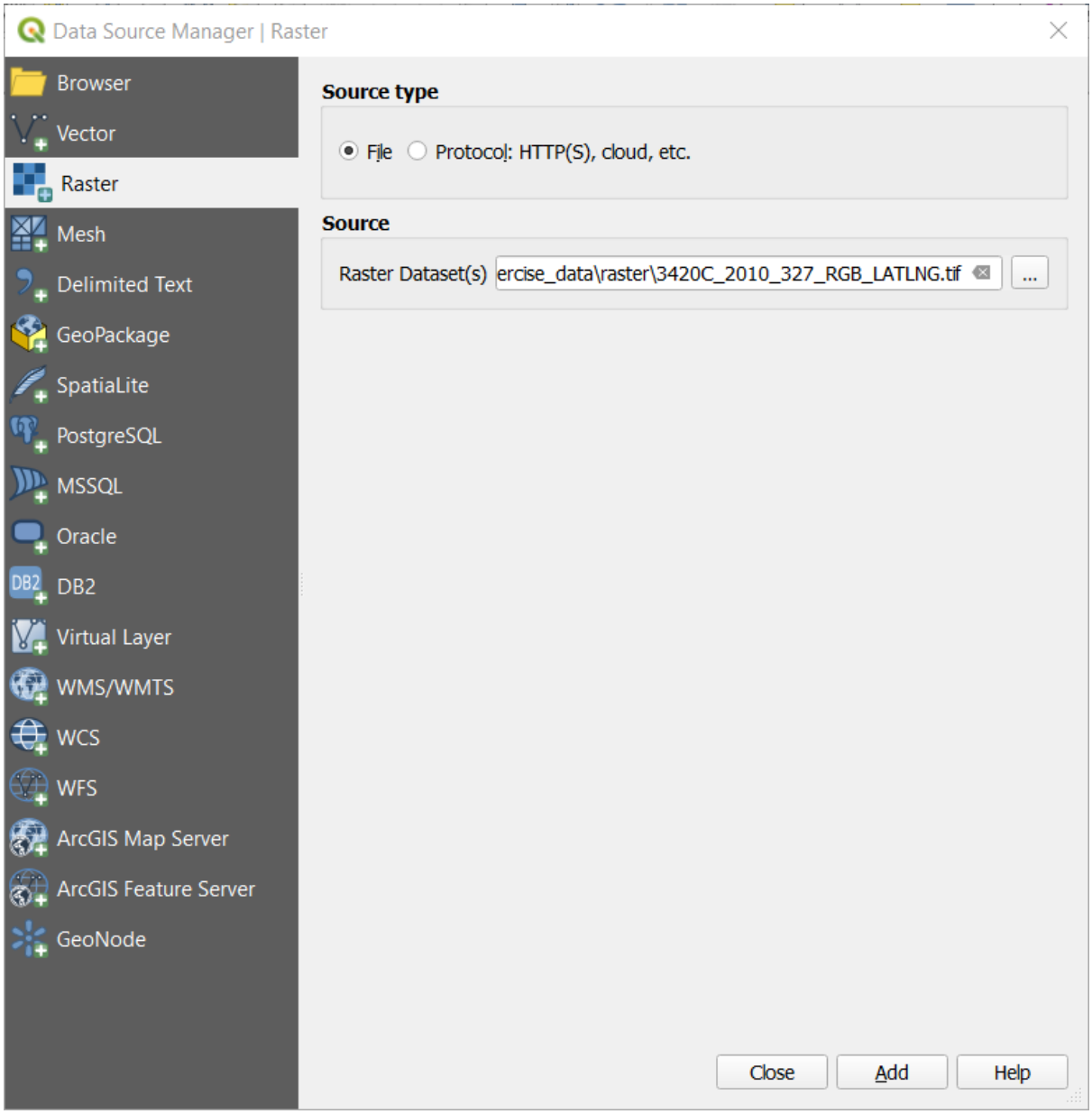
5.1.2 Идем дальше: Источник данных

Когда вы создаете новые данные, очевидно, что речь идет об объектах, которые действительно существуют на земле. Следовательно, вам нужно будет откуда-то получать информацию.

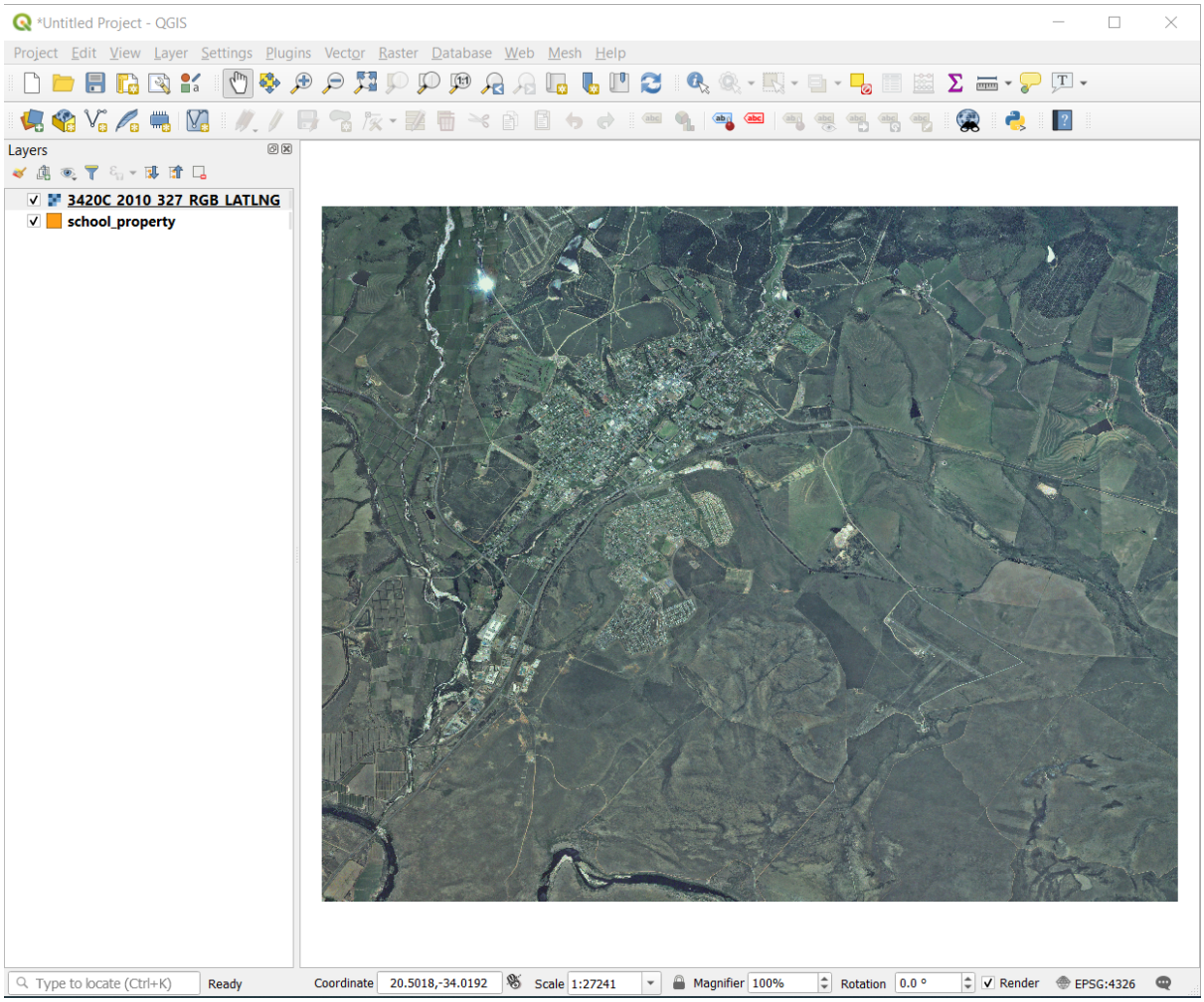
Существует много разных способов получить данные об объектах. Например, вы можете использовать GPS для съемки точек в реальном мире, а затем импортировать данные в QGIS. Или вы можете обследовать точки с помощью теодолита и ввести координаты вручную, чтобы создать новые функции. Или вы можете использовать процесс оцифровки для отслеживания объектов по данным дистанционного зондирования, таким как спутниковые изображения или аэрофотосъемка.

В нашем примере вы будете использовать подход оцифровки. Вам предоставлены образцы наборов растровых данных, поэтому, при необходимости, вам нужно будет их импортировать.

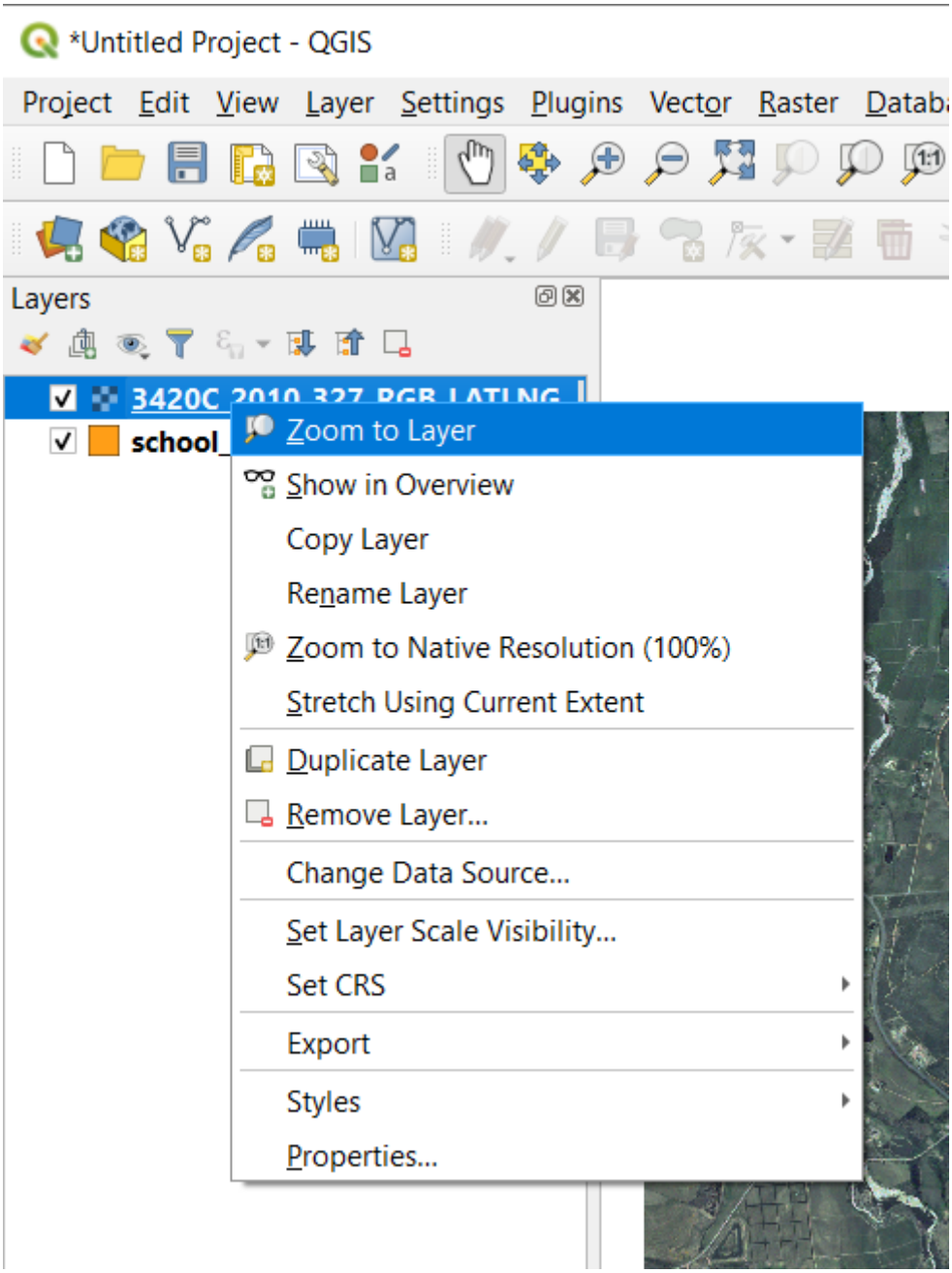
1. Кликните на Диспетчер источников данных  Data Source Manager.
2. Выберите Растр  *Raster* слева.
3. На панели Источник *Source*, кликните кнопку …:
4. Перейдите в файл упражнений `exercise_data/raster/`.
5. Выберите файл `3420C_2010_327_RGB_LATLNG.tif`.
6. Кликните *Open* , чтобы закрыть диалоговое окно.



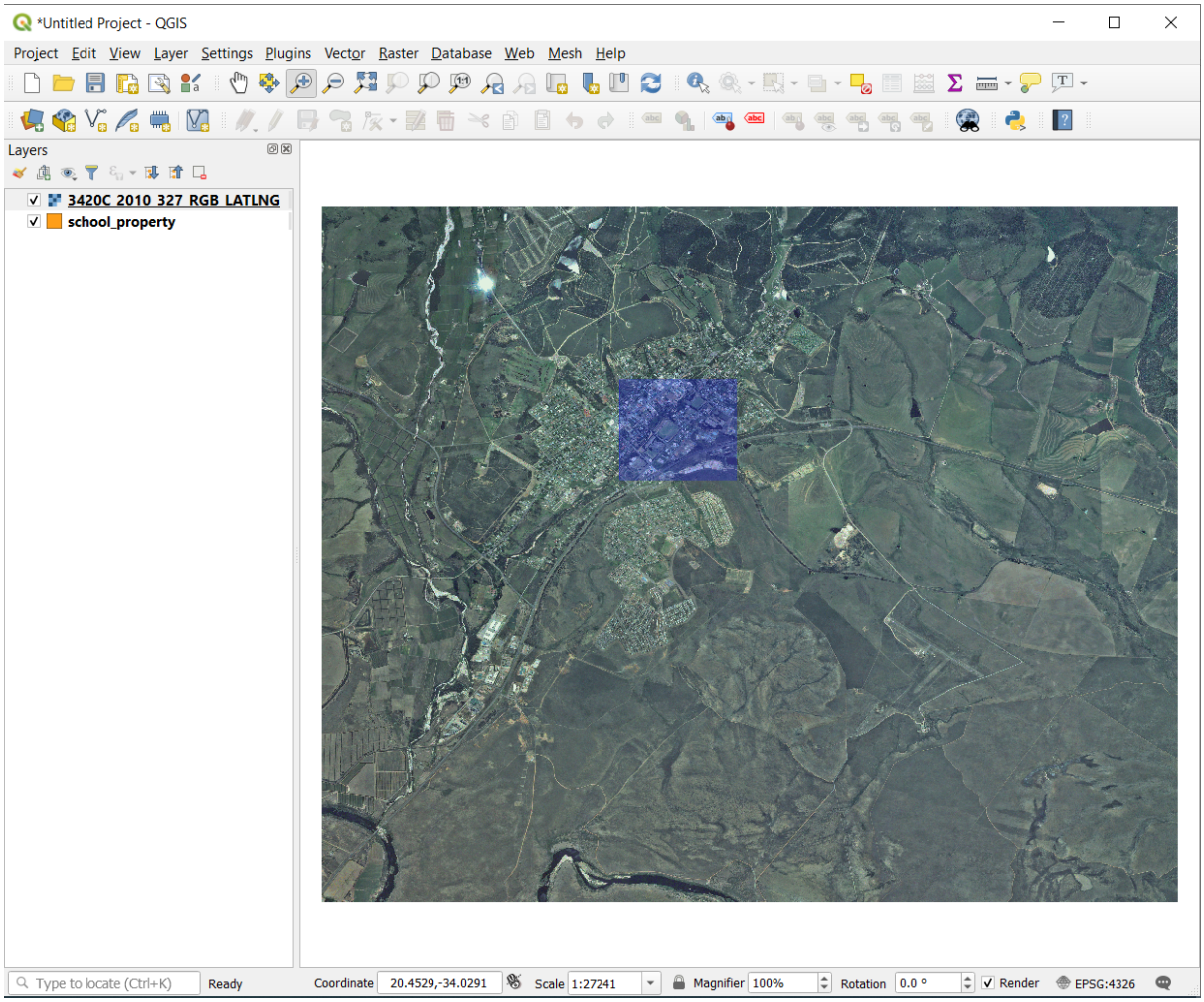
7. Кликните *Add* и *Close*. Изображение загрузится на вашу карту.



8. Если вы не видите аэрофотоснимок, выберите новый слой, кликните правой кнопкой мыши и выберите Приблизить до слоя *Zoom to Layer*. Layer в контекстном меню.



9. Кликните на кнопку Zoom In, и приближайте масштаб до местности, выделенной синим цветом ниже:

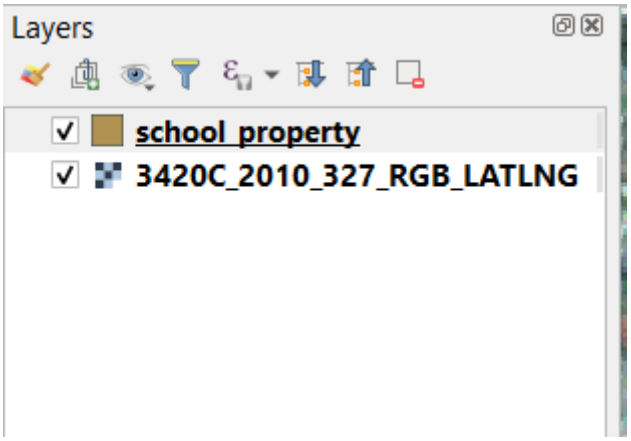


Теперь вы готовы оцифровать эти три поля:



Прежде чем вы начнете оцифровку, давайте передвинем слой `school_property` над аэрофотоснимком.

1. Выберите слой `school_property` на панели *Layers* и перетащите его наверх.



Чтобы начать оцифровку, вам нужно войти в **режим редактирования**. Программное обеспечение GIS

обычно требует этого, чтобы предотвратить случайное редактирование или удаление важных данных. Режим редактирования включается или выключается индивидуально для каждого слоя.

Чтобы войти в режим редактирования для слоя `school_property` необходимо:

1. Кликнуть слой `school_property` на панели *Layers*, чтобы выбрать его.
2. Кликните на кнопку  Toggle Editing.

   Если вы не можете найти эту кнопку, убедитесь, что панель инструментов Оцифровка *Digitizing* включена. Рядом с пунктом меню *View* ► *Toolbars* ► *Digitizing* Digitizing должен быть установлен флажок.

   Как только вы войдете в режим редактирования, вы увидите, что некоторые инструменты оцифровки стали активными:

   - Захватить полигон  Capture Polygon
   - Инструмент Vertex  Vertex Tool

   Другие соответствующие кнопки по-прежнему неактивны, но станут активными, когда мы начнем взаимодействовать с нашими новыми данными.

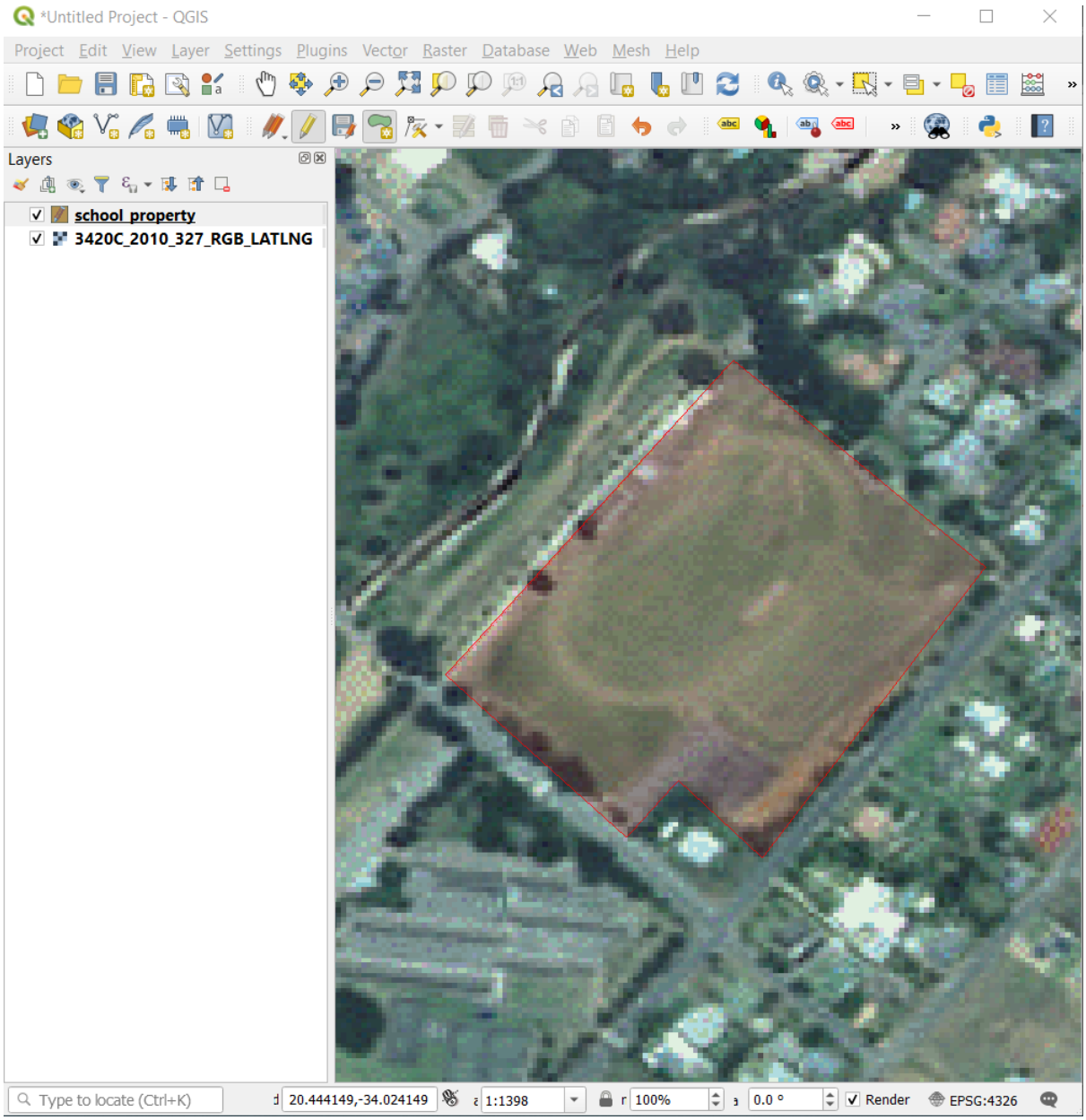
   Обратите внимание, что у слоя `school_property` на панели *Layers* теперь есть значок карандаша, указывающий, что он находится в режиме редактирования.
3. Кликните на кнопку  Capture Polygon чтобы начать оцифровку школьных полей.

   Вы заметите, что курсор мыши превратился в перекрестие. Это позволяет более точно размещать точки, которые вы будете оцифровывать. Помните, что даже когда вы используете инструмент оцифровки, вы можете приближать и отдалять вашу карту, прокручивая колесико мыши, и вы можете перемещаться, удерживая колесико мыши и перетаскивая карту.

   Первое, что вам нужно оцифровать - это спортивная площадка:

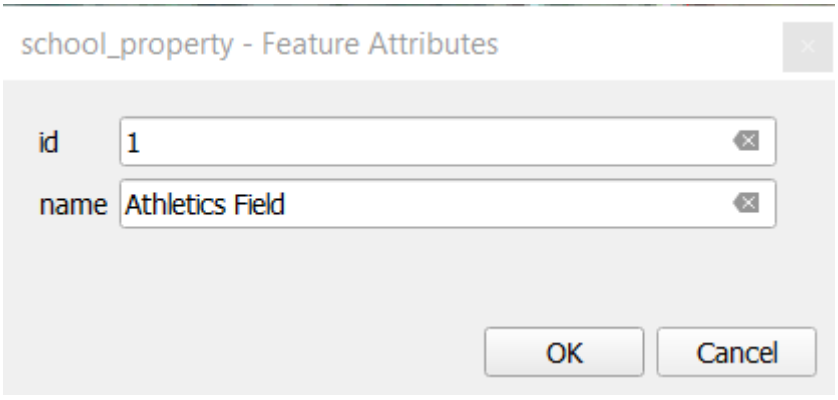


4. Начните оцифровку, кликнув точку где-нибудь на краю поля.
5. Разместите больше точек, кликая дальше по краю, пока фигура, которую вы рисуете, полностью не покроет поле.

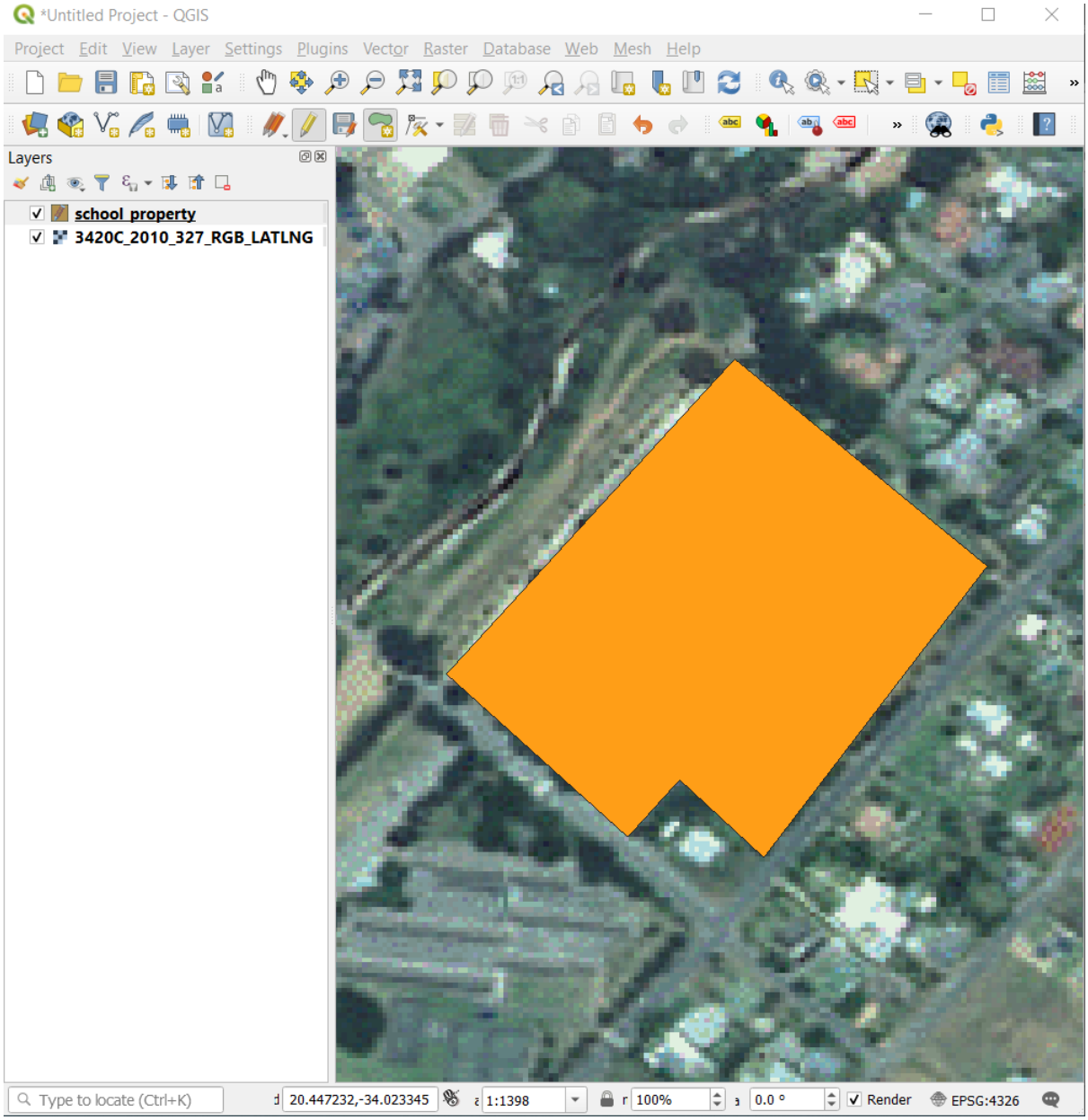


6. После размещения последней точки кликните правой кнопкой мыши, чтобы закончить рисование полигона. Этим вы завершите функцию и увидите диалоговое окно *Attributes*.

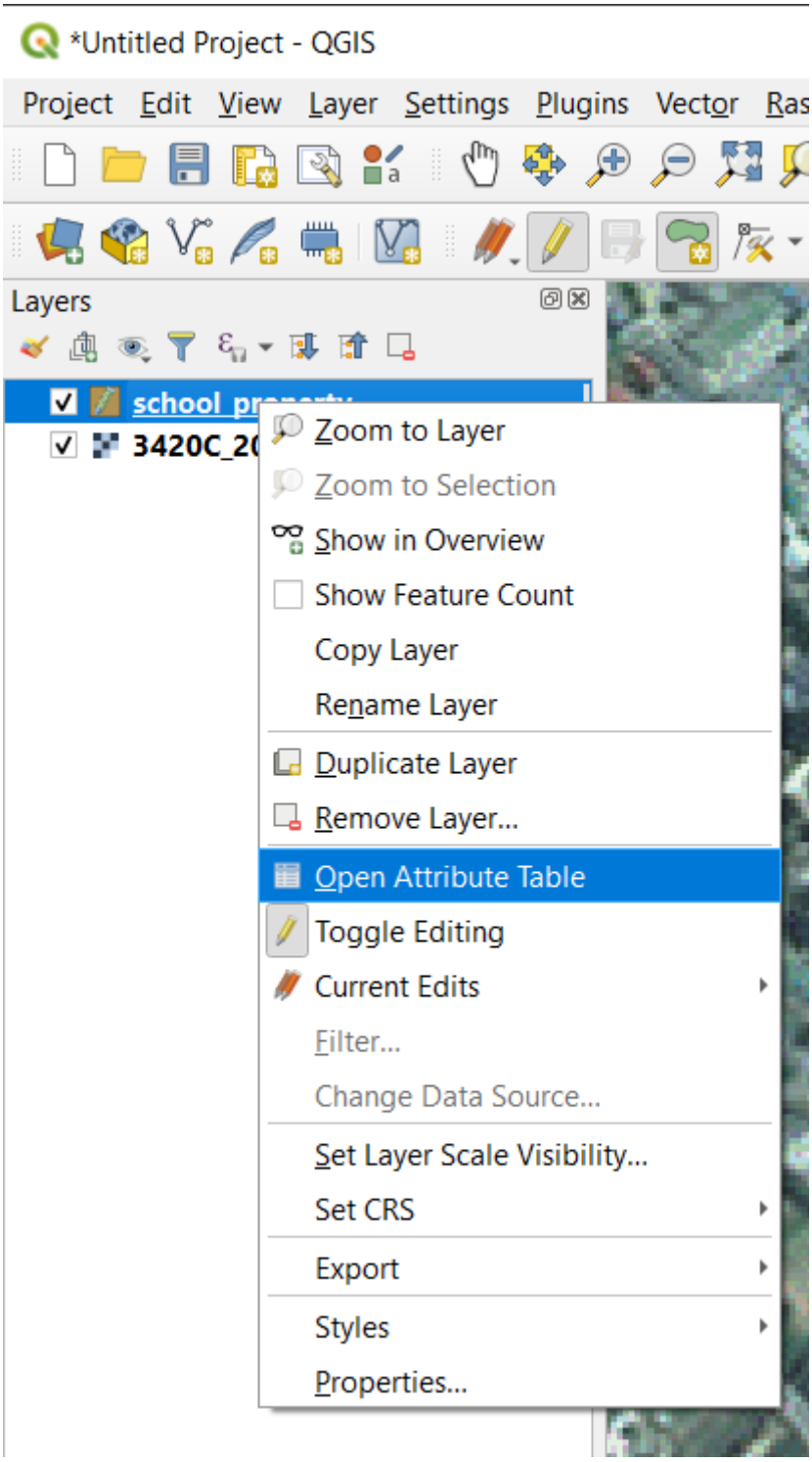
7. Введите значения, как показано ниже:



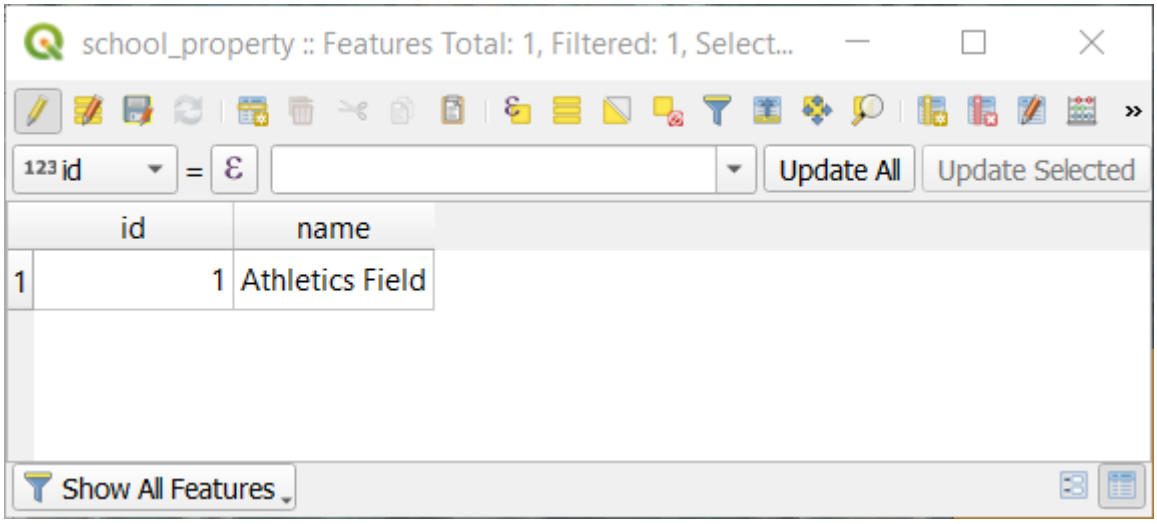
8. Кликните *OK*. Вы создали новую функцию!



9. На панели *Layers* выберите слой `school_property`.
10. Кликните правой кнопкой мыши и выберите *Open Attribute Table* в контекстном меню.



В таблице вы увидите только что добавленную функцию. В режиме редактирования вы можете обновить данные атрибутов, дважды кликнув ячейку, которую вы хотите обновить.



11. Закройте таблицу атрибутов.
12. Кликните кнопку Сохранить изменения Save Edits, чтобы сохранить новую функцию, которую мы только что создали.

Помните, что если вы допустили ошибку при оцифровке объекта, вы всегда можете отредактировать ее после того, как закончите ее создавать. Если вы ошиблись, продолжайте оцифровку, пока не закончите создание объекта, как указано выше. Далее:

1. Кликните кнопку Vertex Tool.
2. Наведите курсор мыши на Vertex, который вы хотите переместить, и кликните на него левой кнопкой мыши.
3. Переместите мышь в правильное место Vertex и кликните левой кнопкой мыши. Это переместит Vertex в новое место.





Эту же процедуру можно использовать для перемещения линейного сегмента, но вам нужно будет навести курсор мыши на среднюю точку линейного сегмента.

IЕсли вы хотите отменить изменение, вы можете кликнуть на кнопку Undo или `Ctrl+Z`.

4. Не забудьте сохранить изменения, кликнув на Save Edits.
5. По завершении редактирования кликните на Переключить редактирование Toggle Editing, чтобы выйти из режима редактирования.

5.1.3  Попробуйте сами, оцифруйте полигоны

Оцифруйте саму школу и поле вверху. Используйте данное изображение для подсказки:



Помните, что каждая новая функция должна иметь уникальное значение `id`!

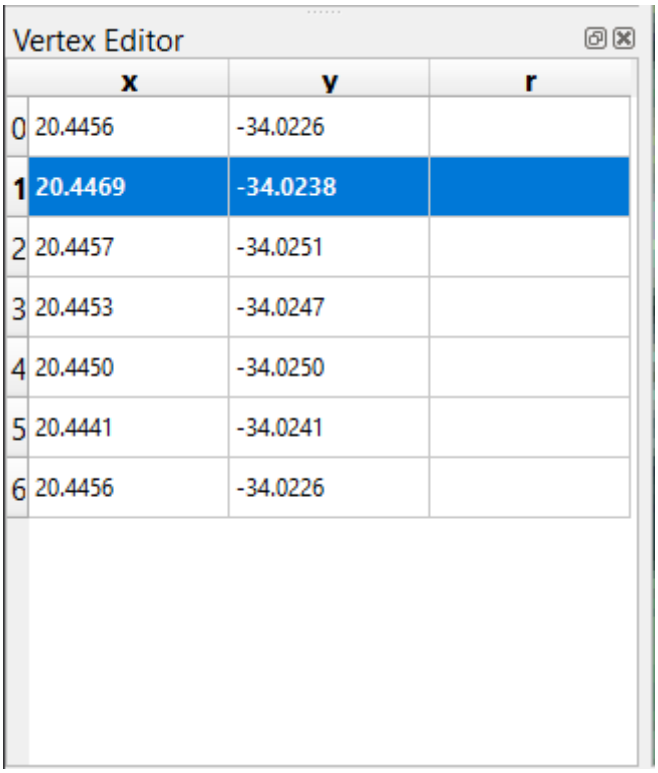
Примечание: Когда вы закончите добавлять функции к слою, не забудьте сохранить изменения, а затем выйти из режима редактирования.

Примечание: Вы можете стилизовать заливку, контур и размещение надписей и форматирование `school_property` используя приемы, изученные на более раннем занятии.

5.1.4 Идем дальше: Использование таблицы редактора Vertex

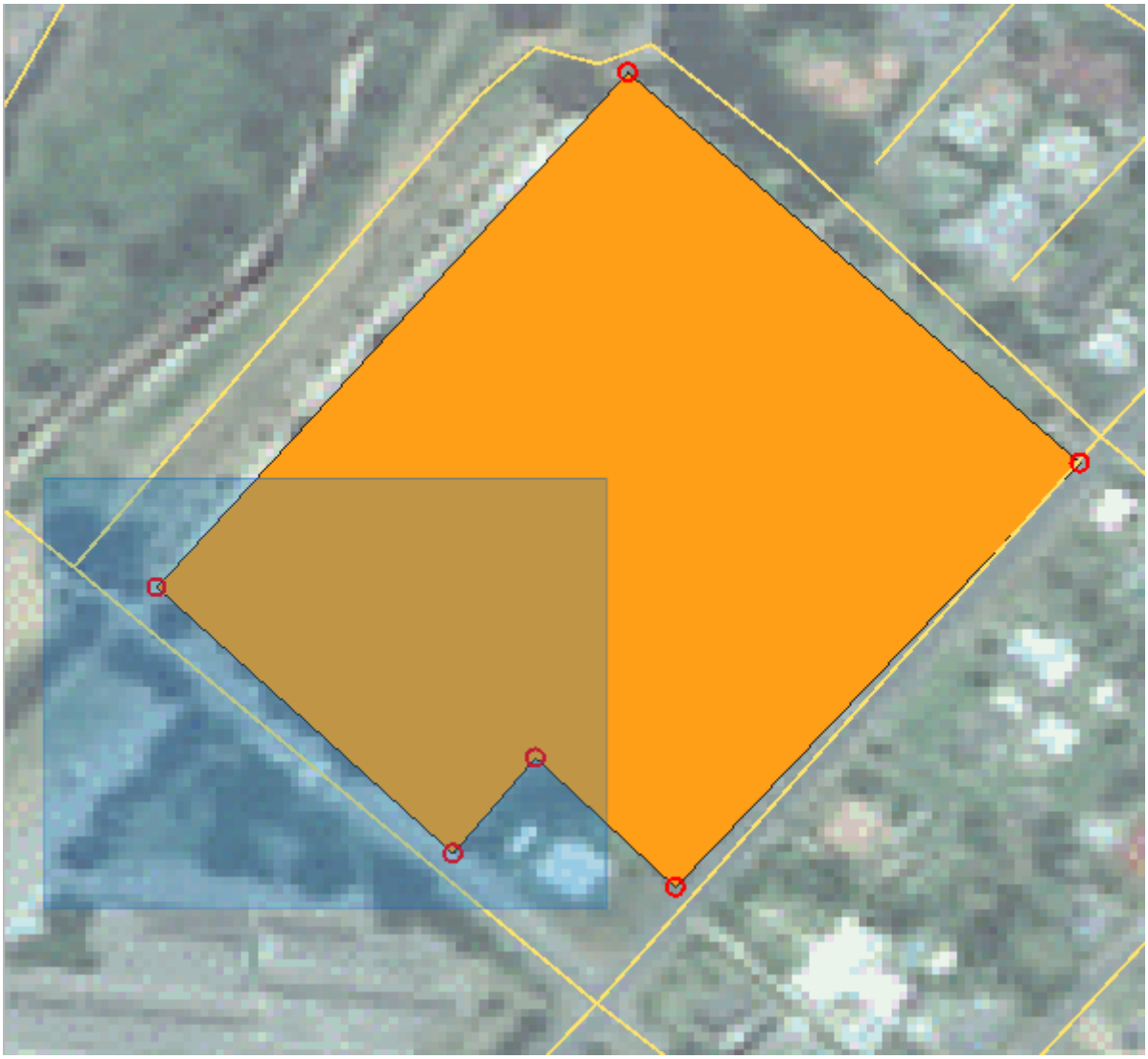
Другим способом редактирования функции является ввод вручную фактических значений координат для каждой вершины с помощью *Vertex Editor*.

1. Убедитесь, что вы находитесь в режиме редактирования на слое `school_property`.
2. Если она еще не активирована, кликните кнопку Vertex Tool.
3. Наведите курсор мыши на одну из функций полигонов, созданных на слое `school_property` и кликните правую кнопку. Таким образом будет выбрана функция и появится панель *Vertex Editor*.

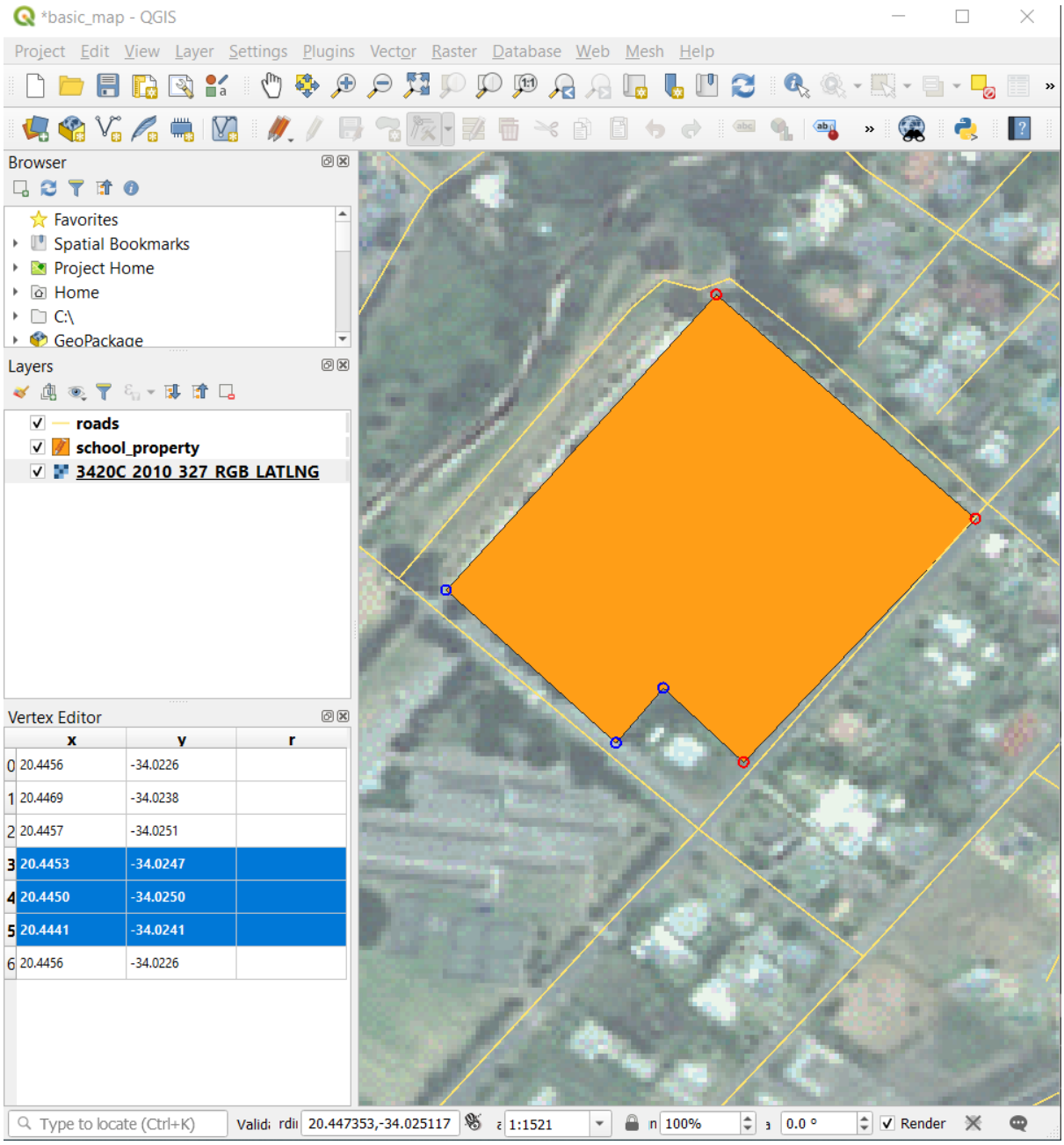


Примечание: Эта таблица содержит координаты вершин элемента. Обратите внимание, что в этой функции имеется семь вершин, но только шесть визуально идентифицированы в полотне карты. При ближайшем осмотре можно заметить, что строки 0 и 6 имеют идентичные координаты. Они являются начальной и конечной вершинами геометрии функции и необходимы для создания замкнутой функции полигона.

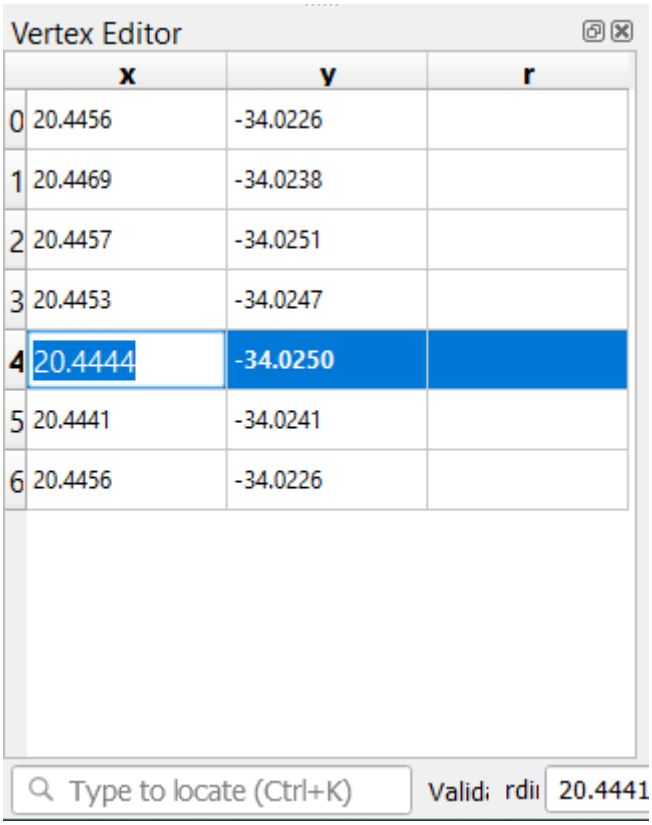
4. Кликните и перетащите рамку над вершиной или несколькими вершинами выбранной функции.



Выбранные вершины поменяют цвет на голубой, а в таблице *Vertex Editor* будут выделены соответствующие строки, содержащие координаты вершин.



5. Чтобы обновить координату, дважды кликните левой кнопкой мыши на ячейку в таблице, которую требуется отредактировать, и введите обновленное значение. В этом примере координата x строки `4` обновлена из `20.4450` на `20.4444`.



6. После ввода обновленного значения кликните клавишу Enter, чтобы применить изменение. Вершина будет перемещена в новое расположение в окне карты.

7. Когда закончите редактирование кликните Toggle Editing, чтобы выйти из режима редактирования, сохраните свои изменения.

5.1.5 Попробуйте сами: Оцифруйте линии

Мы собираемся оцифровать два маршрута, которые еще не отмечены на слое дорог: один это путь, другой - трек. Наш путь проходит вдоль южного края пригорода Райлтона, начинаясь и заканчиваясь отмеченными дорогами:



Наш трек немного дальше к югу:



1. Если слой *дороги* еще не указан на вашей карте, добавьте слой `roads` из файла GeoPackage `training-data.gpkg`, который включен в папку `exercise_data` вашей учебной базы данных, которую вы загрузили. Прочтите главу *Идем дальше: Загружаем векторные данные из Базы данных GeoPackage* для инструкций.
2. Создайте новый набор данных строки Shapefile ESRI с именем `routes.shp` в директории `exercise_data` , с атрибутами `id` и `type` (используйте приведенный выше подход, чтобы научиться).
3. Активируйте режим редактирования на слое *routes*.
4. Поскольку вы работаете с функцией линии, кликните кнопку 

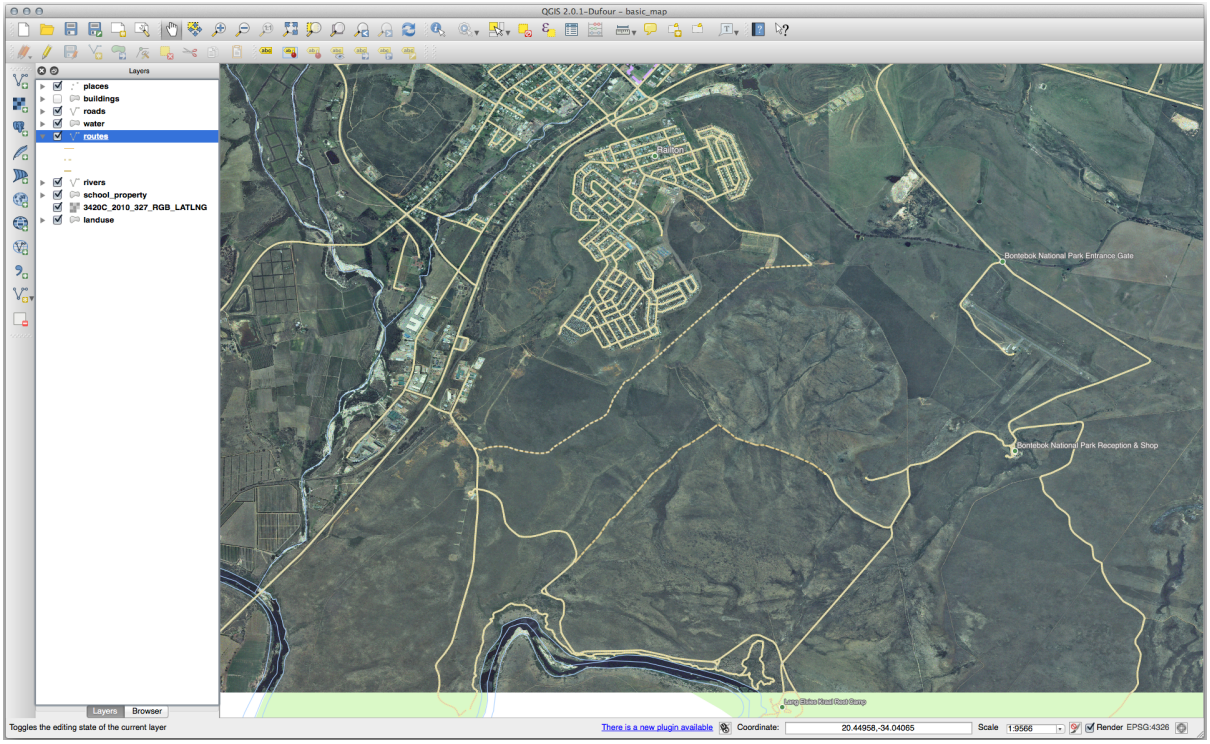
для запуска режима оцифровки линии.
5. Оцифровывайте по одному путь и трек на слое `routes`. Старайтесь как можно точнее следовать маршрутам, добавляя дополнительные точки вдоль углов или поворотов.
6. Задайте значение атрибута `type` для `path` или `track`.
7. Используйте диалоговое окно *Layer Properties* чтобы добавить стиль к маршрутам. Не стесняйтесь ис-

пользовать различные стили для путей и треков.

8. Сохраните изменения и отключите режим редактирования, нажав кнопку  Toggle Editing.

Answer

The symbology doesn't matter, but the results should look more or less like this:



5.1.6 В заключении

Теперь вы знаете, как создавать функции. Этот курс не охватывает добавление точечных функций, потому что это не так уж необходимо, когда вы работаете с более сложными функциями (линии и полигоны). Здесь все работает точно так же, за исключением того, что вы нажимаете только один раз, на то место где вы хотите, чтобы ваша точка располагалась, придайте ей атрибуты, как обычно, и таким образом функция создана.

Знание того как оцифровывать важно потому, что это очень распространенное действие в программах GIS.

5.1.7 Что дальше?

Функции на слое GIS – это не только картинки, но объекты, расположенные в пространстве. Например, смежные полигоны знают где они находятся относительно друг друга. Это называется **топологией**. На следующем занятии вы увидите пример того, почему это может быть полезно.

5.2 Занятие: Топология функций

Топология является полезным аспектом слоев векторных данных, поскольку минимизирует ошибки, такие как перекрытие или пробелы.

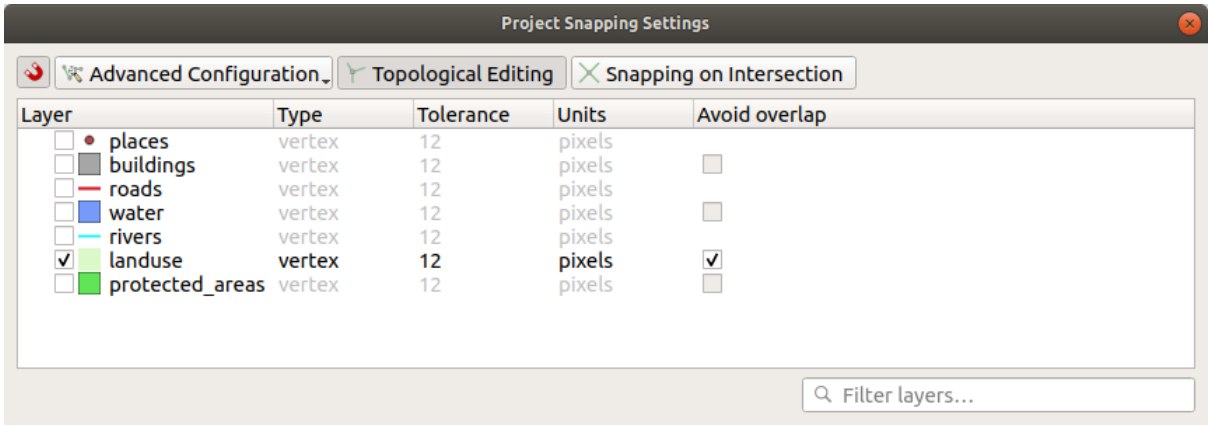
Например, если две функции имеют общую границу, и вы редактируете границу с помощью топологии, то вам не нужно будет редактировать сначала одну функцию, а затем другую и тщательно выстраивать границы так, чтобы они совпадали. Вместо этого можно редактировать их общую границу, и обе функции будут изменяться одновременно.

Цель данного занятия: понять топологию с помощью примеров.

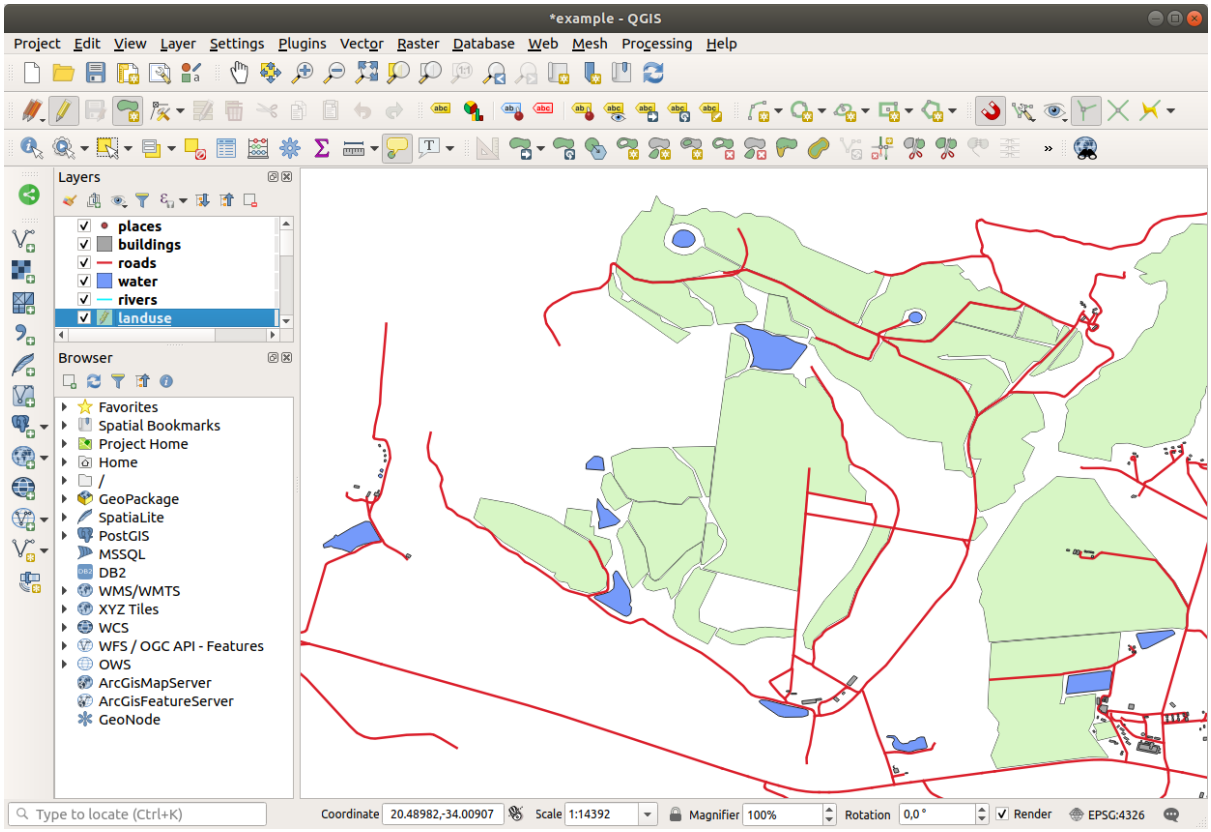
5.2.1  Идем дальше: Привязка

Привязка упрощает топологическое редактирование. Это позволит курсору мыши привязываться к другим объектам во время оцифровки. Чтобы задать опции привязки, выполните следующие действия.

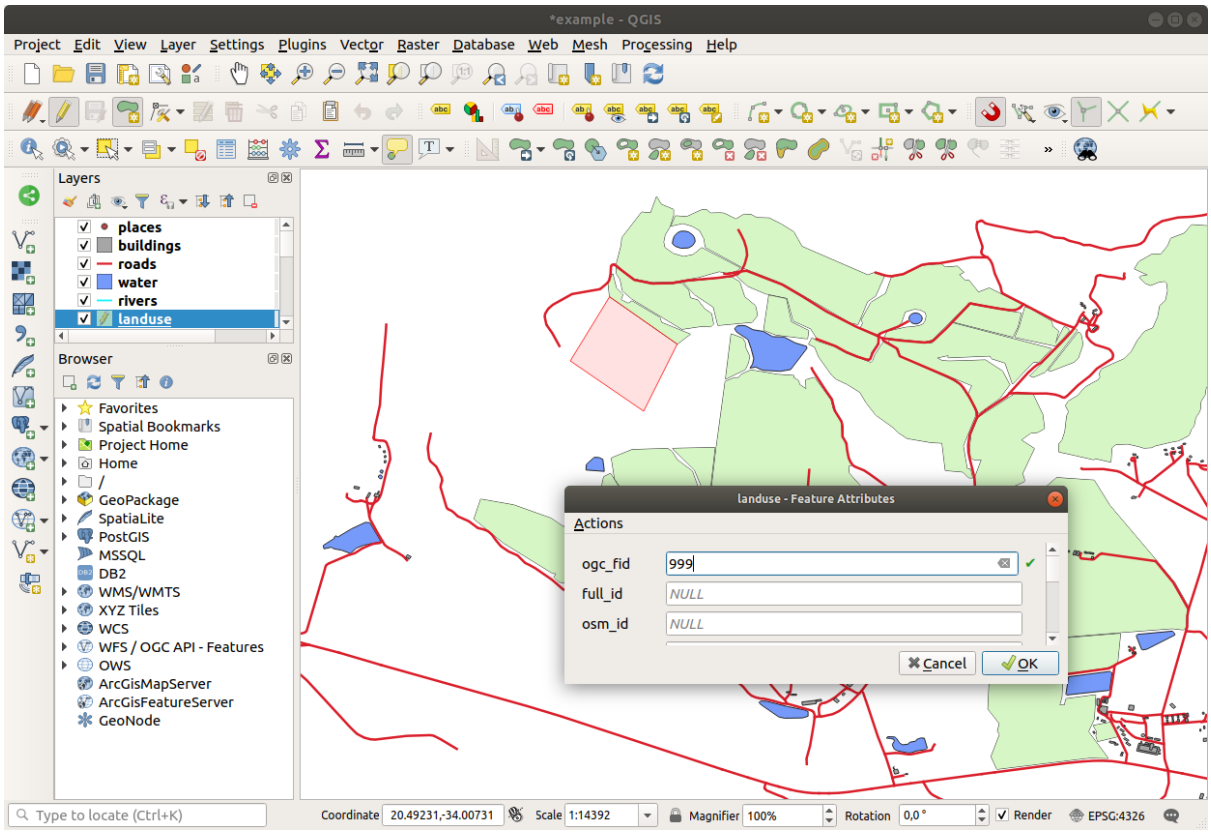
1. Перейдите к меню *Project* ► *Snapping Options...*.
2. Настройте диалоговое окно *Snapping options*, чтобы активировать слой `landuse` с помощью *вершины Type vertex* и допуск в `12` пикселей:



3. Убедитесь что поле в колонке *Avoid overlap* проверено.
4. Выйдите из окна диалога.
5. Выберите слой *landuse* и введите режим ()
6. Проверьте (в *View* ► *Toolbars*) что *Advanced Digitizing* инструмент активирован.
7. Зумирование этой местности (включайте слои и надписи при необходимости):



8. Оцифруйте новую (вымышленную) местность:



9. В ответ на запрос, присвойте ему *OGC_FID* `999`, но можете оставить остальные значения без измене-

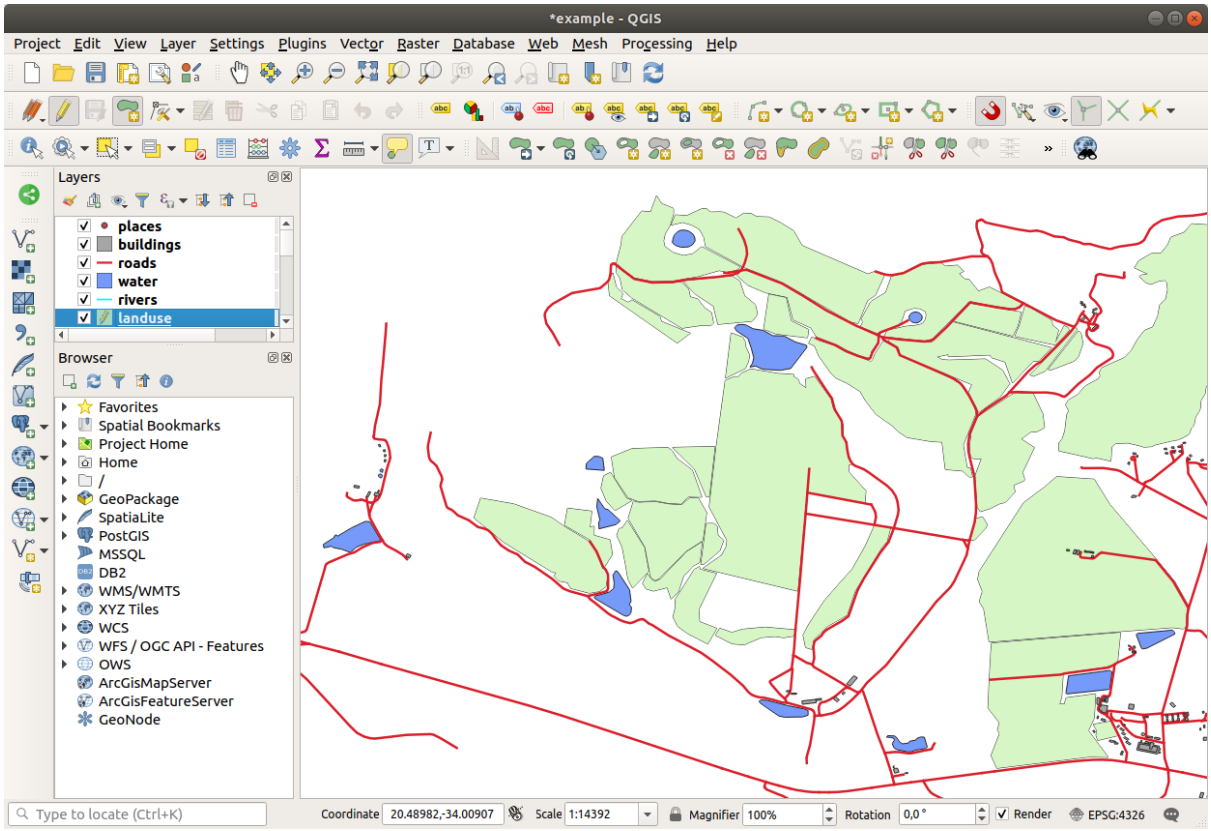
ний.

Если при оцифровке соблюдать осторожность и разрешить привязку курсора к вершинам смежных местностей, то вы заметите, что между новой местностью и существующими смежными местностями не будет никаких промежутков.

10. Обратите внимание на инструменты undo и redo в *Advanced Digitizing* на панели инструментов.

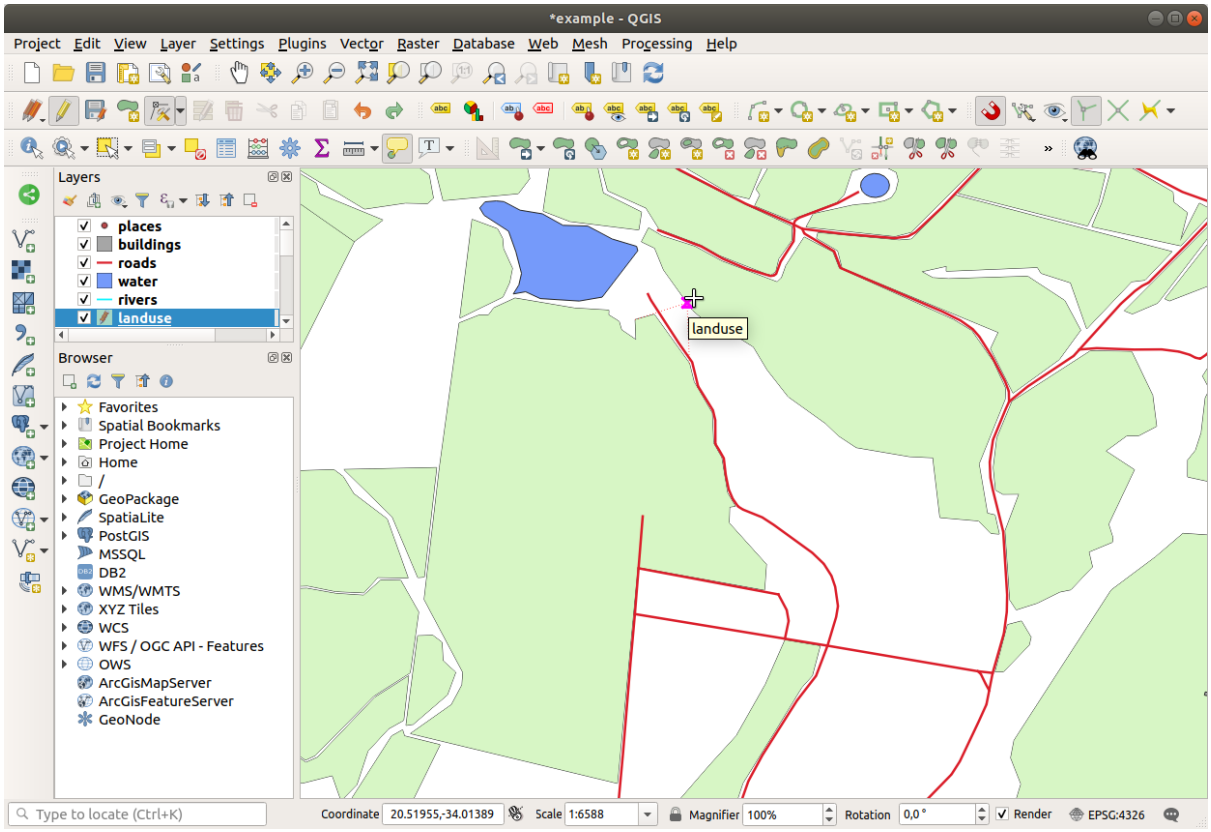
5.2.2 Идем дальше: Правильные топологические Функции

Иногда может потребоваться обновление функций топологии. На нашем участке работы, эта местность была превращена в лес, поэтому слой землепользования `landuse` нуждается в обновлении. Поэтому мы будем расширять и присоединять некоторые лесные функции на этой местности:



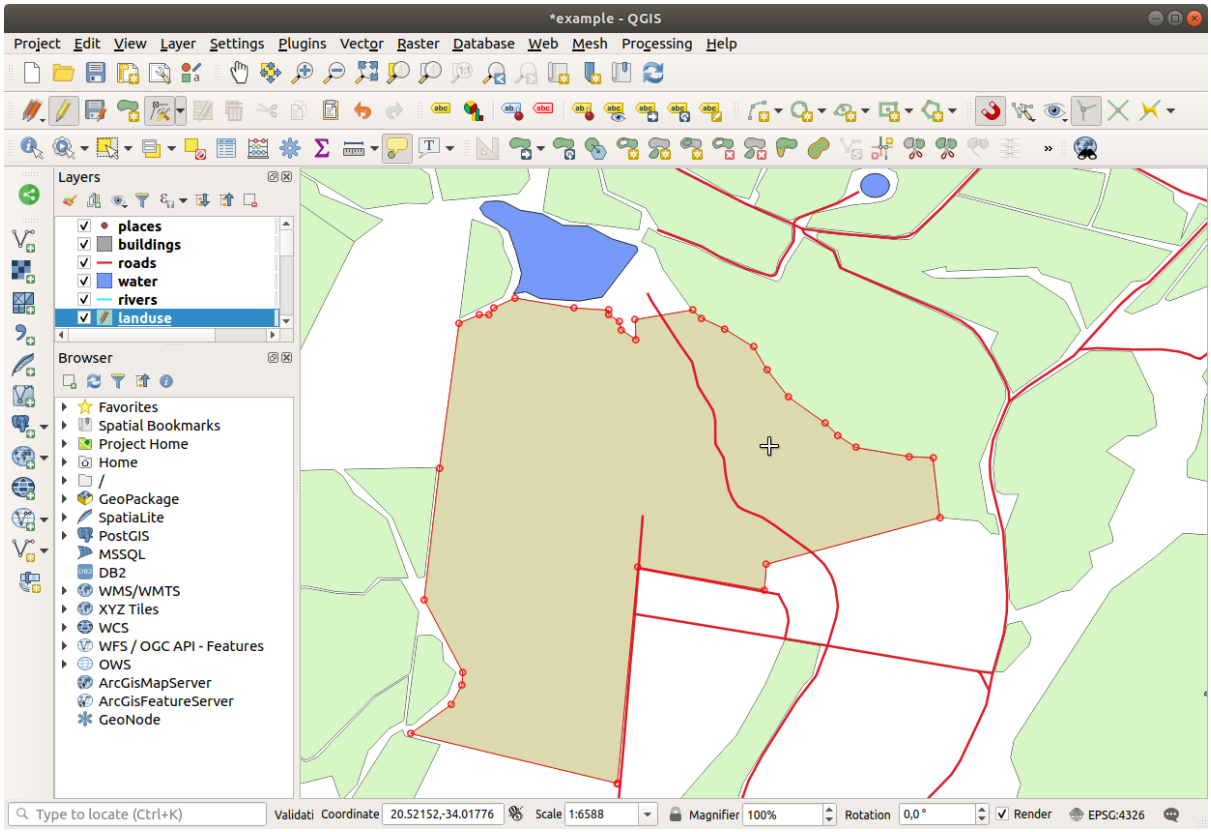
Вместо создания новых полигонов для присоединения к лесным участкам мы будем использовать *Инструмент Vertex* для редактирования и присоединения существующих полигонов.

1. Переход в режим редактирования (если он еще не активен).
2. Выберите инструмент Vertex Tool.
3. Выберите участок леса, выберите вершину и переместите ее на смежную вершину, чтобы две функции леса совместились:



4. Кликните на другие вершины и привяжите их к месту.

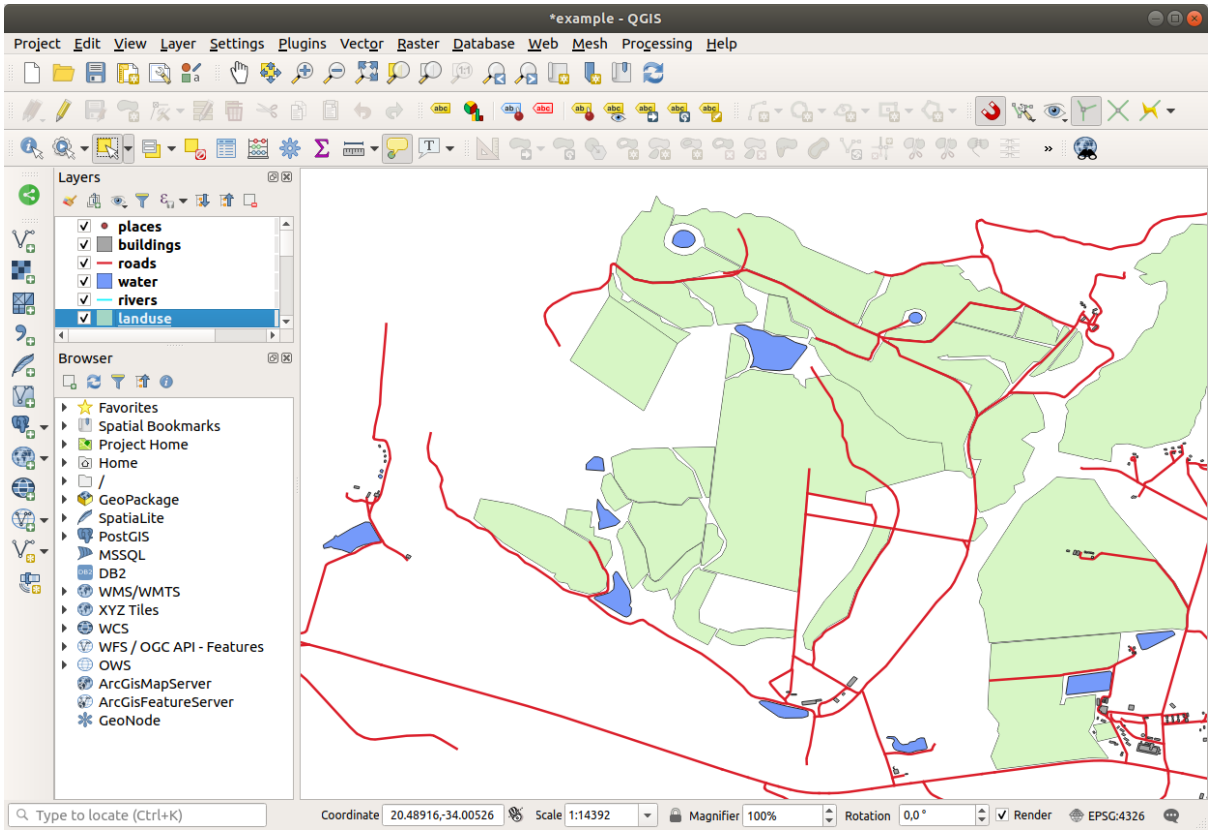
   Топологически правильная граница выглядит следующим образом:



Продолжайте и присоедините еще несколько местностей с помощью *Инструмента Vertex*.

Можете использовать инструмент Add Polygon Feature для того, чтобы заполнить пробел между двумя полигонами «лес». Если включен параметр *Избежать перекрытия*, не нужно добавлять каждую вершину - они будут добавляться автоматически, если ваш новый полигон перекрывает существующие.

Если вы используете наш пример данных, то ваш лесной участок должен выглядеть примерно так:

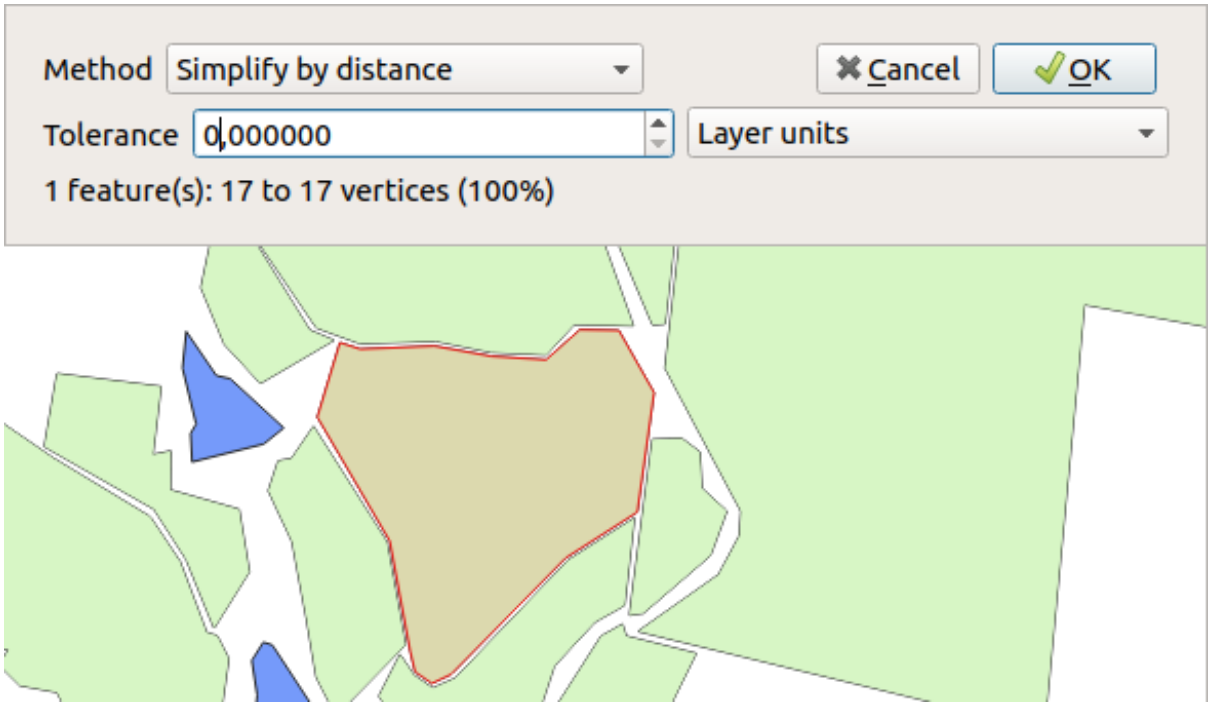


Не волнуйтесь, если вы присоединились к большему, меньшему или разным участкам леса.

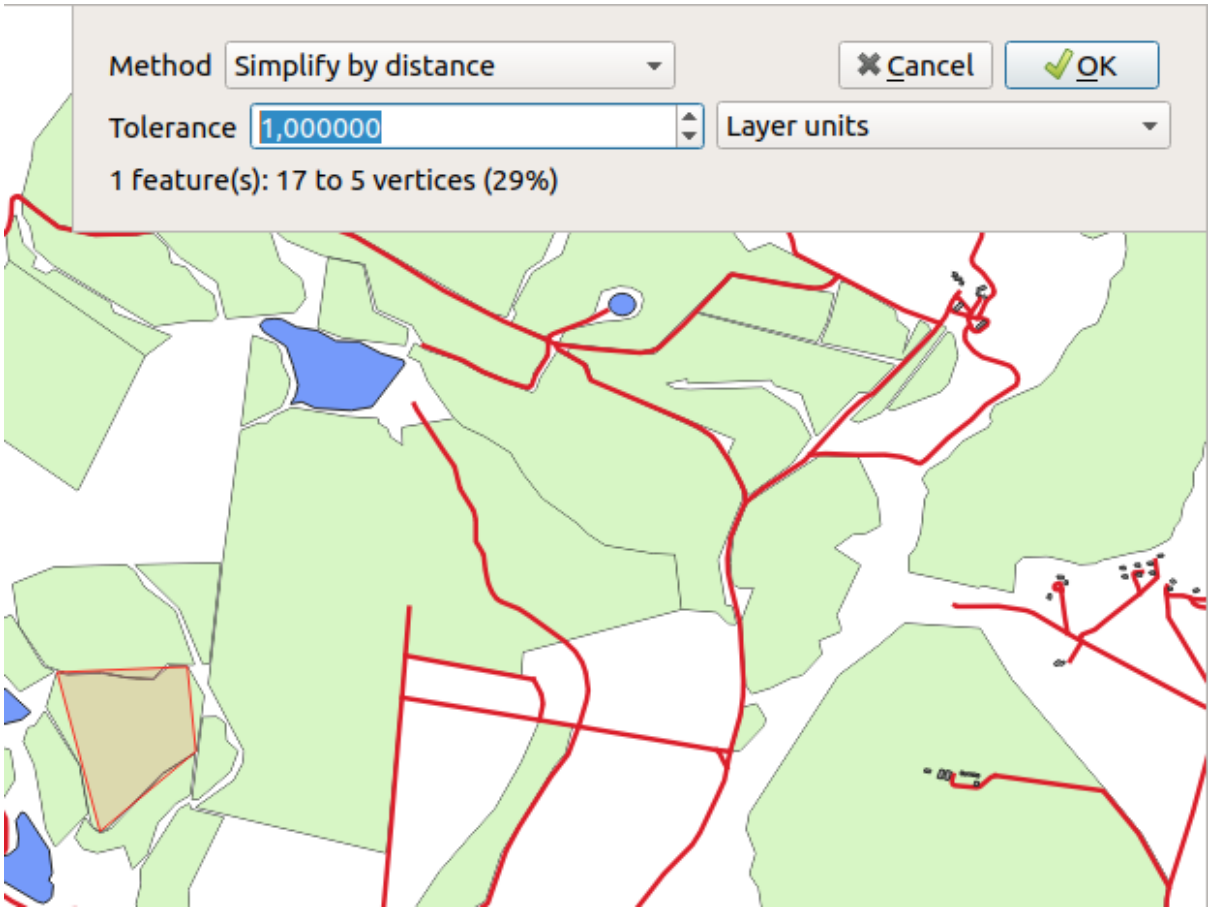
5.2.3 Идем дальше: Инструмент: Упрощение функции

Продолжаем на том же слое, протестируем инструмент Упрощение функции Simplify Feature.

1. Кликните на него и активируйте.
2. Кликните на один из участков, который вы уже присоединили или с помощью *Инструмента Vertex* или инструмента *Add Feature*, вы увидите это диалоговое окно:



3. Измените *Tolerance* и вы увидите что произойдет:



Это позволяет уменьшить число вершин.

4. Кликните *OK*

The advantage of this tool is that it provides you with a simple and intuitive interface for generalization. But notice that the tool ruins topology. The simplified polygon no longer shares boundaries with its adjacent polygons, as it should. So this tool is better suited for stand-alone features.

Прежде чем продолжить, верните полигон в исходное состояние, отменив последнее изменение.

5.2.4  Попробуйте сами Инструмент: Добавить кольцо

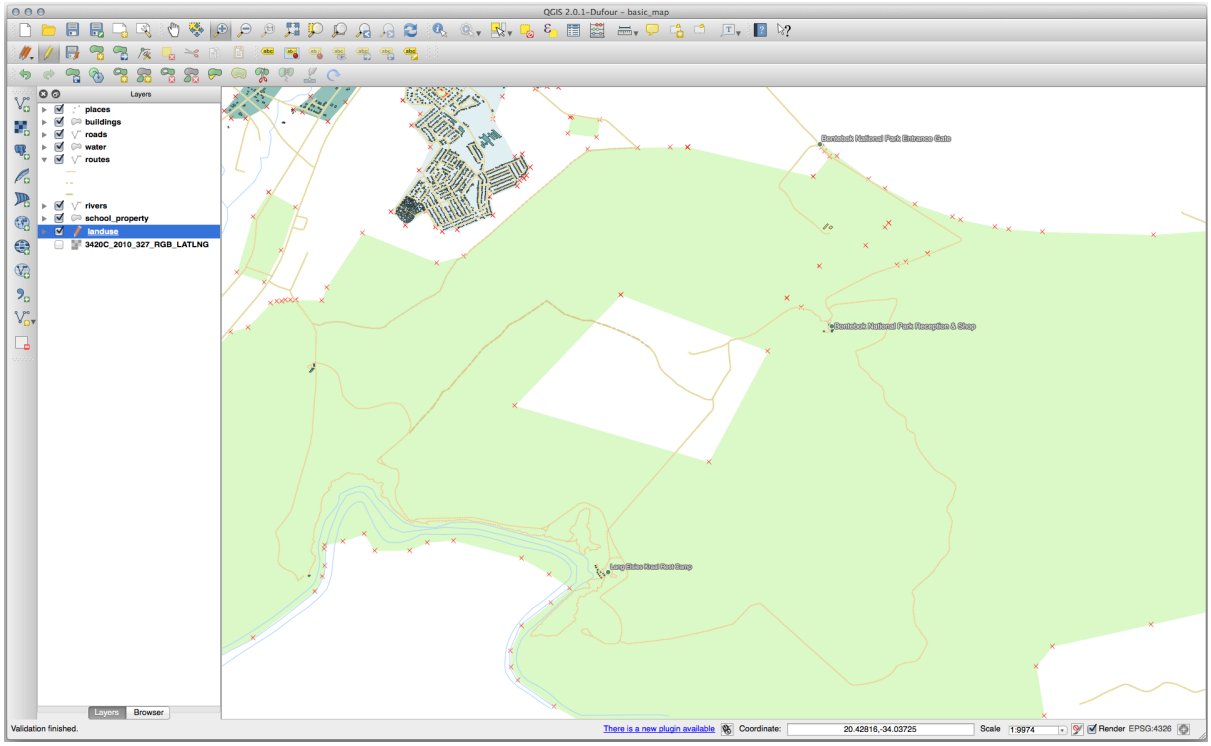
Инструмент  Add Ring позволяет добавить внутреннее кольцо к функции полигона (вырезать отверстие в полигоне), если отверстие полностью находится внутри полигона (если есть касание границы, это OK). Например, если вы оцифровали внешние границы Южной Африки и вам нужно добавить отверстие для Лесото, вы бы использовали это инструмент.

При эксперименте с инструментом вы можете заметить, что опции привязки могут помешать созданию кольца внутри полигона. Поэтому рекомендуется отключить привязку перед вырезанием отверстия.

1. Отключить привязку для слоя `landuse` с помощью кнопки  Enable Snapping (или используйте ярлыки `s`).
2. Используйте инструмент  Add Ring для создания отверстия с середине геометрии полигона.
3. Нарисуйте полигон над целевой функцией, как если бы вы использовали инструмент Добавить полигон  Add polygon.
4. Когда вы нажмете правый клик, отверстие станет видимым.
5. Remove the hole you just created using the  Delete Ring tool. Click inside the hole to delete it.

Answer

The exact shape doesn't matter, but you should be getting a hole in the middle of your feature, like this one:



- Undo your edit before continuing with the exercise for the next tool.

5.2.5 Попробуйте сами Инструмент: Добавить часть

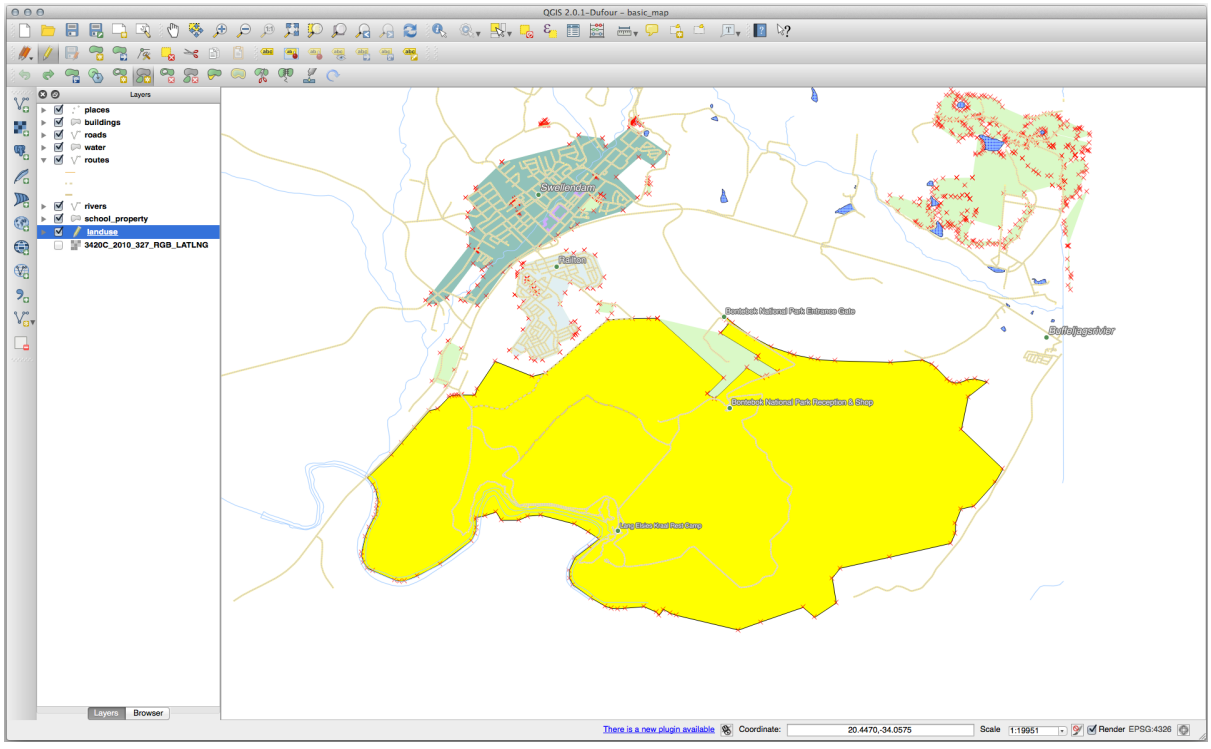
Инструмент  Add Part позволяет добавить новую частьдеталь к функции, которая не связана напрямую с основной функцией. Например, если вы оцифровали границы материковой части Южной Африки, но еще не добавили острова Принца Эдуарда, вы бы использовали этот инструмент для их создания.

1. Выберите полигон, к которому требуется добавить деталь, с помощью инструмента Выберите функцию по местности или одним кликом  Select Features by area or single click.
2. Используйте инструмент *Add Part* для добавления внешней области.
3. Удалите только что созданную часть с помощью инструмента  Delete Part.

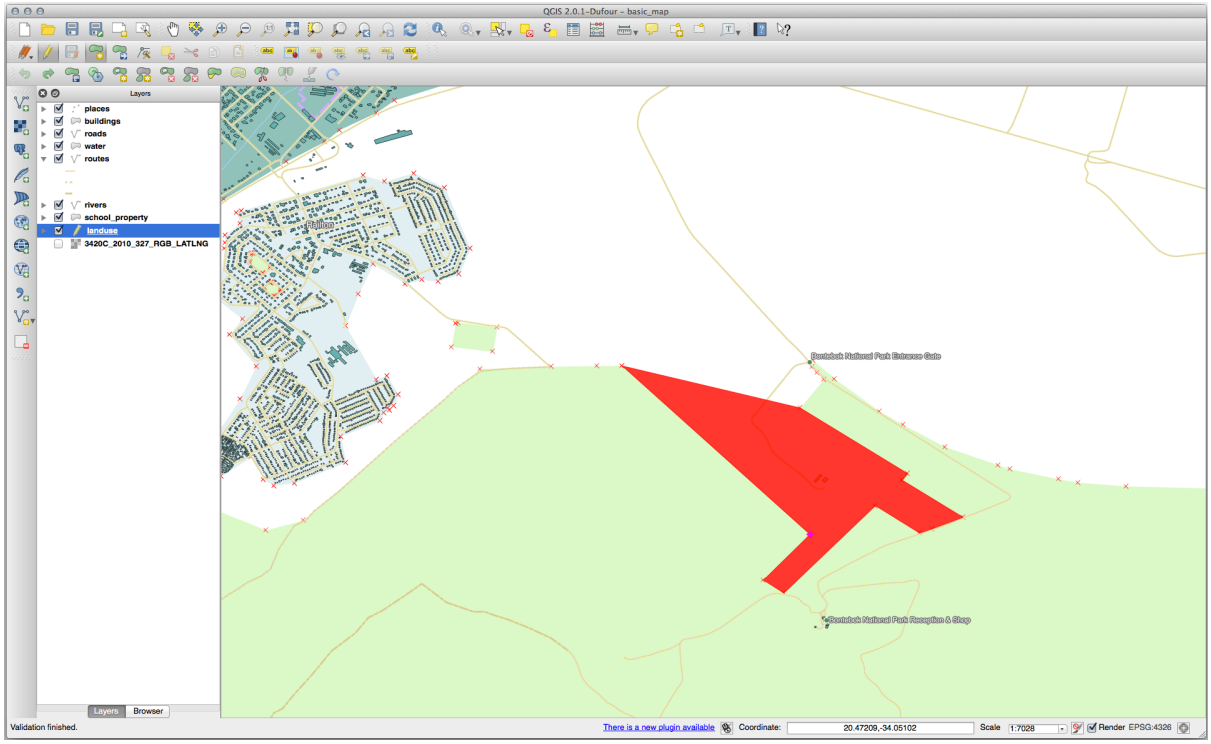
Примечание: Кликните внутри детали, чтобы удалить ее.

Answer

1. First select the Bontebok National Park:



2. Now add your new part:



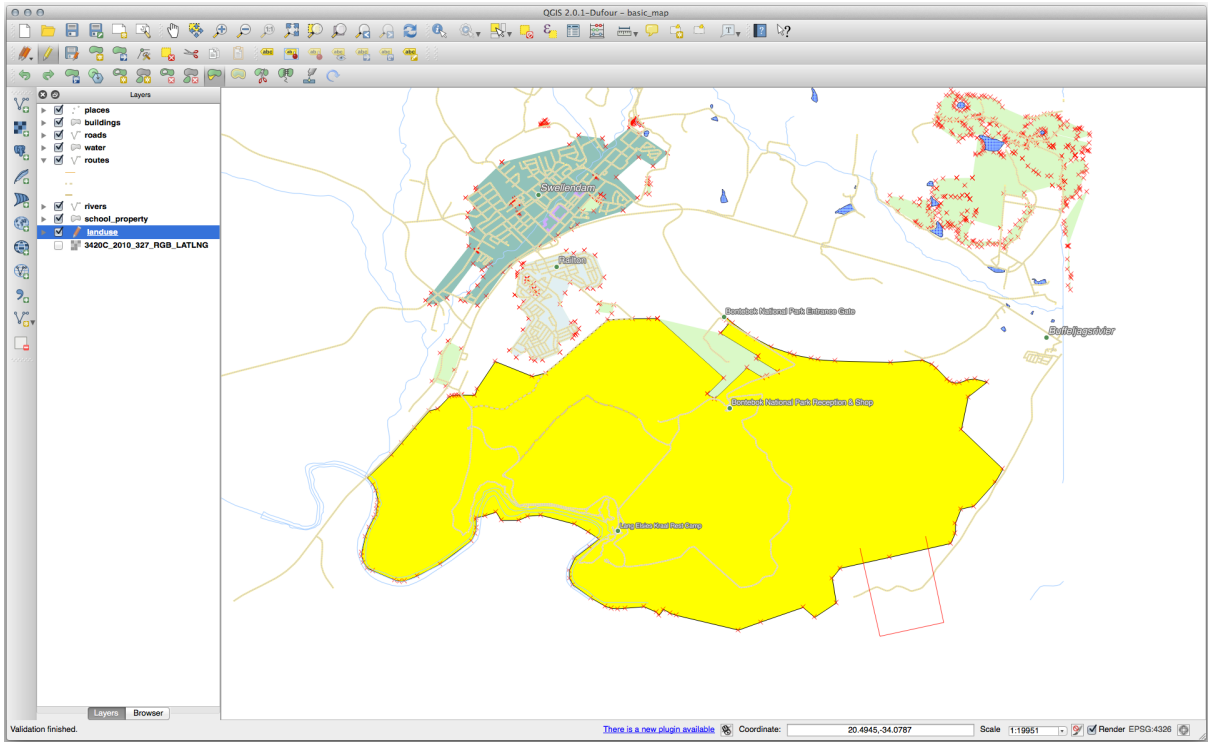
3. Undo your edit before continuing with the exercise for the next tool.

5.2.6  Идем дальше: Инструмент: Изменить форму Функции

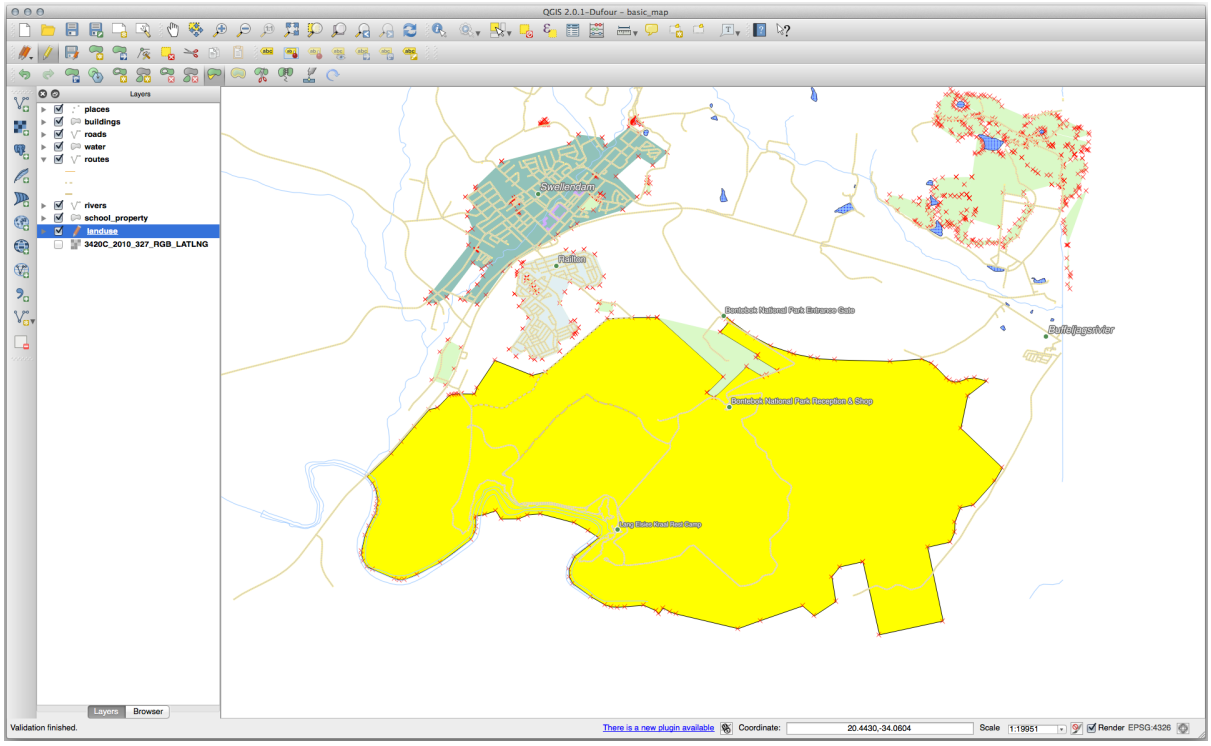
Инструмент  Reshape Features используется для расширения функции полигона или вырезания его части (вдоль границы).

Расширение:

1. Выберите полигон с помощью инструмента  Select Features by area or single click.
2. Кликните левой кнопкой мыши внутри полигона, чтобы начать рисование.
3. Нарисуйте форму вне полигона. Последняя вершина должна быть снова внутри полигона.
4. Кликните правой кнопкой мыши для завершения фигуры:



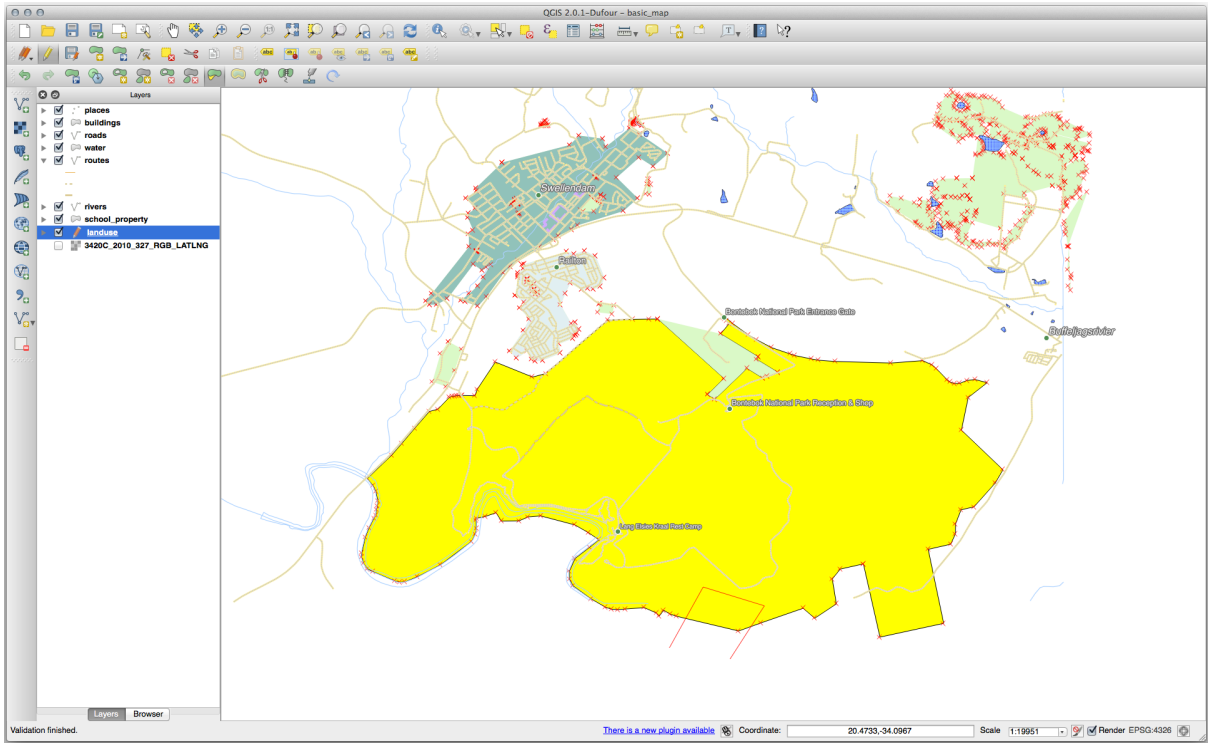
Это даст результат, аналогичный нижеследующему:



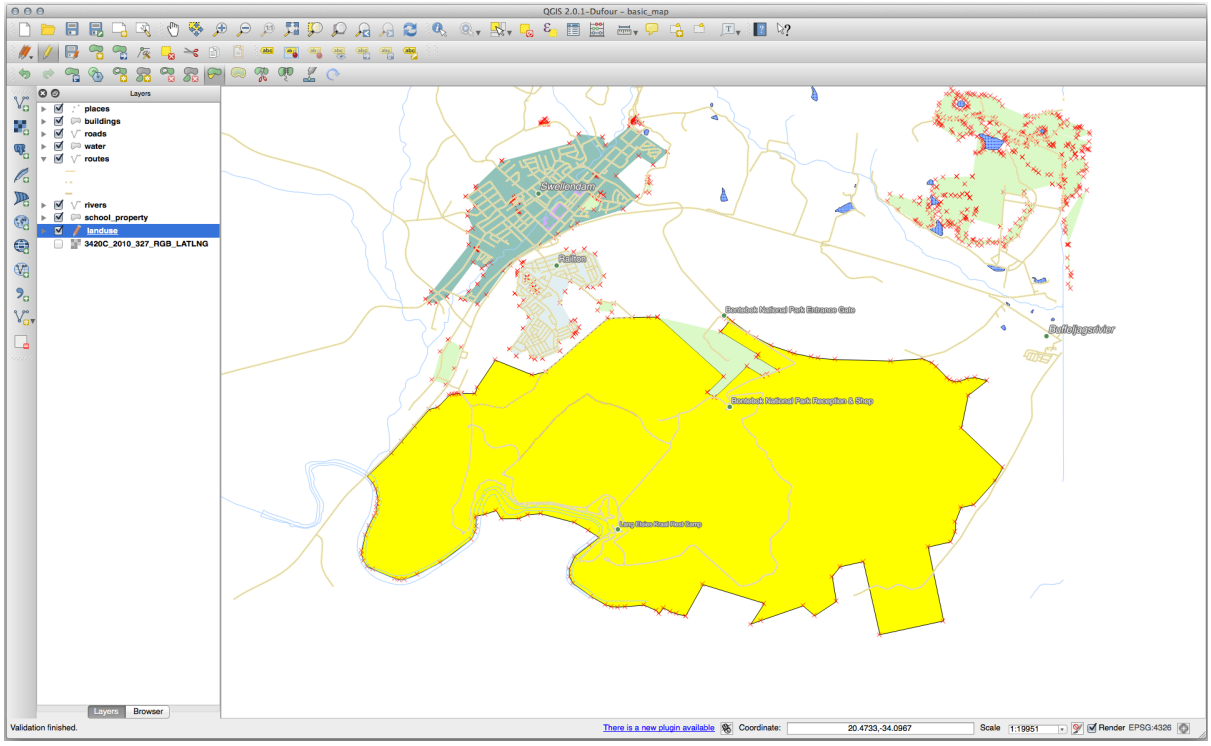
Вырежьте часть:

1. Выберите полигон с помощью инструмента  Select Features by area or single click.
2. Кликните за пределами полигона.
3. Нарисуйте фигуру внутри полигона. Последняя вершина должна быть возвращена за пределы полигона.

4. Кликните правой кнопкой мыши вне полигона:



Результат того, что было выше:

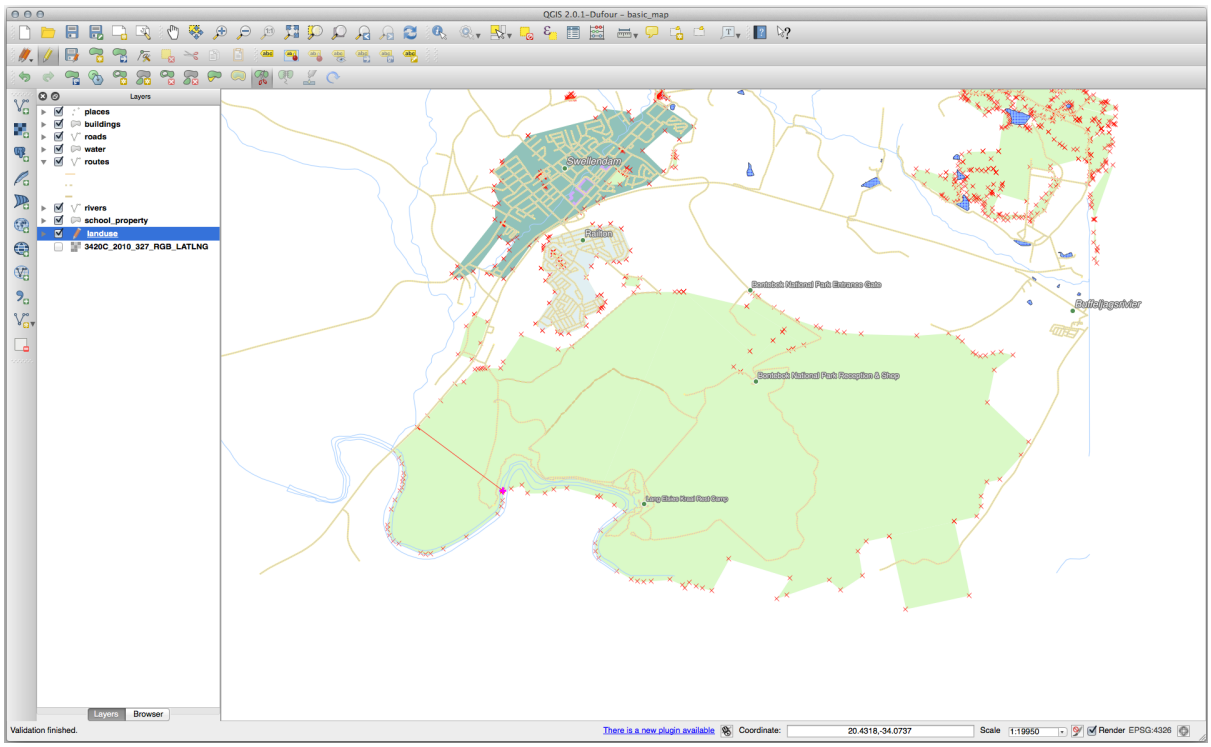


5.2.7 Попробуйте сами Инструмент: Разделить Функции

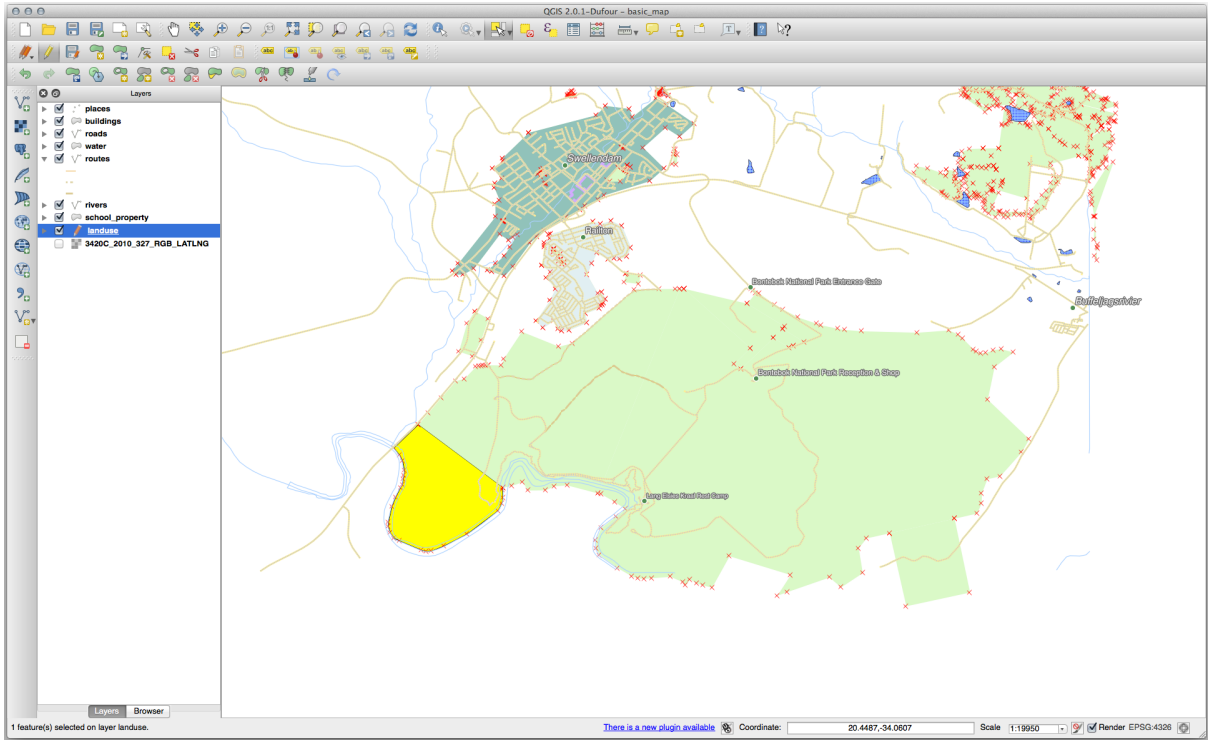
Инструмент Split Features аналогичен инструменту Reshape Features кроме одного, того что он не удаляет ни одной из двух частей. И наоборот, сохраняет обе.

Мы будем использовать инструмент, чтобы отделить угол от полигона.

1. Сначала выберите слой `landuse` и повторно включите для него привязку.
2. Выберите инструмент Split Features и кликните на вершину для того, чтобы начать рисовать линию.
3. Нарисуйте ограничивающую линию.
4. Кликните вершину на «противоположной» стороне полигона, который вы хотите разделить, и кликните правой кнопкой мыши для завершения линии:



5. На данный момент может показаться, что ничего не произошло. Но помните, что слой `landuse` визуализируется без линий границы, поэтому новая линия разделения не будет показана.
6. Используйте инструмент Select Features by area or single click чтобы выделить только что разделенную часть-деталь; теперь новый элемент будет выделен:



5.2.8 Попробуйте сами: Инструмент: Слить Функции

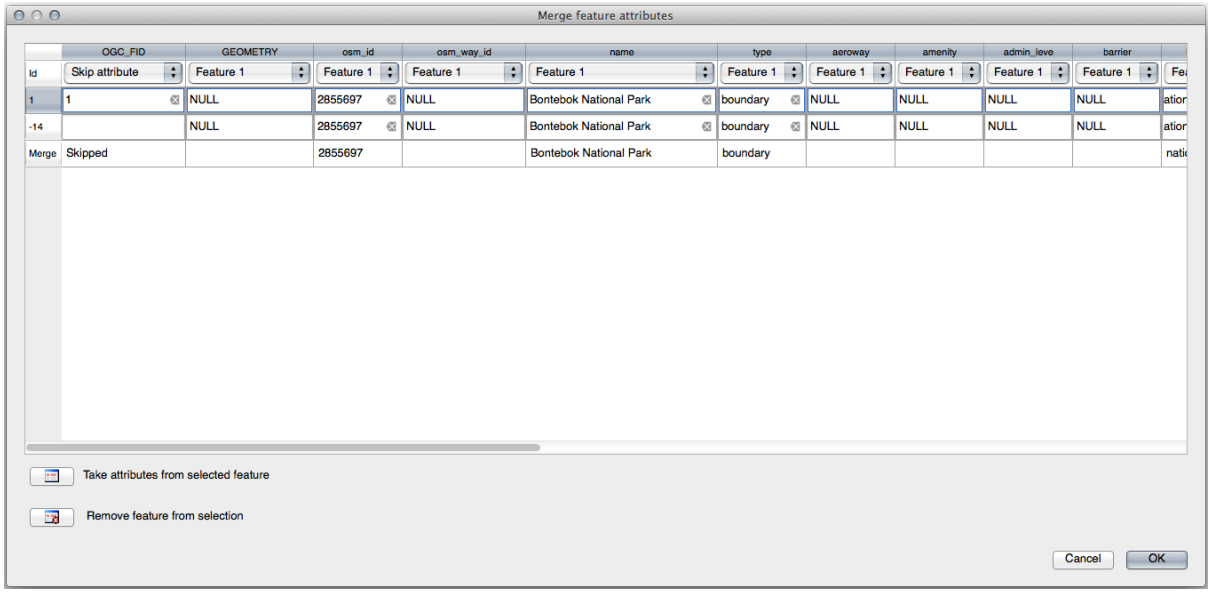
Теперь мы снова присоединим элемент, который вы только что разделили, к оставшейся части полигона:

1. Поэкспериментируйте с инструментами Merge Selected Features и Merge Attributes of Selected Features.
2. Отметьте разницу.

Answer

- Use the *Merge Selected Features* tool, making sure to first select both of the polygons you wish to merge.
- Use the feature with the *OGC_FID* of `1` as the source of your attributes (click on its entry in the dialog, then click the *Take attributes from selected feature* button):

If you're using a different dataset, it is highly likely that your original polygon's *OGC_FID* will not be `1`. Just choose the feature which has an *OGC_FID*.



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

5.2.9 В заключении

Редактирование топологии - это мощный инструмент, который позволяет быстро и легко создавать и изменять объекты, обеспечивая при этом их топологическую корректность.

5.2.10 Что дальше?

Теперь вы знаете, как легко оцифровать форму объектов, но добавить атрибут - это пока еще заморочка! Далее мы покажем, как использовать формы, делая редактирование атрибутов более простым и эффективным.

5.3 Занятие: Формы

При добавлении новых данных с помощью оцифровки появляется диалоговое окно, позволяющее заполнить атрибуты для этой функции. Однако этот диалоговое окно по умолчанию выглядит не очень приятно. Это может вызвать проблему для удобства использования, особенно если вам надо создать большие наборы данных или если вы хотите, чтобы другие люди помогали вам оцифровывать, а они считают, что формы по умолчанию запутаны.

К счастью, QGIS позволяет создавать собственные диалоговые окна для слоя. На этом занятии показывается как это сделать.

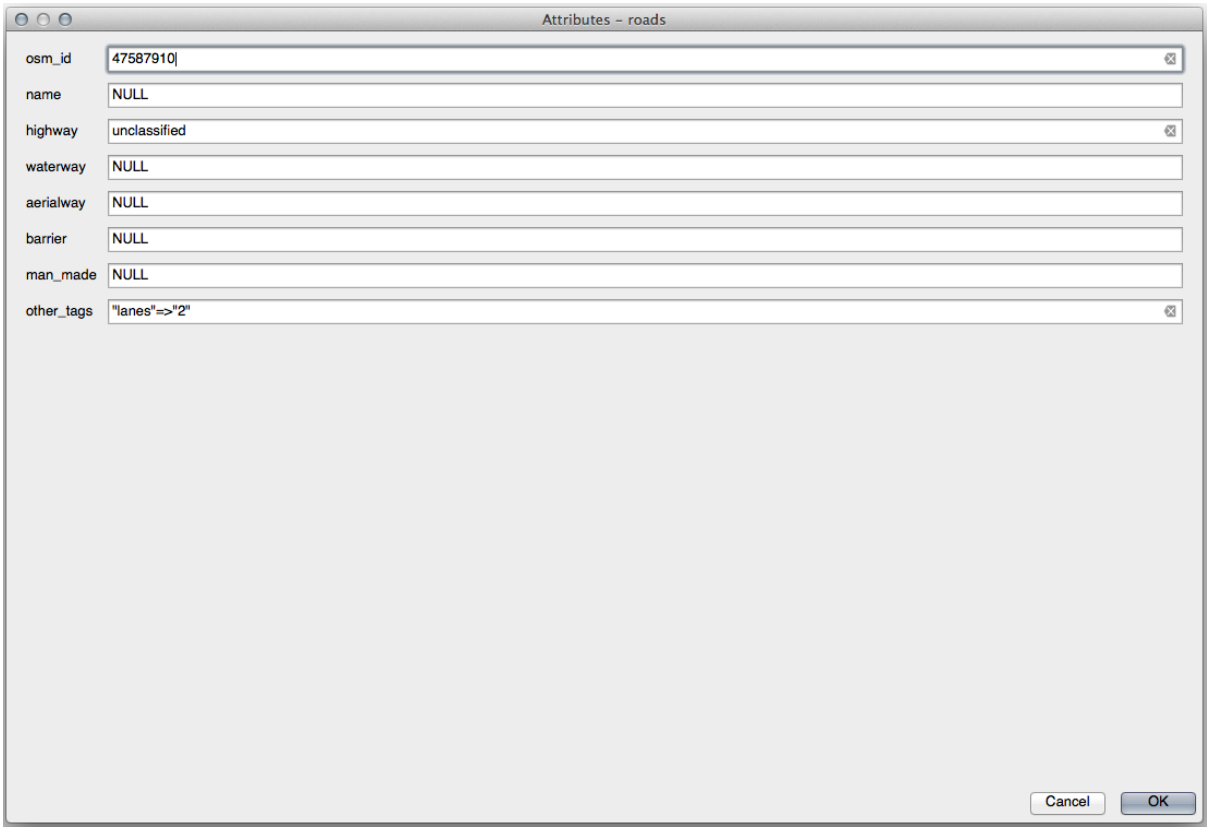
Цель этого занятия: создать форму для слоя.

5.3.1 Идем дальше: Используем Функциональность дизайна Формы QGIS

1. Выберите слой `roads` на панели *Layers*.
2. Переходите в *режим редактирования*, как и ранее.
3. Откройте таблицы атрибутов слоя `roads` дорог.
4. Кликните правой кнопкой мыши любую ячейку в таблице. Появится короткое меню, включающее *Open form*.
5. Кликните его, чтобы увидеть форму, которую QGIS создает для этого уровня.

Очевидно, что было бы неплохо делать это, глядя на карту, вместо того, чтобы постоянно искать конкретную улицу в таблице атрибутов *Attribute Table*.

1. Выберите слой *roads* на панели *Layers*.
2. С помощью инструмента Идентификация функций Identify Features кликните любую улицу на карте.
3. Панель *Identify Results* откроется, на которой отображается «дерево» значений полей и другая общая информация о выбранной функции.
4. At the top of the panel, check the *Auto open form for single feature results* checkbox in the Identify Settings menu.
5. Теперь кликните еще раз на любой улице на карте. В предыдущем диалоговом окне *Identify Results* отображается знакомая форма:

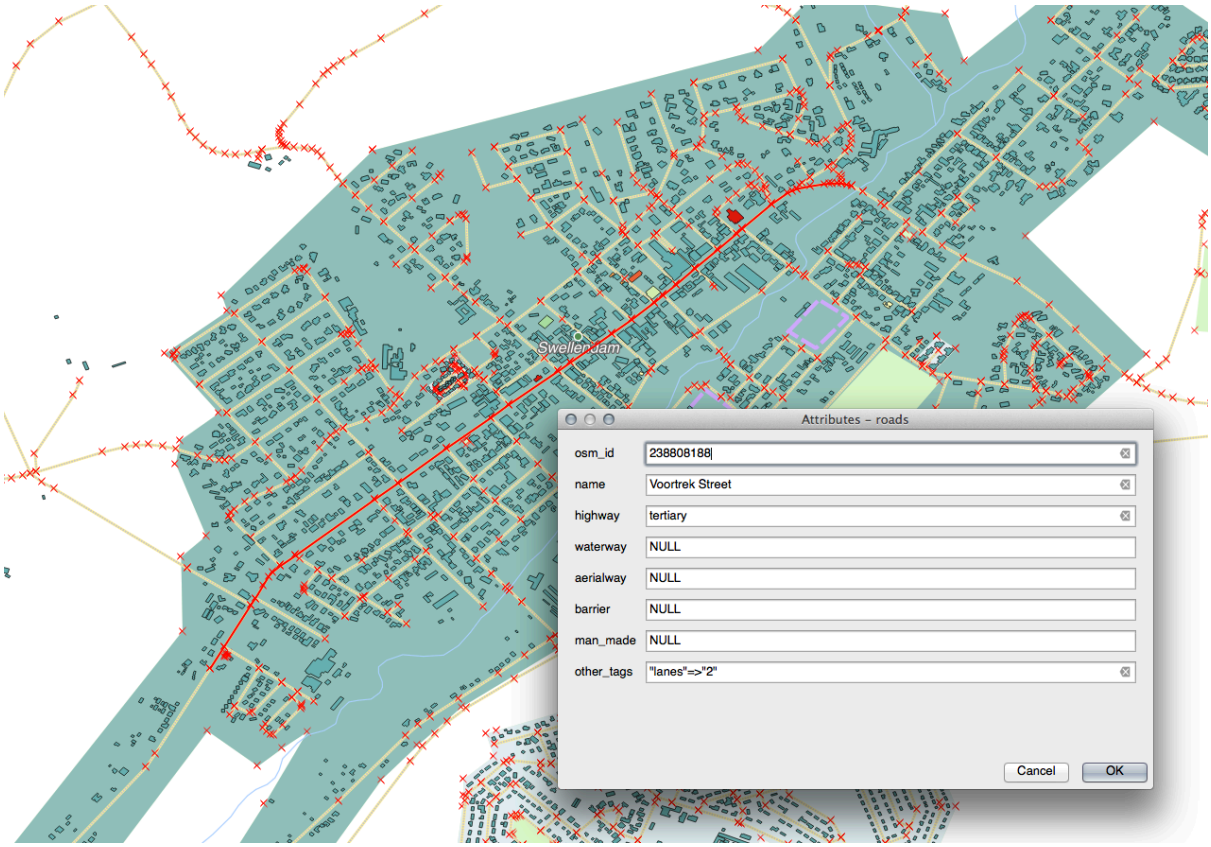


6. Каждый раз когда вы кликаете по одной функции с помощью инструмента *Идентифицировать* его форма появляется при условии, что установлен флажок Автоматически открывать форму *Auto open form*.

5.3.2 Попробуйте сами, используйте формы для редактирования значений

Если вы находитесь в режиме редактирования, эту форму можно использовать для редактирования атрибутов функции.

1. Активировать режим редактирования (если он еще не активирован).
2. С помощью инструмента Определить функции Identify Features кликните главную улицу, проходящую через Swellendam Swellendam:

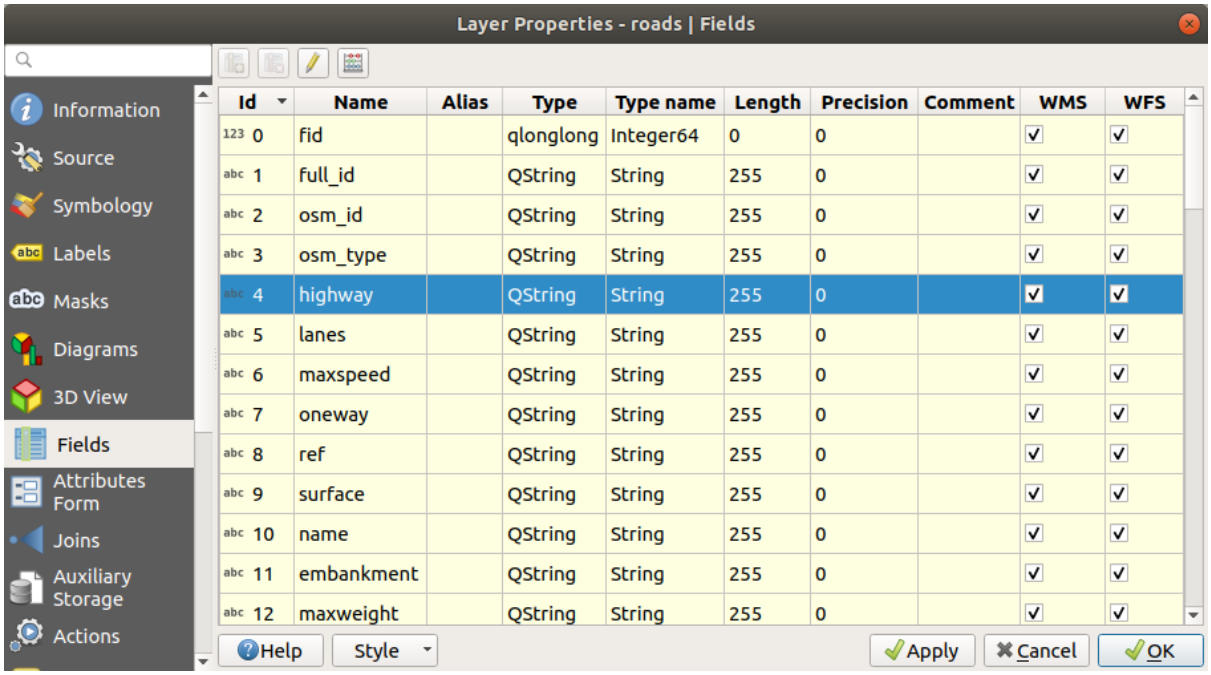


3. Редактируйте его значение *highway* как `secondary`.
4. Выходите из режима редактирования и сохраняйте изменения.
5. Откройте таблицу атрибутов *Attribute Table* и отметьте что значение было обновлено в таблице атрибутов и, следовательно, в исходных данных.

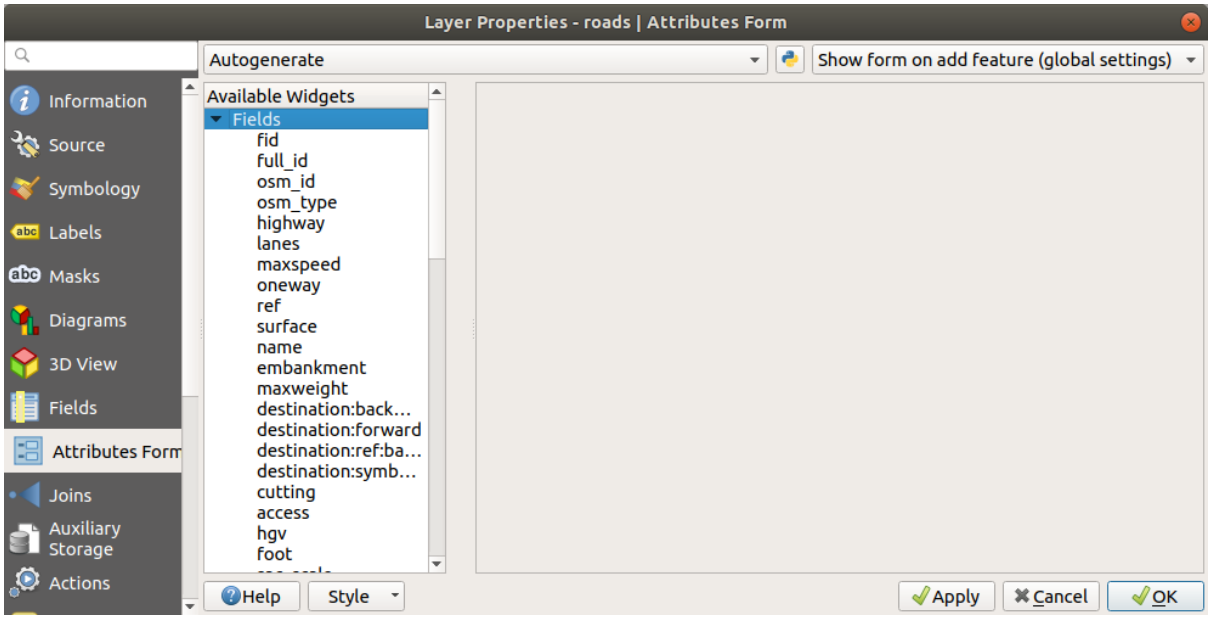
5.3.3 Идем дальше: Настройка типов полей формы

Приятно редактировать вещи с помощью формы, но все равно надо вводить все вручную. К счастью, формы имеют различные виды так называемых *виджетов*, которые позволяют редактировать данные различными способами.

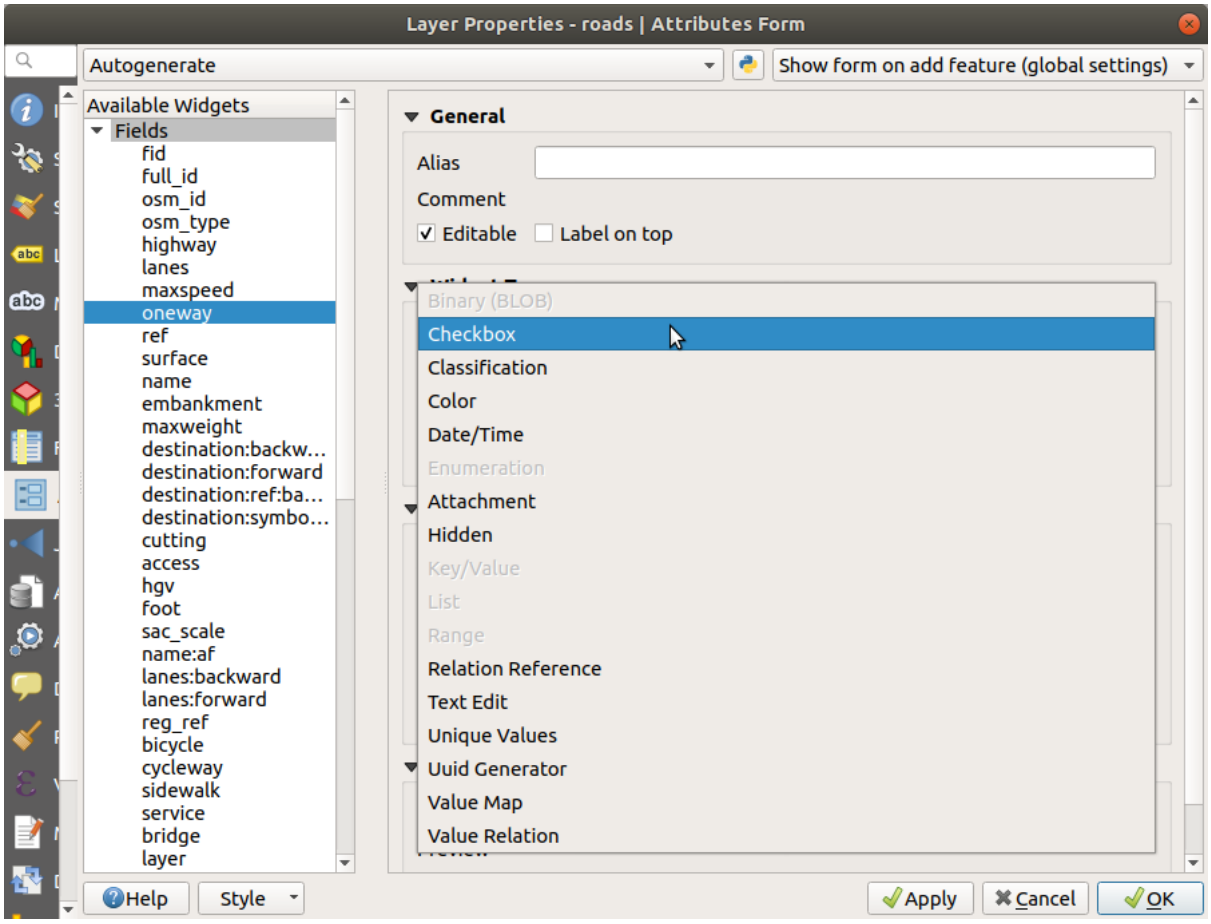
1. Откройте *roads* слоя *Properties...*
2. Переедите на вкладку *Fields*. Вы увидите это:



3. Перейдите на вкладку *Attributes Form*. Вы увидите это:



4. Кликните на ряд `oneway` и выберите *Checkbox* как *Widget Type* в списке вариантов:



5. Кликните *OK*
6. Переход в режим редактирования (если слой `roads` не находится уже в режим редактирования)
7. Кликните на инструмент  Identify Features.
8. Кликните на ту же главную дорогу, которую вы выбрали ранее.

Теперь вы увидите, что *oneway* однонаправленный атрибут имеет флажок рядом с ним, обозначающий `True` (флажок) или `False` (нет флажка).

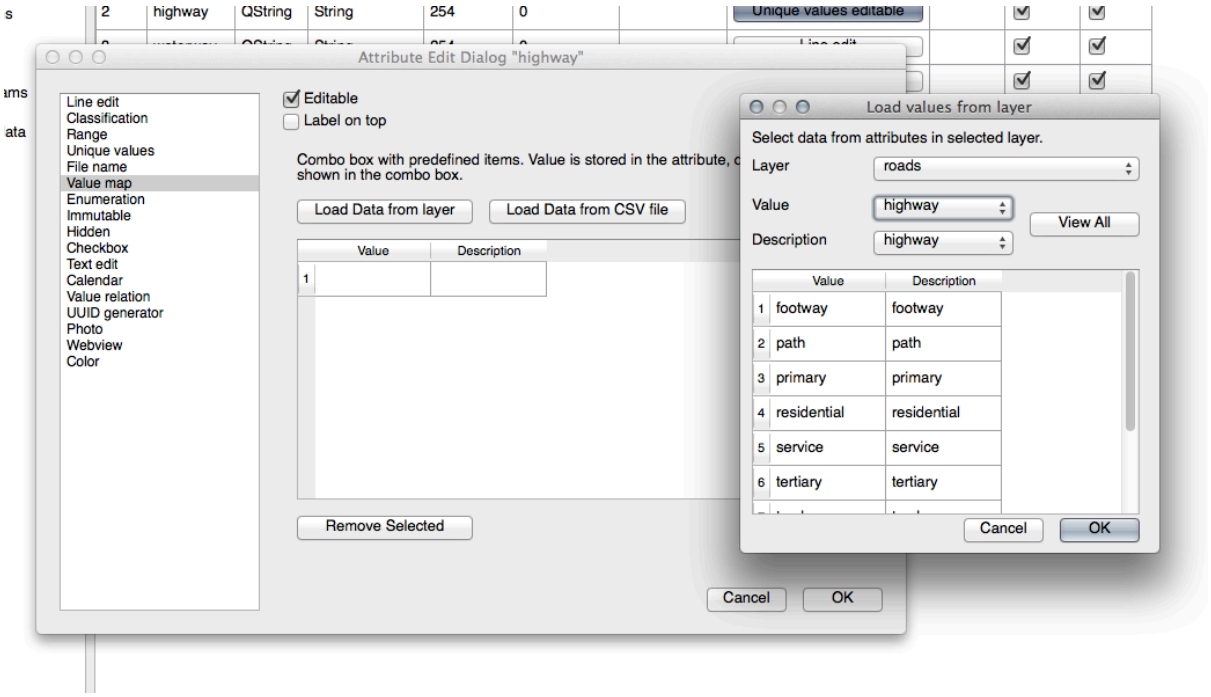
5.3.4  Попробуйте сами

Задайте более подходящий виджет формы для поля *highway*.

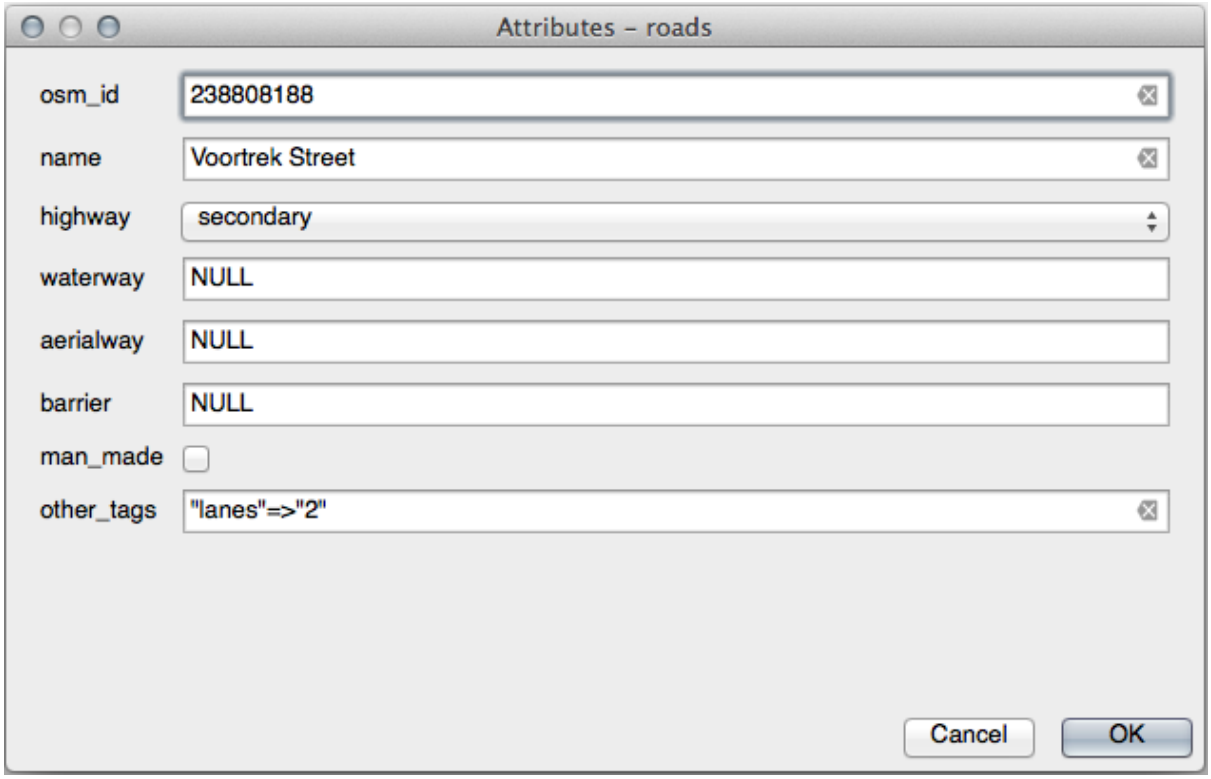
Answer

For the *TYPE*, there is obviously a limited amount of types that a road can be, and if you check the attribute table for this layer, you'll see that they are predefined.

1. Set the widget to *Value Map* and click *Load Data from Layer*.
2. Select *roads* in the *Label* dropdown and *highway* for both the *Value* and *Description* options:



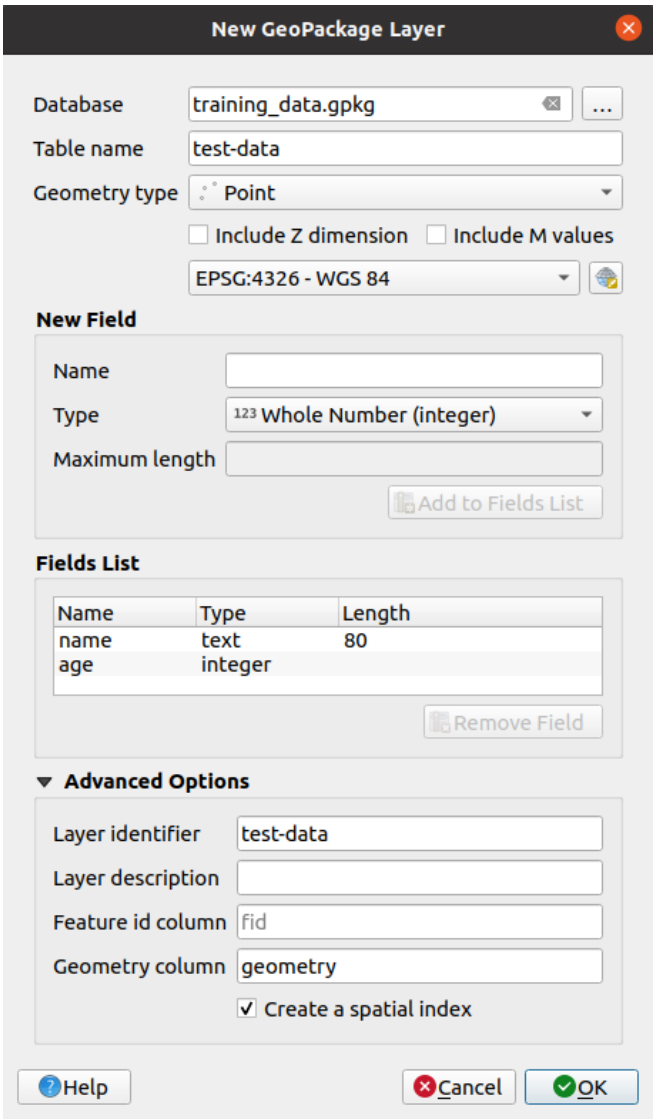
3. Click *OK* three times.
4. If you use the *Identify* tool on a street now while edit mode is active, the dialog you get should look like this:



5.3.5 Попробуйте сами, создайте тестовые данные

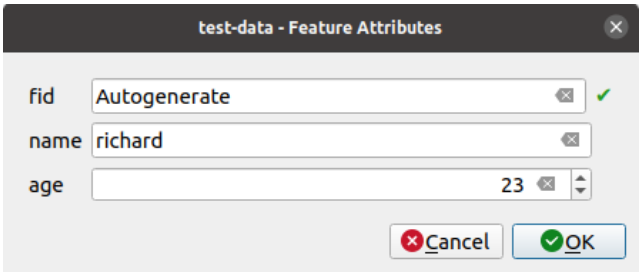
Также можно полностью создать собственную пользовательскую форму.

1. Создайте простой слой точек с именем `test-data` с двумя атрибутами:
   - название (текст)
   - возраст (целое число)



2. Зафиксируйте несколько точек на новом слое с помощью инструментов оцифровки, чтобы у вас было немного данных для воспроизведения. При каждой записисъемке новой точки вам должна быть представлена форма записи атрибутов, сгенерированных QGIS по умолчанию.

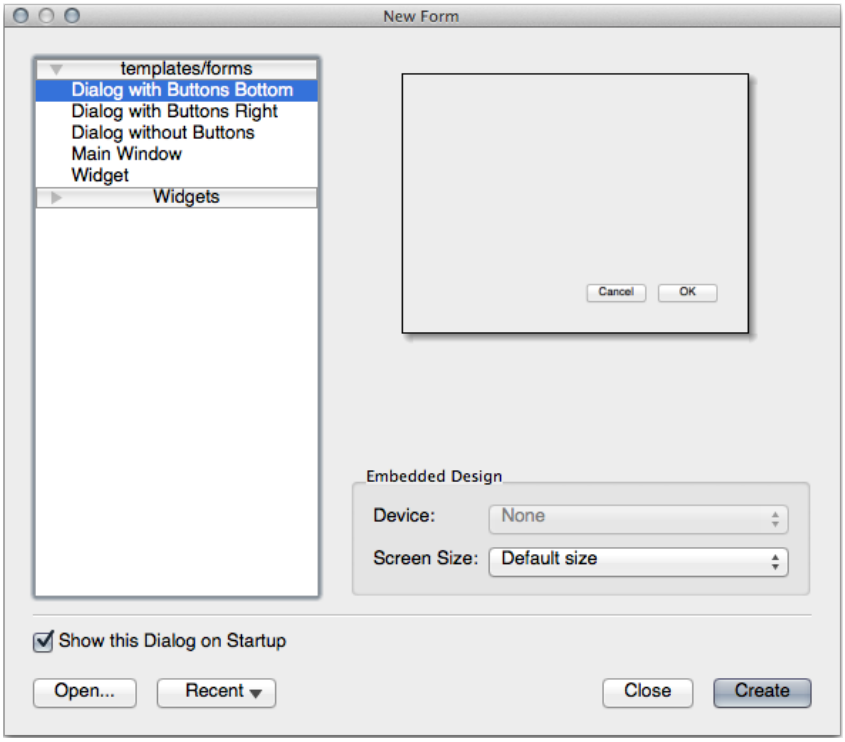
Примечание: Может потребоваться отключить привязку, если она все еще включена для предыдущих задач.



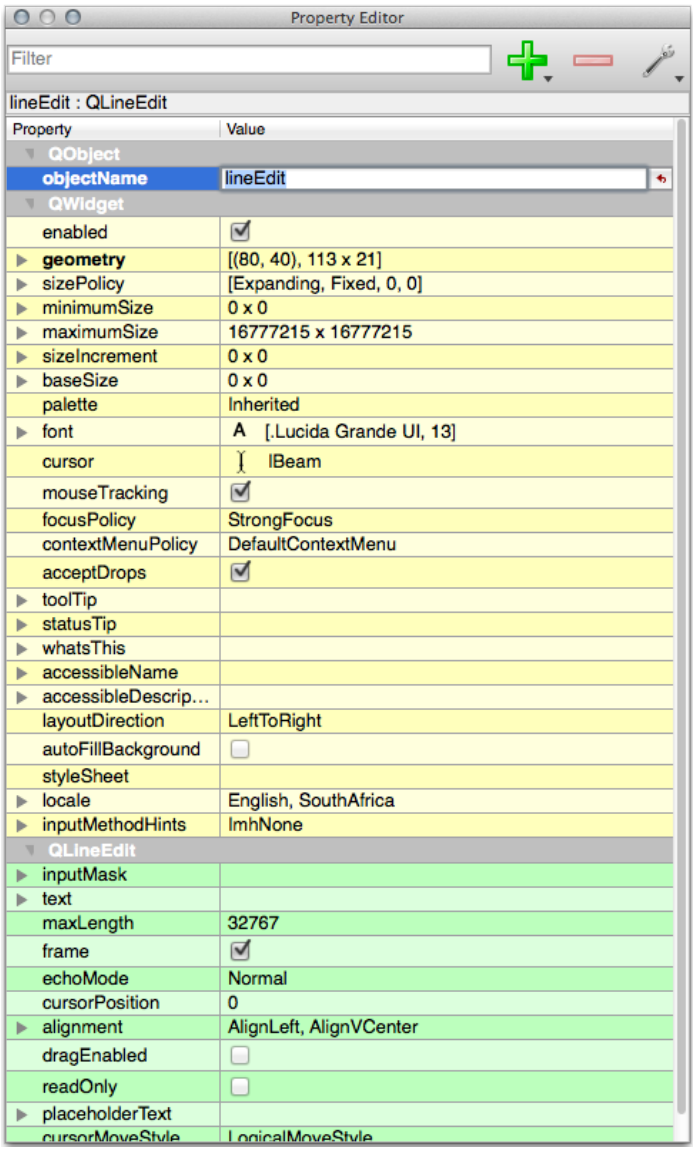
5.3.6 Идем дальше: Создание новой формы

Теперь мы хотим создать собственную пользовательскую форму для фазы сбора данных атрибутов. Для этого необходимо установить Дизайнер QT (только для лица, создающего формы).

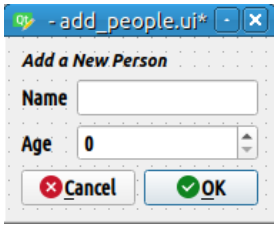
1. Запустить *QT Designer*.
2. В появившемся диалоговом окне создайте новое диалоговое окно:



3. Найдите *Widget Box* в левой части экрана (по умолчанию). Он содержит элемент с именем *Line Edit*.
4. Кликните и перетащите этот элемент в форму. Это создаст новый *Line Edit* в форме.
5. При выборе нового элемента редактирования строки его свойства будут отображаться на боковой стороне экрана (по умолчанию справа):



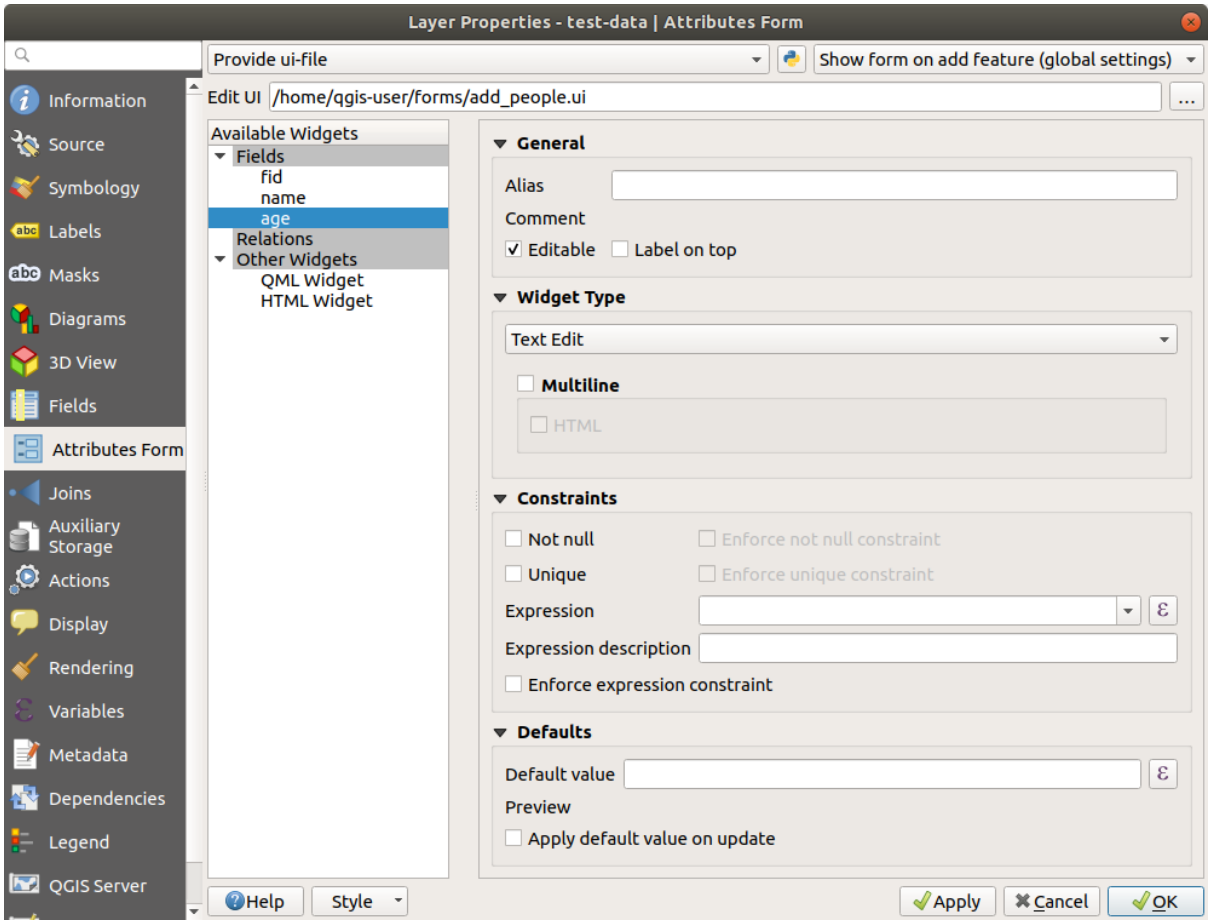
6. Задайте для его имени значение `name`.
7. Используя тот же подход, создайте новый *Spin Box* Box и задайте для его имени значение `age`.
8. Добавьте *Label`с текстом жирным шрифтом ``Add a New Person`* (просмотрите *свойства* объекта, чтобы узнать, как это задать). Можно также задать заголовок самого диалогового окна (вместо добавления надписи).
9. Добавьте *Label* к вашему *Line Edit* и *Spin Box*.
10. Упорядочить элементы по собственному желанию.
11. Кликните в любом месте диалогового окна.
12. Найдите кнопку *Компоновка в макете формы* (по умолчанию в инструментах вдоль верхнего края экрана). При этом диалоговое окно будет размещено автоматически.
13. Установите максимальный размер диалогового окна (в его свойствах) равным `200` (ширина) на `150` (высота).
14. Теперь форма должна выглядеть примерно так:



15. Сохраните свою новую форму как `exercise_data/forms/add_people.ui`
16. Когда закончится сохранение, можете закрыть *Qt Designer*

5.3.7 Идем дальше: Связывание формы со слоем

1. Вернитесь в QGIS.
2. Кликните два раза на слой *test-data* в легенде для того чтобы открыть свойства.
3. Кликните на *Attributes Form* в диалоговом окне *Layer Properties*.
4. В раскрывающемся списке *Attribute editor layout* выберите *Provide ui-file*.
5. Кликните кнопку многоточия и выберите только что созданный файл `add_people.ui`:



6. Кликните *OK* в диалоговом окне *Layer Properties*.
7. Переходите в режим редактирования и берите новую точку.

8. При этом будет представлено ваше пользовательское диалоговое окно (вместо общего, которое обычно создает QGIS).
9. Если вы кликните по одной из точек с помощью инструмента  Identify Features теперь можно вызвать форму, кликнув правой кнопкой мыши в окне определения результатов, и выбрать *View Feature Form* в контекстном меню.
10. Если вы находитесь в режиме редактирования для этого слоя, то вместо этого контекстное меню будет отображать *Edit Feature Form*. После этого можно скорректировать атрибут в новой форме даже после первоначальной записисъемке точки.

5.3.8 В заключении

С помощью форм можно облегчить себе жизнь при редактировании или создании данных. Редактируя типы виджетов или создавая совершенно новую форму с нуля, можно контролировать работу того, кто оцифровывает новые данные для этого слоя, минимизируя тем самым недоразумения и ненужные ошибки.

5.3.9 Дополнительные материалы для чтения

Если вы завершили усложнённый раздел выше и знаете о программе Python, вы можете посмотреть `this blog entry `_ запись блога о создании пользовательских форм функций с логикой Python, которая позволяет использовать расширенные функции, включая проверку данных, автозавершение и т.д.

5.3.10 Что дальше?

Открытие формы для идентификации функции является одним из стандартных действий, которые может выполнить QGIS. Однако его можно также направить на выполнение определяемых пользователем действий. Это тема следующего занятия.

5.4 Занятие: Действия

Теперь, когда вы видели действие по умолчанию в предыдущем занятии, пришло время определить собственные действия.

Действие - это то, что происходит когда вы нажимаете на функцию. Оно может добавить много дополнительных функциональных возможностей на карту, позволяя получить дополнительную информацию об объекте, для примера. Назначение действий может добавить на карту совершенно новое измерение!

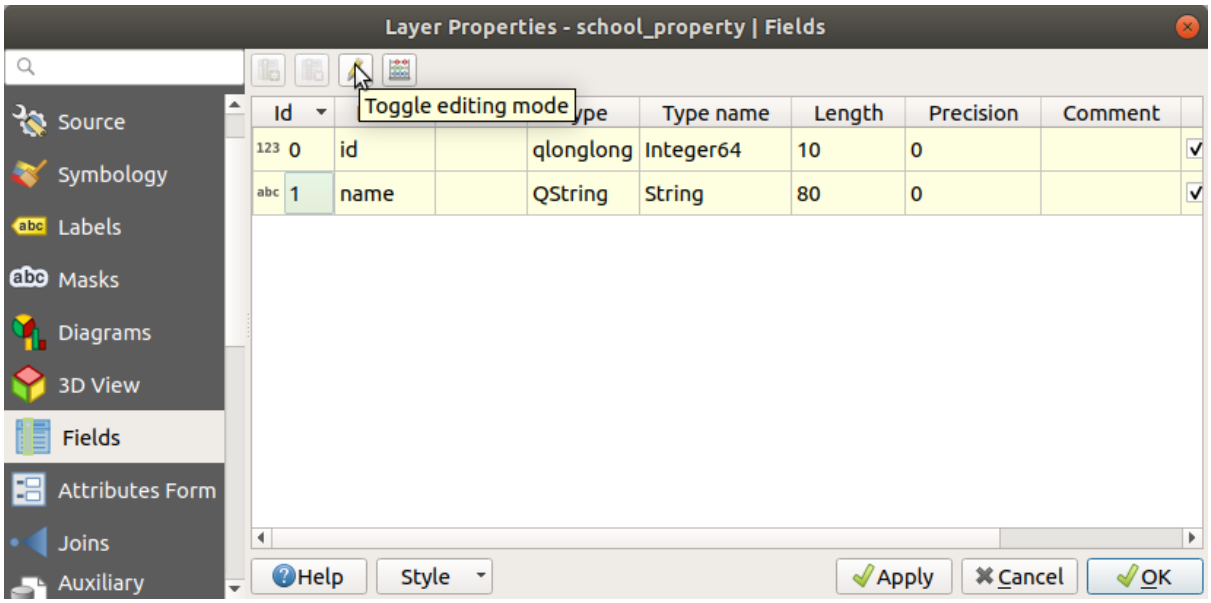
Цель этого занятия: Научиться добавлять пользовательские действия.

На этом занятии вы будете использовать слой *school_property*, который вы ранее создавали previously. Примеры данных включают фотографии каждого из трех оцифрованных свойств. Мы собираемся связать каждое свойство с его изображением. Затем создадим действие, которое откроет изображение для свойства при щелчке на свойстве.

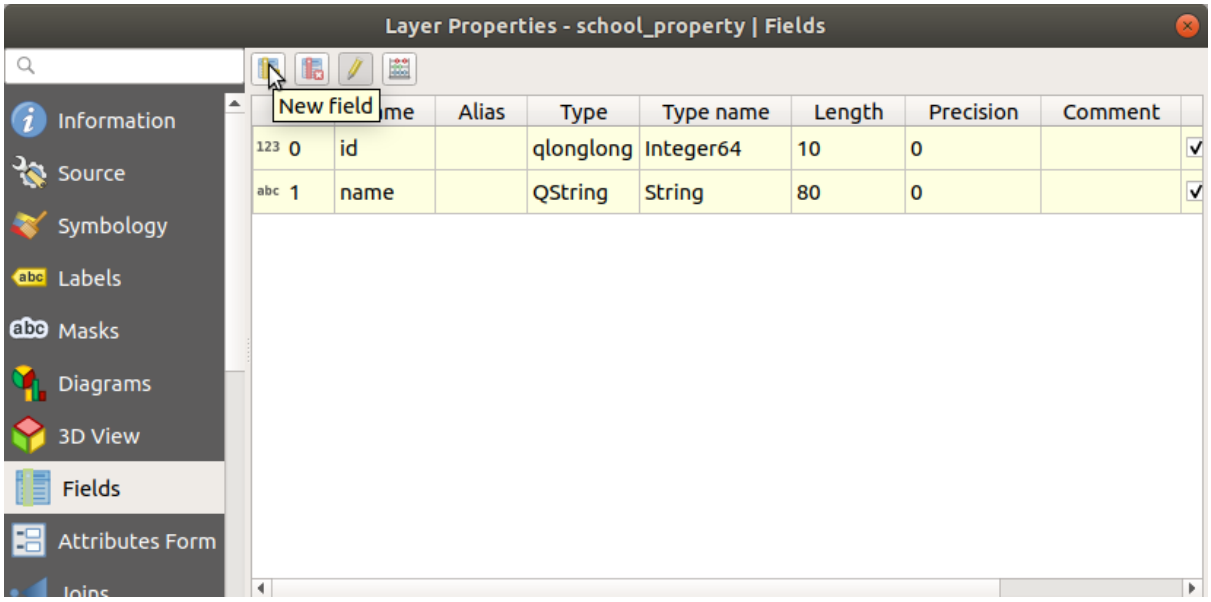
5.4.1 Идем дальше: Добавление поля для изображений

В слое `school_property` пока не удается связать изображение со свойством. Сначала создадим поле для этой цели.

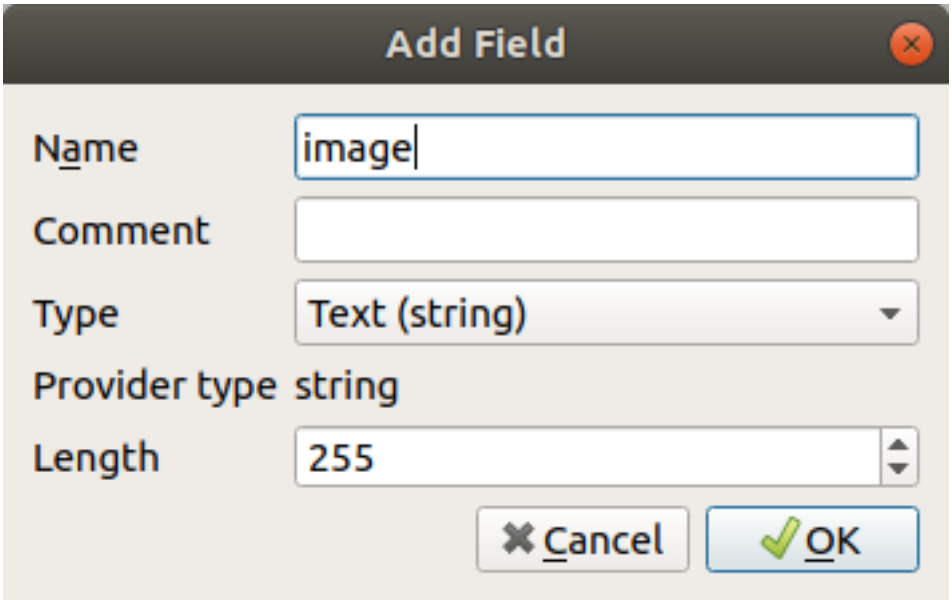
1. Откройте диалоговое окно *Layer Properties*.
2. Кликните *Fields*.
3. Режим редактирования:



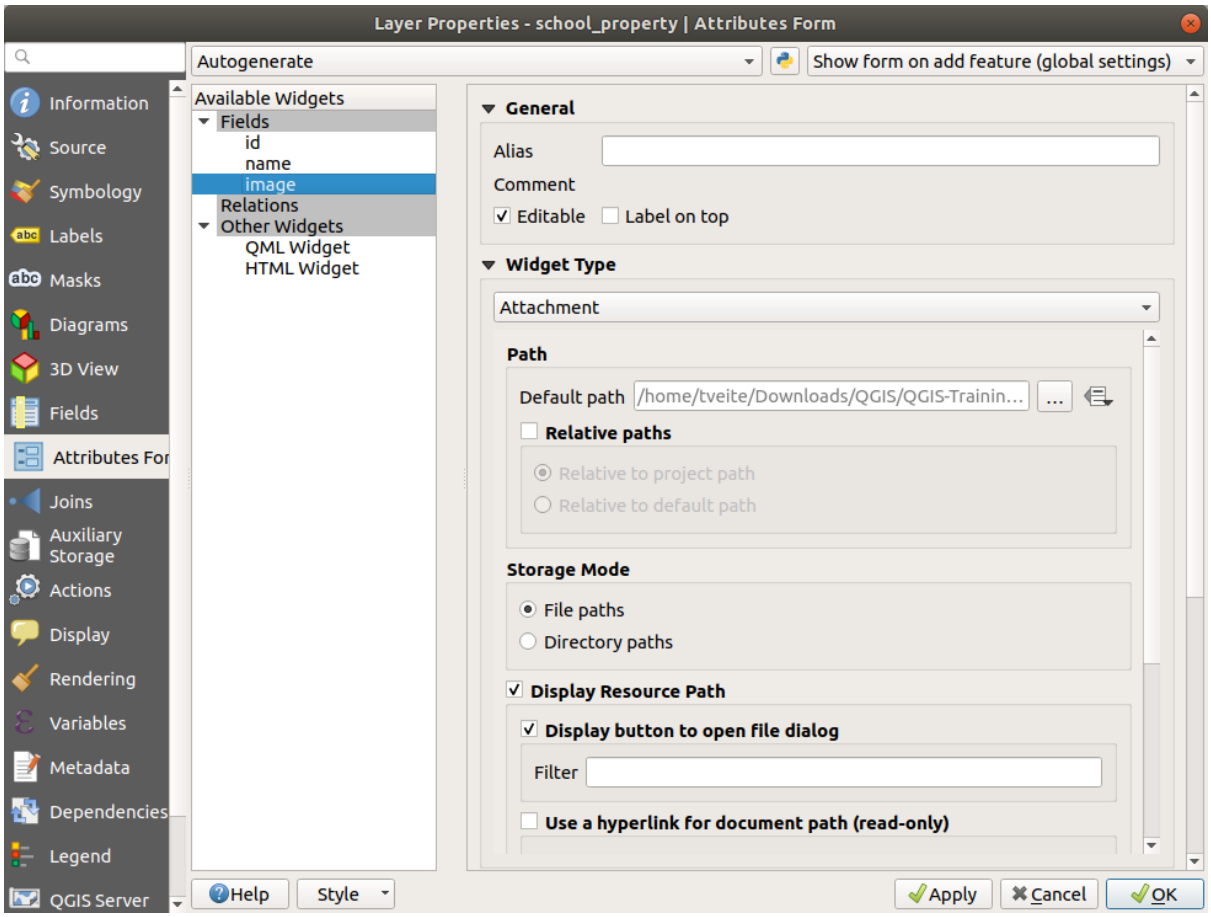
4. Добавьте новую колонку:



5. Ниже введите значения:



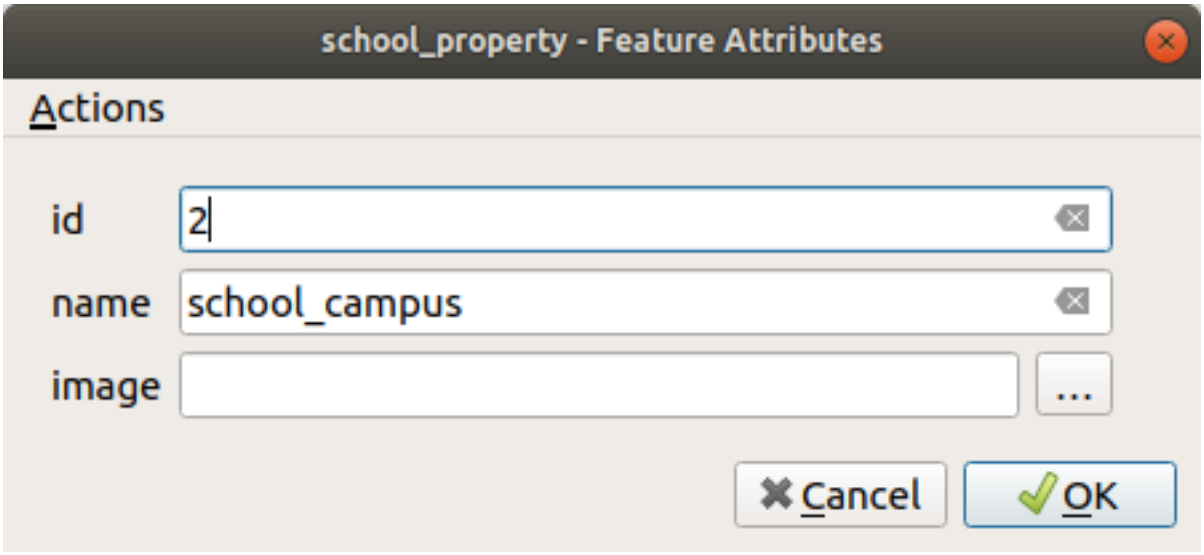
6. После того как вы создали поле перейдите к *Attributes Form* и выберите поле *image`*.
7. Задать тип виджета *Widget Type* для вложения *Attachment*:



8. Кликните *OK* в диалоговом окне *Layer Properties*.
9. Используйте инструмент *Identify* для клика на одну из трех функций в слое *school_property*.

   Поскольку вы все еще находитесь в режиме редактирования, диалоговое окно должно быть активным

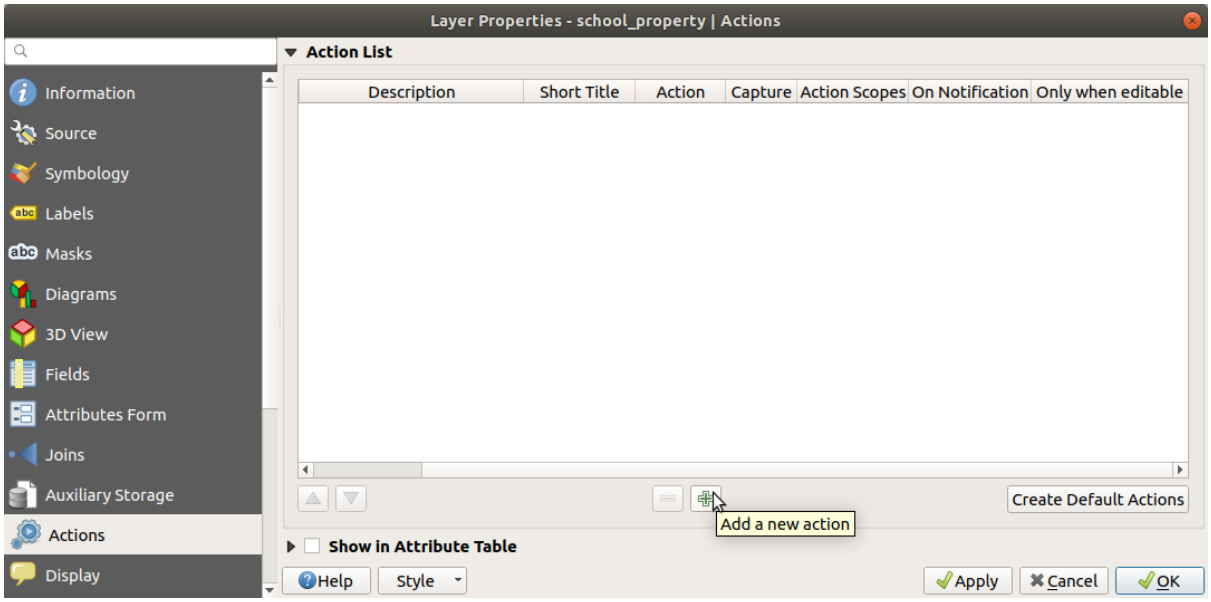
и выглядеть следующим образом:



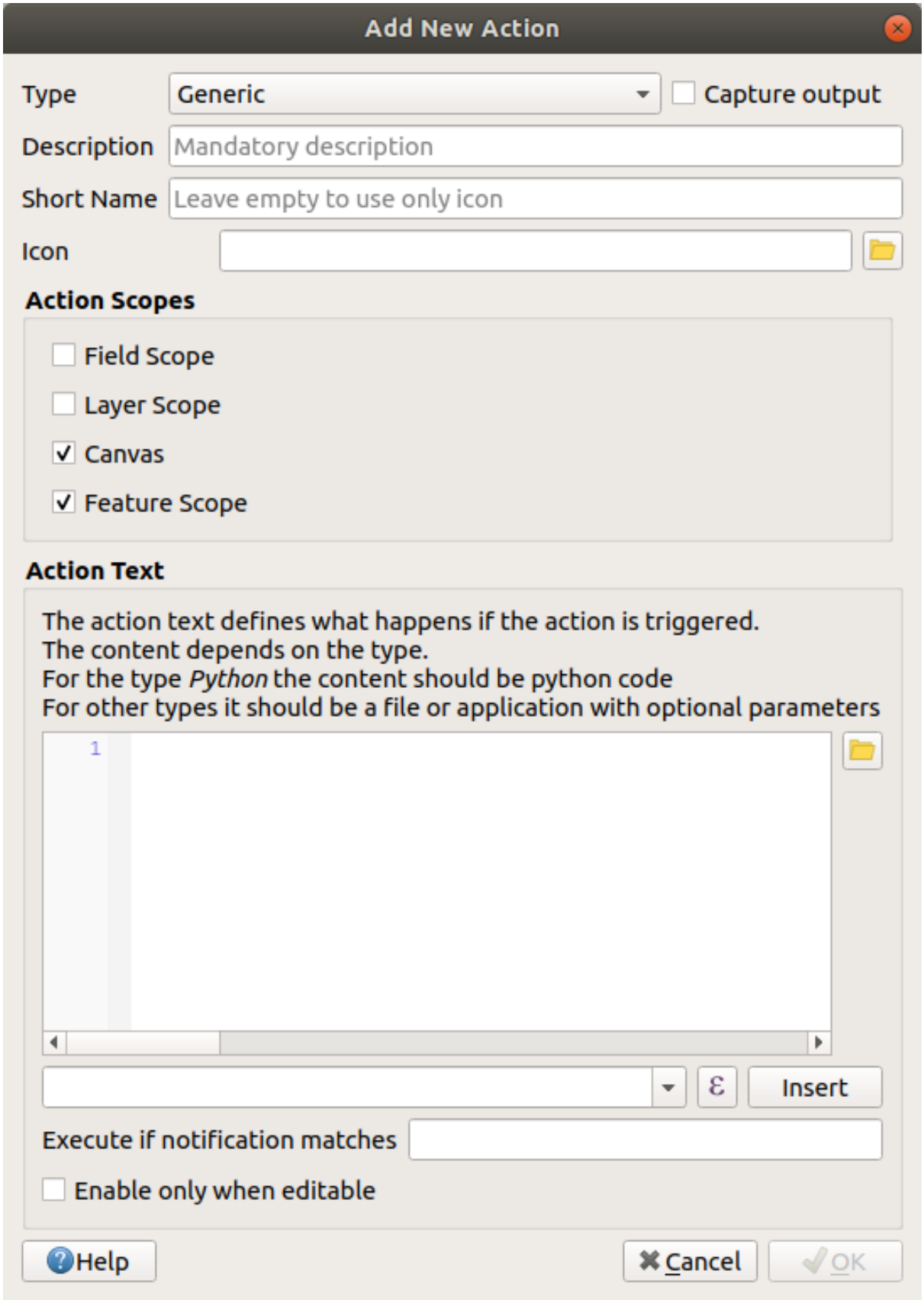
10. Кликните на кнопку просмотра (… рядом с полем *image*).
11. Выберите путь к вашему изображению. Изображения в `exercise_data/school_property_photos/` и имеют то же имя, что и функции, с которыми они должны быть связаны.
12. Кликните *OK*.
13. Свяжите все изображения с правильными функциями с помощью этого метода.
14. Сохраните изменения и выйдите из режима редактирования.

5.4.2 Идем дальше: Создание действия

1. Откройте вкладку *Actions* для слоя *school_property* и кликните на кнопку Add a new action.



2. В диалоговом окне *Add New Action* введите слова `Show Image` в поле *Description*:



Что делать дальше, зависит от операционной системы, поэтому выберите подходящий курс:

- Windows

Кликните *Type* и выберите *Open*.

- Ubuntu Linux

  В поле *Action*, напишите `eog` для Средства просмотра изображений Gnome *Gnome Image Viewer*, или напишите `display` чтобы использовать *ImageMagick*. Помните, что нужно ставить пробел после команды!

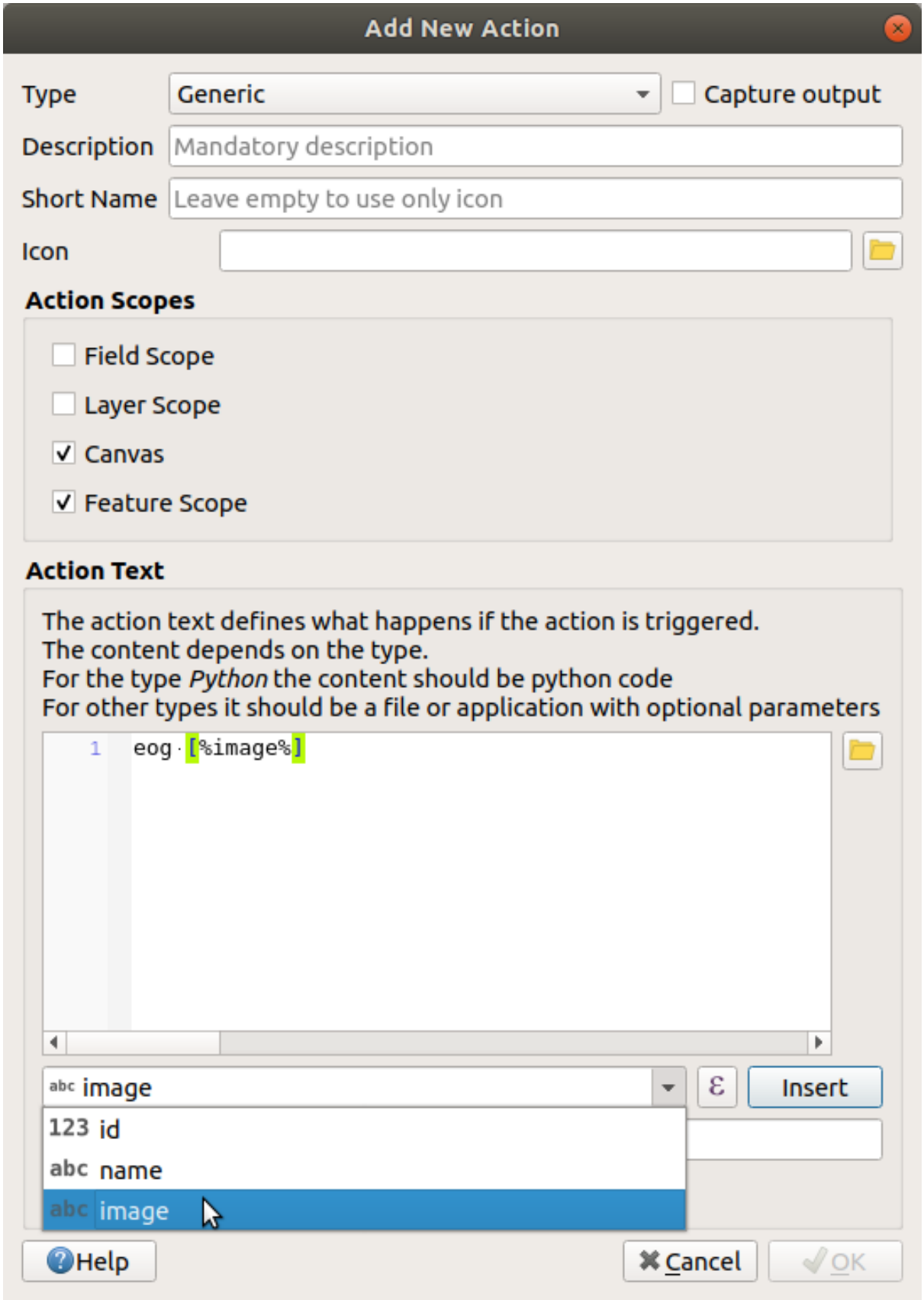
- MacOS

  1. Жмите на выпадающий *Type* и выберите *Mac*.
  2. В *Action*, пишите `open`. Помните, что нужно ставить пробел после команды!

Теперь можно продолжить запись команды.

Вы хотите открыть изображение, и QGIS знает, где оно находится. Все, что ему нужно сделать, это сказать *Action* там, где изображение.

3. Выберите *image* из списка:

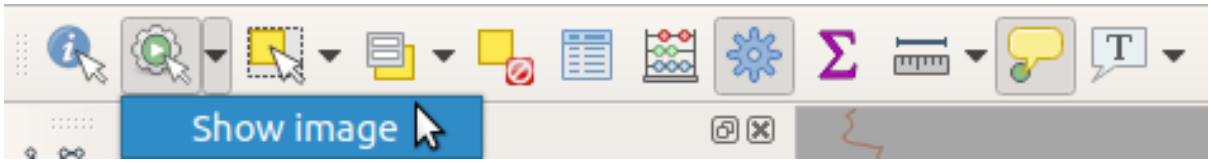


4. Кликните на кнопку *Insert field*. QGIS добавит фразу `[% "image" %]` в поле *Action Text*.
5. Кликните на кнопку *OK*, чтобы закрыть диалоговое окно *Add New Action*.

6. Кликните на кнопку *OK* чтобы закрыть диалоговое окно *Layer Properties*.

Теперь пришло время протестировать новое действие:

1. Кликните на слой *school_property* на панели *Layers* чтобы оно выделилось.
2. Найдите кнопку  Run feature action (в *Attributes Toolbar*).
3. Кликните стрелку вниз справа от этой кнопки. Для этого слоя на данный момент определено только одно действие, которое только что создано.



4. Кликните кнопку, чтобы активировать инструмент.
5. С помощью этого инструмент кликните любое из трех свойств школы.

   Должно открыться изображение для этого свойства.

5.4.3  Идем дальше: Поиск в Интернете

Допустим, мы смотрим на карту и хотим узнать больше о районе, в котором находится ферма. Предположим, вы ничего не знаете о данной местности и хотите найти общую информацию о ней. Ваш первый импульс, учитывая, что вы используете компьютер прямо сейчас, вероятно, будет найти в Google название местности. Давайте попросим QGIS сделать это автоматически для нас!

1. Откройте таблицу атрибутов в слое *landuse*.

   Мы будем использовать поле `name` для каждого района землепользования чтобы найти в Google.
2. Закройте таблицу атрибутов.
3. Вернитесь к *Actions* в *Layer Properties*.
4. Кликните на кнопку *Create Default Actions*, чтобы добавить ряд предварительно определенных действий.
5. Удалите все действия кроме *Open URL* с коротким именем *Search Web* с помощью кнопки  Remove the selected action, которая расположена ниже.
6. Кликните два раза на оставшееся действие, чтобы редактировать его.
7. Поменяйте *Description* на `Google Search`, и удалите содержимое поля *Short Name*.
8. Убедитесь, что Полотно *Canvas* находится среди *Action scopes* с флажками.

   Что делать дальше, зависит от операционной системы, поэтому выберите подходящий курс:

   - Windows

     В *Type* выберите *Open*. Это позволит Windows открыть адрес Интернета в браузере по умолчанию, например Internet Explorer.
   - Ubuntu Linux

     В *Action* напишите `xdg-open`. Это позволит Ubuntu открыть интернет-адрес в браузере по умолчанию, например Chrome или Firefoxhis.
   - MacOS

     В *Action* напишите `open`. Это позволит MacOS открыть интернет-адрес в браузере по умолчанию, например Safari.

Теперь можно продолжить запись команды.

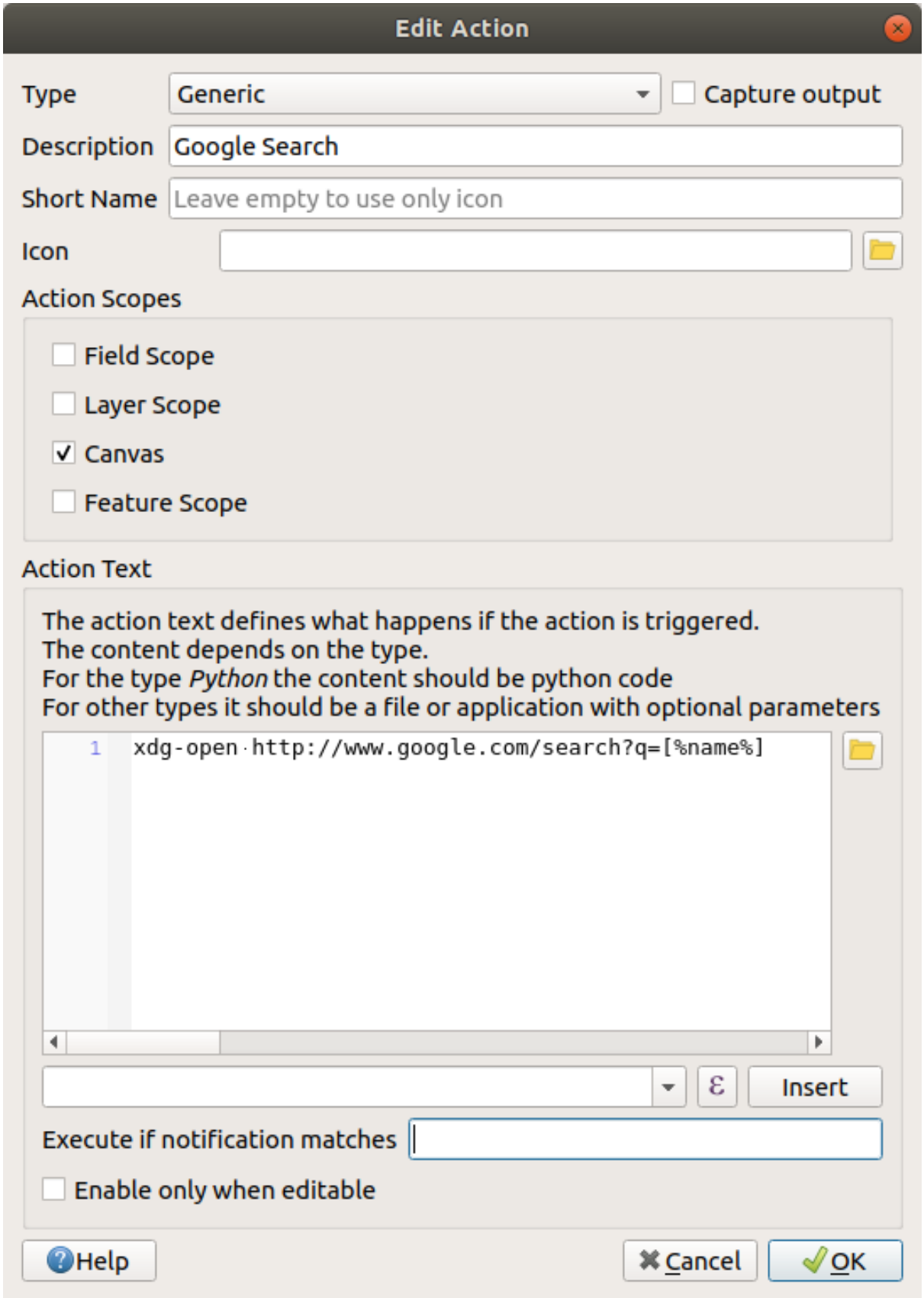
Независимо от того, какую команду вы использовали выше, вы должны сообщить ей, какой адрес Интернета открыть следующим. Вы хотите, чтобы она посетила Google и искала фразу автоматически.

Обычно при использовании Google вы вводите свою поисковую фразу в строку поиска Google. Но в этом случае вы хотите, чтобы ваш компьютер сделал это для вас. То, как вы зададите Google поиск чего-то (если вы не хотите использовать его строку поиска напрямую) является путем предоставления адреса вашему интернет-браузеру `https://www.google.com/search?q=SEARCH_PHRASE`, где `SEARCH_PHRASE`` это то что вы ищете. Поскольку мы пока не знаем, какую фразу искать, мы просто введем первую часть (без поисковой фразы).

9. В поле *Action* впишите `https://www.google.com/search?q=`. Не забывайте добавить пробел после начальной команды перед тем как впишите это!

   Теперь вы хотите, чтобы QGIS сказал браузеру, чтобы Google искал значение имени `name` для любой функции, которую вы можете нажать.

10. Выберите поле *name*.
11. Кликните кнопку *Insert*:



Это означает, что QGIS собирается открыть браузер и отправить его по адресу `https://www.google.com/search?q=[% "name" %]`. Команда `[% "name" %]` говорит QGIS использовать содержимое поля `name` как фразу поиска.

Так, например, если местность землепользования, на которую вы нажимаете, называется `Marloth Nature Reserve`, QGIS собирается отправить браузер на `https://www.google.com/search?q=Marloth%20Nature%20Reserve`, что приведет к тому, что ваш браузер посетит Google, который в свою очередь будет искать «Marloth Nature Reserve».

12. Если вы еще не сделали этого, настройте все, как описано выше.
13. Кликните на кнопку *OK*, чтобы закрыть диалоговое окно *Add New Action*.
14. Кликните на кнопку *OK* чтобы закрыть диалоговое окно *Layer Properties*.

Теперь, чтобы проверить новое действие.

1. Слой *landuse* должен быть активен на панели *Layers*, теперь кликните на стрелку вниз справа от кнопки  Run feature action, и выберите единственное действие (`Google Search`), определенное для этого слоя.
2. Кликните на любом участке землепользования, который можно увидеть на карте. Ваш браузер откроется и начнет поиск в Google места, записанного как значение `name` на этой местности.

Примечание: Если действие не работает, убедитесь, что все введено правильно; опечатки часто встречаются при таких работах!

5.4.4  Идем дальше: Открытие веб-страницы непосредственно в QGIS

Выше, вы видели, как открыть веб-страницу во внешнем браузере. Есть некоторые недостатки в таком подходе, так как он добавляет неизвестную зависимость - будет ли у конечного пользователя программное обеспечение, необходимое для выполнения действия в своей системе? Как вы видели, они не обязательно даже имеют один и тот же вид базовой команды для одного и того же вида действий, если вы не знаете, какую OS они будут использовать. В некоторых версиях OS вышеуказанные команды для открытия браузера могут вообще не работать. Это может быть непреодолимой проблемой.

Однако QGIS находится в топе невероятно мощной и универсальной библиотеки Qt. Кроме того, действия QGIS могут быть произвольными, маркированными (т.е. с использованием информации переменной на основе содержимого атрибута поля) командами Python!

Теперь вы увидите, как использовать действие python для отображения веб-страницы. Здесь лежит та же самая общая идея, что и открытие сайта во внешнем браузере, но она не требует браузера в системе пользователя, поскольку он использует класс Qt QWebView (который является HTML-виджетом на основе webkit) для отображения содержимого во всплывающем окне.

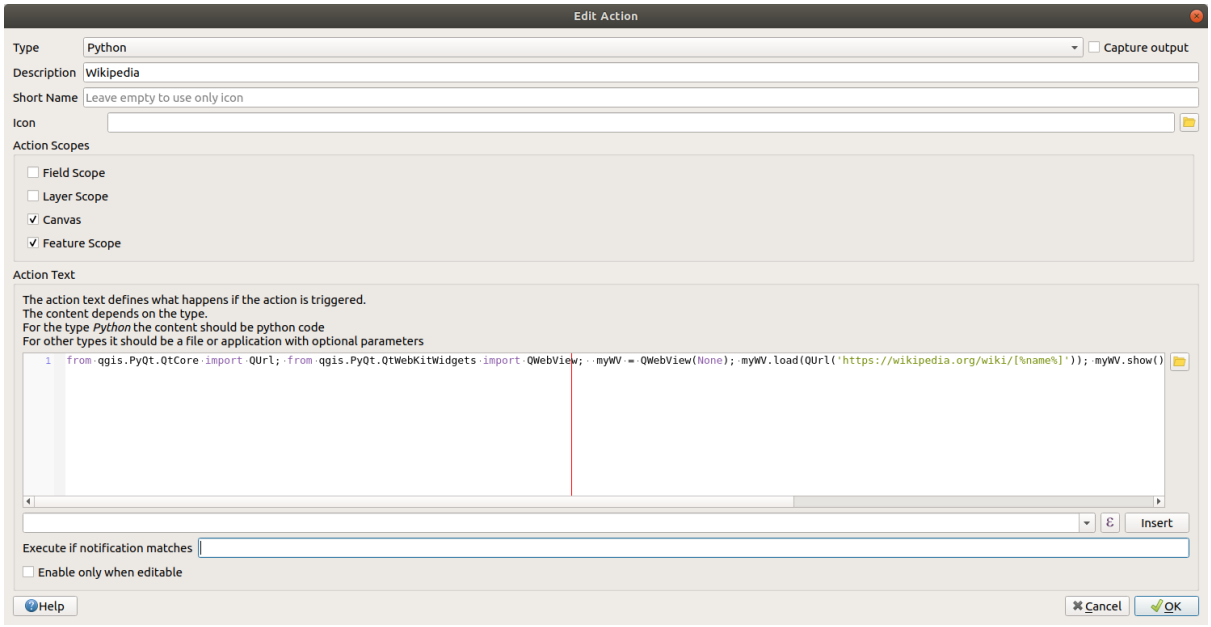
Давайте используем Википедию на этот раз. Таким образом, запрашиваемый URL будет выглядеть следующим образом:

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

Чтобы создать действие слоя:

1. Откройте диалоговое окно *Layer Properties* и перейдите к вкладке *Actions*.
2. Настройте новое действие, используя для него следующие свойства действия:
   - *Type*: `Python`
   - *Description*: `Wikipedia`
   - *Action Text* (все в одну линию):

```
from qgis.PyQt.QtCore import QUrl; from qgis.PyQt.QtWebKitWidgets import␣
↪QWebView; myWV = QWebView(None); myWV.load(QUrl('https://wikipedia.org/
↪wiki/[%name%]')); myWV.show()
```



Здесь происходит несколько вещей:

- Весь код Python находится в одной строке с разделяющими командами точка с запятой (вместо новых строк, это обычный способ разделения команд Python).
- `[%name%]` будет заменено фактическим значением атрибута при вызове действия (как и ранее).
- Код просто создает новый экземпляр `QWebView`, задает его URL, а затем вызывает `show()` на нем, чтобы сделать его видимым в виде окна на рабочем столе пользователя.

Заметьте, что это несколько надуманный пример. Python работает с семантически значимым отступом, поэтому разделять вещи точкой с запятой не лучший способ его написания. Таким образом, в реальном мире вы с большей вероятностью импортируете логику из модуля Python, а затем вызовете функцию с атрибутом поля в качестве параметра.

Этот подход можно также использовать для того чтобы показать изображение без необходимости наличия у пользователей определенного средства просмотра изображений в их системе.

3. Попробуйте использовать описанные выше методы для загрузки страницы Википедии с помощью только что созданного действия Википедии.

5.4.5 В заключении

Действия позволяют предоставить вашей карте дополнительные функциональные возможности, полезные для конечного пользователя, который просматривает ту же карту в QGIS. Благодаря тому, что можно использовать команды оболочки для любой операционной системы, а также для Python, ваши возможности безграничны с точки зрения функций, которые вы могли бы включить!

5.4.6 Что дальше?

Теперь, когда вы освоили все виды создания векторных данных, вы научитесь анализировать данные для решения проблем. А это тема следующего модуля.

ГЛАВА 6

Модуль: Векторный анализ

Теперь после того как вы отредактировали несколько функций вам наверное захочется узнать, что еще можно с ними сделать. То, что у вас есть функции с атрибутами - это хорошо, но когда все сделано и сказано, это не скажет вам ничего такого, чего не может сделать обычная карта, не входящая в систему GIS.

Основное преимущество GIS заключается в том, что *она может дать ответы на вопросы*.

В следующих трех модулях мы попытаемся дать ответ на вопрос исследования, используя функции ГИС. Например, вы являетесь агентом по продаже недвижимости и ищете жилое помещение в Свеллендаме / Swellendam для своих клиентов со следующими критериями:

1. Жилое помещение должно находиться в Свеллендаме / Swellendam.
2. Оно должно находиться в пределах умеренного расстояния от школы (например 1 км).
3. Его площадь должна быть более 100 квадратных метров.
4. Находиться на расстоянии менее чем 50 метров от главной дороги.
5. Находиться на расстоянии менее чем 500 метров от какого-нибудь ресторана.

В рамках следующих нескольких модулей мы воспользуемся возможностями инструментов ГИС-анализа, чтобы найти подходящие фермы для этого нового жилого комплекса.

6.1 Занятие: пере-проецирование и преобразование данных

Давайте еще раз поговорим о справочной системе координат (ССК). Мы вкратце касались этой темы раньше, но не обсуждали, что это означает на практике.

Цель этого урока: Пере-проецирование и преобразование векторных наборов данных.

6.1.1  Идем дальше: Проецирование

ССК, в которой сейчас находятся все данные, а также сама карта, называется WGS84. Это очень распространенная географическая система координат (ГСК) для представления данных. Но имеется проблема, которую мы далее изучим.

1. Сохраните вашу текущую карту.
2. Затем откройте карту мира, которую вы найдете здесь `exercise_data/world/world.qgs`.
3. Приближайте Южную Африку с помощью команды *Zoom In*.
4. Попробуйте установить масштаб в поле *Scale*, которая находится в *Панели Состояния* в нижней части экрана. Находясь над Южной Африкой, установите значение `1:5 000 000` (от одного до пяти миллионов).
5. Перемещайтесь по карте и одновременно смотрите на поле *Scale*.

Вы заметили изменение масштаба? Потому, что вы удаляетесь от той точки, на которую вы увеличили масштаб `1:5 000 000`, который находился в центре экрана. Вокруг этой точки масштаб другой.

Для того, чтобы понять почему так происходит, представьте себе земной шар. По нему проходят линии с севера на юг. Эти линии долготы далеко друг от друга на экваторе, но встречаются на полюсах.

В ГСК вы работаете над этой сферой, но ваш экран плоский. Когда вы пытаетесь изобразить сферу на плоской поверхности, возникает искажение, подобное тому, что произошло бы, если бы вы разрезали теннисный мяч и попытались его расплющить. На карте это означает, что линии долготы находятся на одинаковом расстоянии друг от друга, даже на полюсах (там, где они должны пересекаться). Это означает, что по мере того как вы удаляетесь от экватора на вашей карте, масштаб объектов, которые вы видите, становится все больше и больше. На практике это означает, что на нашей карте нет постоянного масштаба!

Чтобы решить эту проблему, давайте вместо этого используем проекционную систему координат (ПСК). ПСК «проецирует» либо конвертирует данные таким образом, чтобы учитывать изменение масштаба и корректирует его. Поэтому, чтобы масштаб был постоянным, мы должны пере-проецировать наши данные для того чтобы использовать ПСК.

6.1.2  Идем дальше: Пере-проецирование «на лету»

По умолчанию QGIS пере-проецирует данные «на лету». Это означает, что даже если сами данные находятся в другой ССК, QGIS может проецировать их, как если бы они находились в ССК, которую вы выбрали.

Вы можете изменить ССК проекта, нажав на  Current projection в правом нижнем углу QGIS.

1. В диалоговом окне, которое появляется, введите слово `global` в поле *Filter*. Несколько ССК должны появиться в поле *Predefined Reference Systems* ниже.
2. Выберите *WGS 84 / NSIDC EASE-Grid 2.0 Global | EPSG:6933*, кликнув по ней, а затем кликните на *OK*.

   Обратите внимание, как поменялся облик Южной Африки. Все проекции работают, изменяя видимые формы объектов на Земле.
3. Увеличьте масштаб до `1:5 000 000` снова, как и раньше.
4. Прокрутите карту.

   Обратите внимание на то, что масштаб остался прежним!

Пере-проецирование «на лету» также используется для объединения наборов данных, которые находятся в разных ССК.

1. Добавьте на карту еще один слой вектора, который содержит данные только по Южной Африке. Он появится как `exercise_data/world/RSA.shp`.

2. Загрузите его. Можно быстро посмотреть на его ССК после наведения указателя мыши на слой в легенде. Она такая `EPSG:3410`.

   Что вы заметили?

   Слой виден, даже если он имеет другую ССК, которая отличается от ССК *continents*.

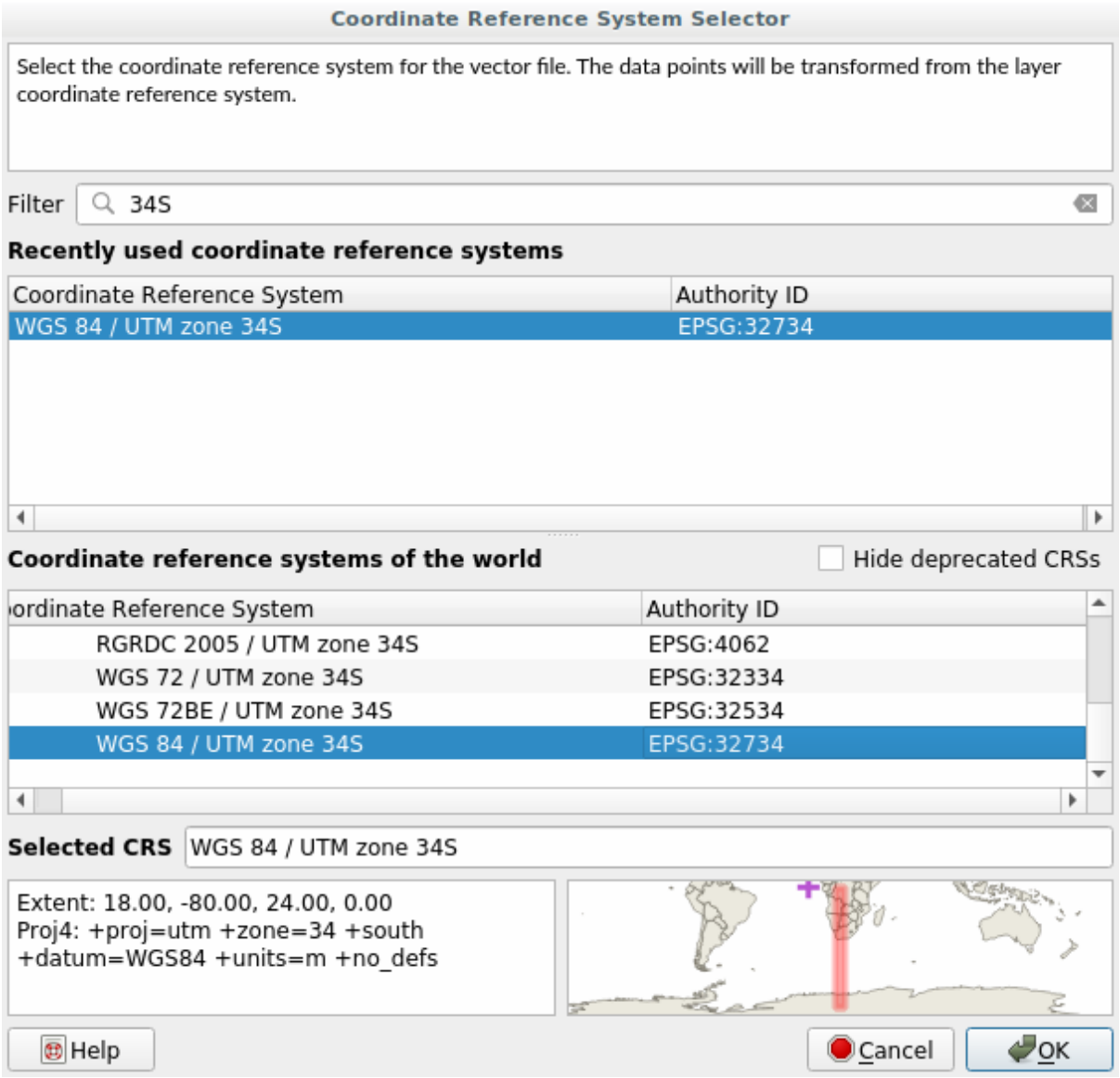
6.1.3  Идем дальше: Сохраняем набор данных в другой справочной системе координат (ССК)

Иногда вам необходимо экспортировать существующий набор данных с другой ССК. Как мы потом уводим в последующем уроке, если вам надо рассчитать расстояния на слое, всегда лучше иметь слой в проецированной системе координат.

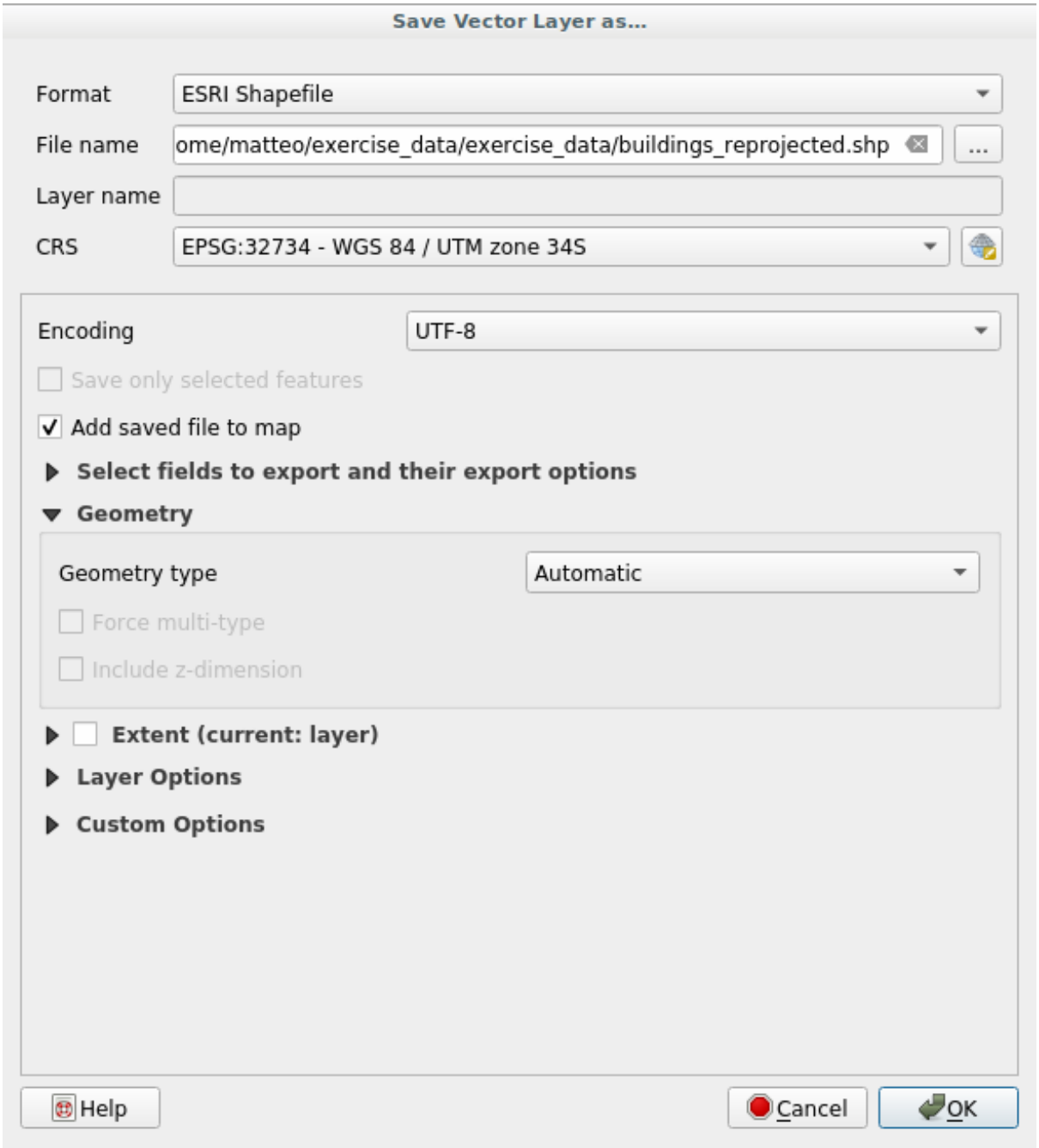
Имейте в виду, что пере-проецирование, осуществленное «на лету», имеет отношение к **проекту**, а не к отдельным слоям. Это означает, что слой может иметь другую ССК, которая отличается от проектного, даже если вы видите его в *правильном* положении.

Вы сможете легко экспортировать слой с другим ССК.

1. Добавьте набор данных `buildings` из `training_data.gpkg`.
2. Кликните правой кнопкой мыши на слой `buildings` в панели *Layers*.
3. Выберите *Export* ► *Save Features As...* в появившемся меню. Вы увидите диалоговое окно *Save Vector Layer as...*.
4. Кликни на кнопку *Browse* рядом с полем *File name*.
5. Перейдите к `exercise_data/` и укажите имя нового слоя как `buildings_reprojected.shp`.
6. Измените значение *CRS*. В раскрывающемся меню появится только недавно использованные ССК. Кликните на кнопку  Select projection рядом с раскрывающимся меню.
7. Появится диалоговое окно *Coordinate Reference System Selector*. В поле *Filter* сделайте поиск `34S`.
8. Выберите *WGS 84 / UTM zone 34S | EPSG:32734* из списка.



9. Остальные параметры оставьте без изменений. Диалоговое окно *Save Vector Layer as...* выглядит теперь так:



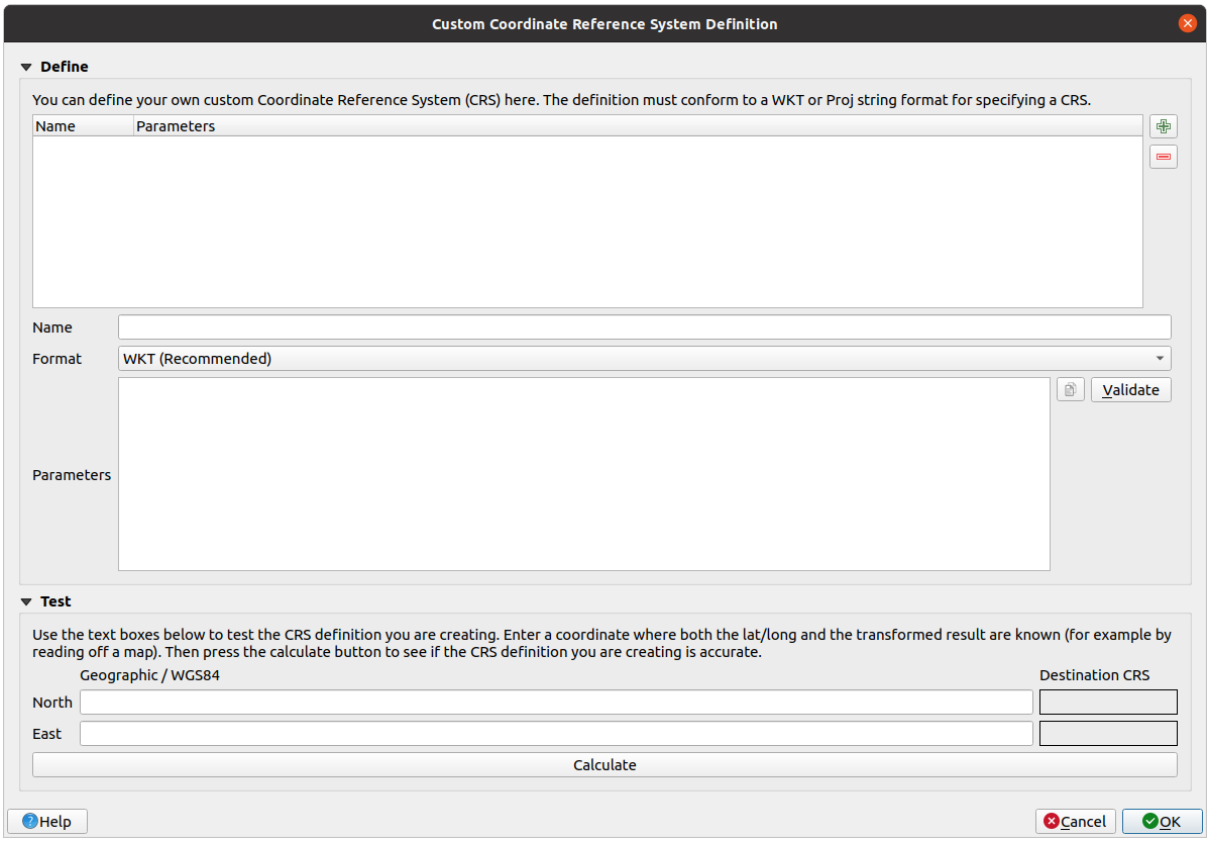
10. Кликните на кнопку *OK*.

Теперь вы сможете сравнить старую и новую проекции слоя и увидеть, что они находятся в двух разных ССК, но они все еще перекрываются.

6.1.4 Идем дальше: Создаем собственную проекцию

Кроме тех проекций, которые включены в QGIS по умолчанию, существует еще много других проекций. Вы также сможете создавать свои собственные проекции.

1. Запустите новую карту.
2. Загрузите набор данных `world/oceans.shp`.
3. Перейдите к *Settings* ► *Custom Projections...* и вы увидите это диалоговое окно.



4. Кликните на кнопку  Add new CRS для создания новой проекции.
5. Есть интересная проекция для использования, которая называется `Van der Grinten I`. Введите ее название в *Name*.

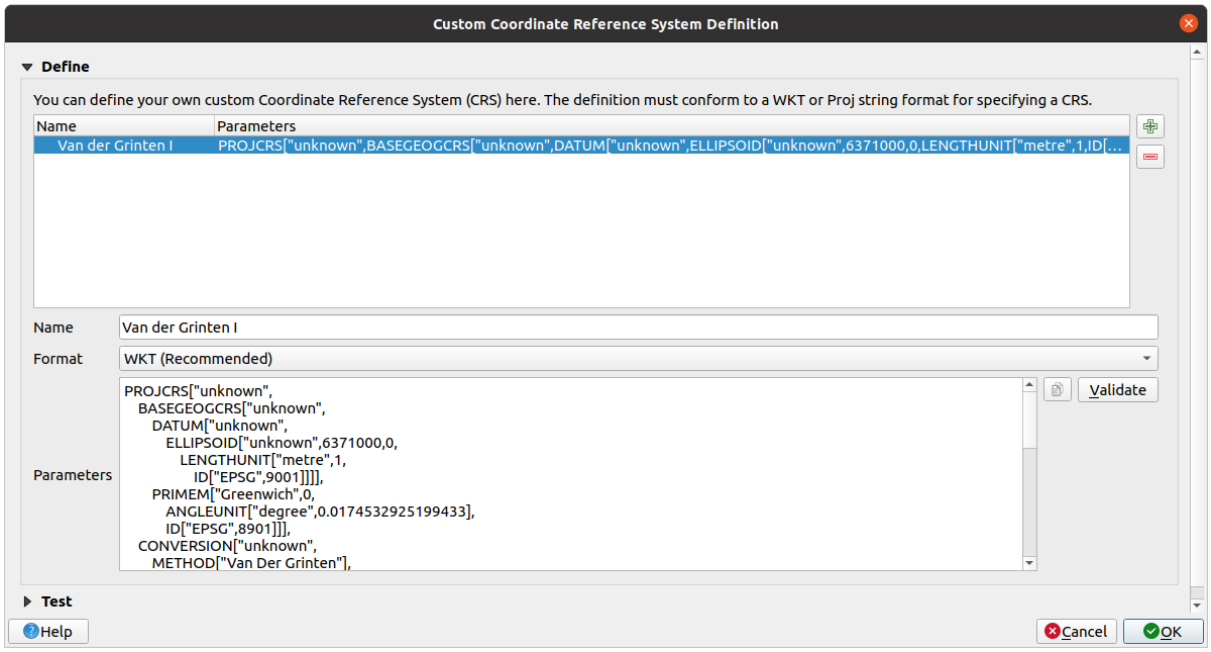
   Эта проекция представляет Землю в круглом поле вместо прямоугольного как в большинстве других проекциях.
6. В поле *Format* выберите *WKT (Recommended)*.
7. Добавьте следующую строку в поле *Parameters* field:

```
PROJCRS["unknown",
    BASEGEOGCRS["unknown",
        DATUM["unknown",
            ELLIPSOID["unknown",6371000,0,
                LENGTHUNIT["metre",1,
                    ID["EPSG",9001]]]],
        PRIMEM["Greenwich",0,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8901]]],
```

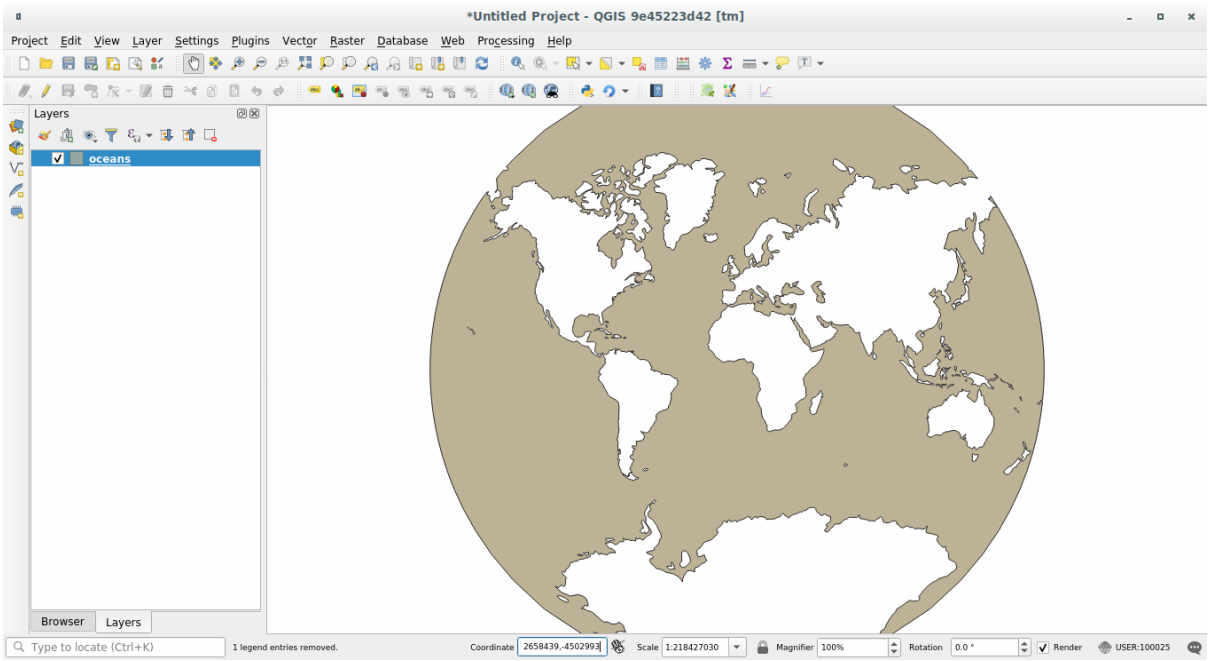
(continues on next page)

(продолжение с предыдущей страницы)

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



8. Кликните на кнопку *OK*.
9. Кликни на кнопку Текущая проекция  Current CRS справа от строки состояния для изменения проекта ССК.
10. Выберите свою новую проекцию (сделайте поиск по названию в поле *Filter*).
11. При применении этой проекции карта будет пере-проецирована следующим образом:



6.1.5 В заключении

Разные проекции полезны для разных целей. Выбрав правильную проекцию, вы можете быть уверены, что объекты на вашей карте будут представлены точно.

6.1.6 Дополнительные материалы для чтения

Материалы раздела *Продвинутый Уровень* этого урока были взяты из `этой статьи `_.

Ознакомиться дополнительной информацией о Справочной Системе Координат.

6.1.7 Что дальше?

В следующем уроке вы научитесь анализировать векторные данные с помощью различных инструментов векторного анализа QGIS.

6.2 Занятие: Векторный анализ

Векторные данные также можно анализировать, чтобы узнать, как различные объекты взаимодействуют друг с другом в пространстве. Есть множество различных функций, связанных с анализом, но мы не будем их все рассматривать. Мы лучше зададим вопрос и попытаемся найти на него ответ с помощью инструментов QGIS.

Цель этого урока: Задать вопрос и решить его с помощью инструментов анализа.

6.2.1  Процесс GIS

Перед тем как начать было бы хорошо сделать краткий обзор процесса, который надо пройти для решения проблемы. Он представлен ниже:

1. Обозначить проблему.
2. Получить данные.
3. Проанализировать проблему.
4. Представить результаты.

6.2.2  Проблема

Начнем с определения проблемы, которую нужно решить. Например, вы являетесь агентом по продаже недвижимости и ищете жилое помещение в Свеллендаме / Swellendam для своих клиентов со следующими критериями:

1. Жилое помещение должно находиться в Swellendam.
2. Оно должно находиться в пределах умеренного расстояния от школы (например, 1 км).
3. Его площадь должна быть более 100 квадратных метров.
4. Находиться на расстоянии менее чем 50 метров от главной дороги.
5. Находиться на расстоянии менее чем 500 метров от какого-нибудь ресторана.

6.2.3  Данные

Чтобы ответить на эти вопросы, нам нужны следующие данные:

1. Жилые помещения (здания) в этом районе.
2. Дороги внутри города и вокруг него.
3. Месторасположение школ и ресторанов.
4. Размер зданий.

Эти данные можно получить через OSM (OpenStreetMap) и вы узнаете, что набор данных, который вы использовали в этом руководстве, также можно использовать для выполнения этого урока.

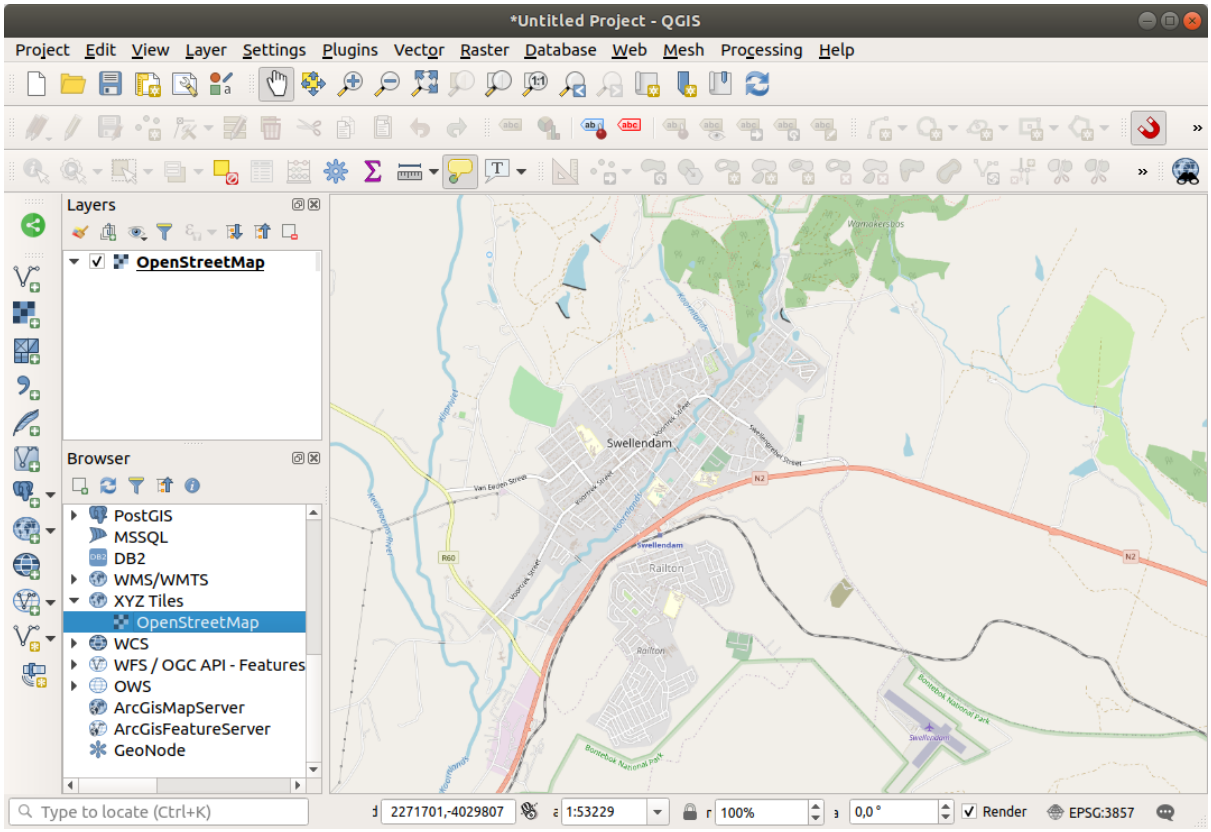
Если вы хотите загрузить данные из другой местности, перейдите в *Раздел Введение* и ознакомьтесь, как это можно сделать.

Примечание: Хотя загружаемые файлы OSM содержат согласованные поля данных, охват и детализация могут быть разными. Если вы обнаружили, что выбранный вами регион, к примеру, не содержит сведения по ресторанам, то вам, возможно, надо выбрать другой регион.

6.2.4 Идем дальше: Начать проект и получить данные

Сначала нам надо загрузить данные, с которыми мы будет работать.

1. Начать новый проект QGIS.
2. Если хотите, можете добавить фоновую карту. Откройте *Browser* и загрузите фоновую карту *OSM* из меню *XYZ Tiles*.



3. В базе данных Geopackage `training_data.gpkg` вы сможете найти большинство наборов данных, которые мы будем использовать в это разделе:

   1. `buildings`
   2. `roads`
   3. `restaurants`
   4. `schools`

   Загрузите их, а также `landuse.sqlite`.

4. Увеличьте масштаб слоя, чтобы увидеть Свеллендам в Южной Африке.

   Прежде чем пойти дальше, мы должны отфильтровать слой *roads*, чтобы работать только с некоторыми конкретными типами дорог.

   Некоторые дороги в наборах данных OSM перечислены как `unclassified`, `tracks`, `path`, а также `footway`. Нам надо исключить их из нашего набора данных и сосредоточиться на других типах дорог, которые больше подходят для этого упражнения.

   Кроме того, не все данные OSM могут быть актуализированы и мы также надо исключить значения `NULL`.

5. Кликните правой кнопкой мыши на слой `roads` и выберите *Filter…*.

6. В диалоговом окне, которое появится, мы должны отфильтровать эти объекты с помощью следующего выражения:

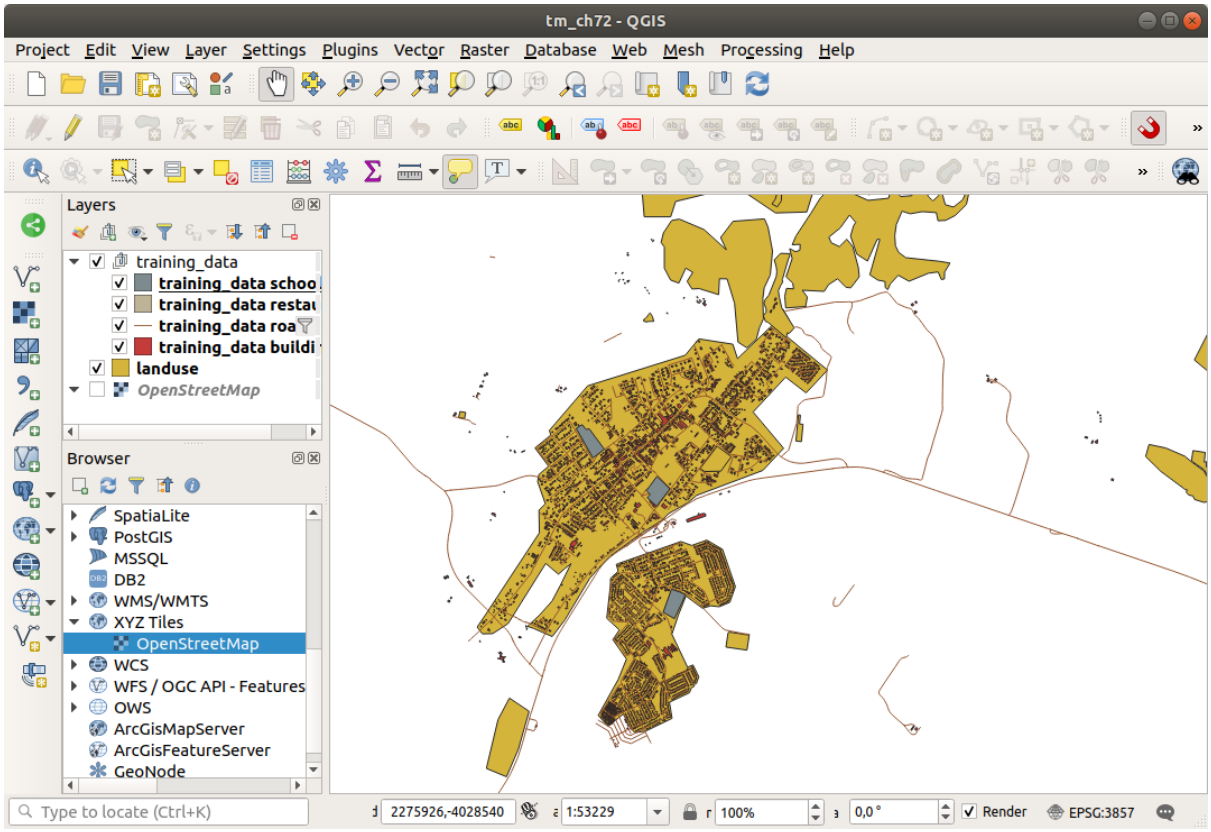
```
"highway" NOT IN ('footway', 'path', 'unclassified', 'track') AND "highway" IS␣
↪NOT NULL
```

Объединение двух команд `NOT` и `IN` исключает все объекты, которые имеют эти атрибуты значений в поле `highway`.

`IS NOT NULL` в сочетании с `AND` исключает дороги, не имеющие значения в поле `highway`.

Примечание: значок  рядом со слоем *roads*. Он поможет вам запомнить, что на этом слое активирован фильтр и потому некоторые функции могут быть недоступны в проекте.

Карта со всеми данными должна выглядеть следующим образом:



6.2.5  Попробуйте сами: Сконвертировать СКК слоев

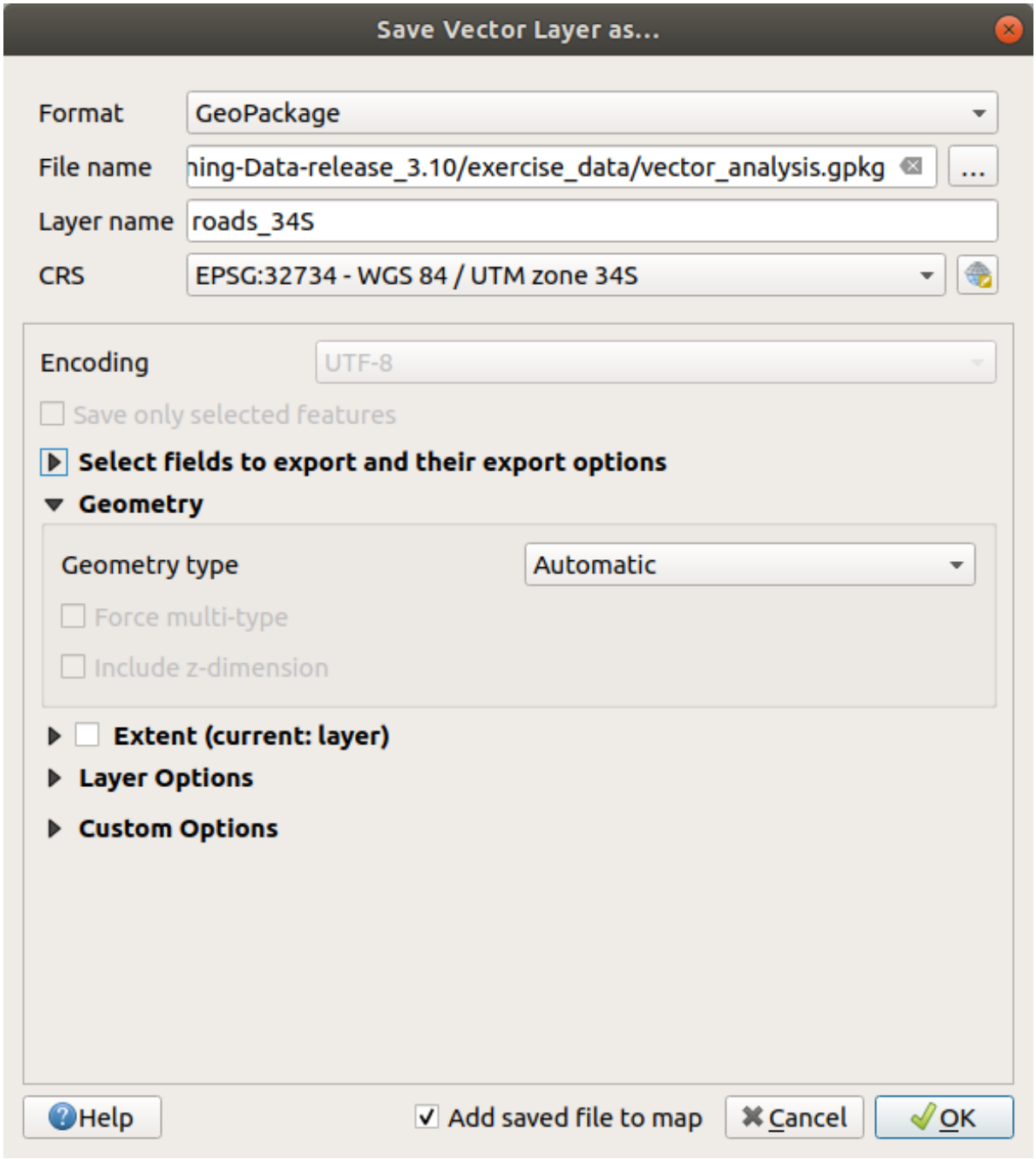
Так как мы собираемся измерить расстояния внутри наших слоев, нам надо изменить СКК слоев. Для этого нам надо выбрать каждый слой по очереди, сохранить слой по-новому с нашей новой проекцией, а затем импортировать этот новый слой на нашу карту.

У вас появится много различных опций, например, вы сможете экспортировать каждый слой как набор данных формата ESRI Shapefile, вы можете добавить слои в существующий файл GeoPackage или создать другой файл GeoPackage и заполнить его новыми пере-проецированными слоями. Мы покажем последнюю опцию, чтобы `training_data.gpkg` остался чистым. Вы сможете выбрать для себя свой лучший рабочий процесс.

Примечание: На этом примере мы используем СКК WGS 84 / UTM zone 34S, но вы должны использовать UTM СКК, который больше подходит для вашего региона.

---

1. Кликните правой кнопкой мыши на слой *roads* в панели *Layers*.
2. Кликните *Export –> Save Features As…*
3. В диалоговом окне *Save Vector Layer As* выберите *GeoPackage* в качестве *Format*.
4. Кликните на … для *File name* и назовите новый GeoPackage `vector_analysis`.
5. Изменить *Layer name* на `roads_34S`.
6. Измените *CRS* на *WGS 84 / UTM зону 34S*.
7. Кликните на *OK*:



   Это позволит создать новую базу данных GeoPackage и добавить слой `roads_34S`.

8. Повторите этот процесс для каждого слоя, создав новый слой в `vector_analysis.gpkg` файле GeoPackage с `_34S` как приложение к исходному названию и убрав каждый из старых слоев из проекта.

Примечание: Когда вы решите сохранить слой в существующий GeoPackage, QGIS **добавит** этот слой в GeoPackage.

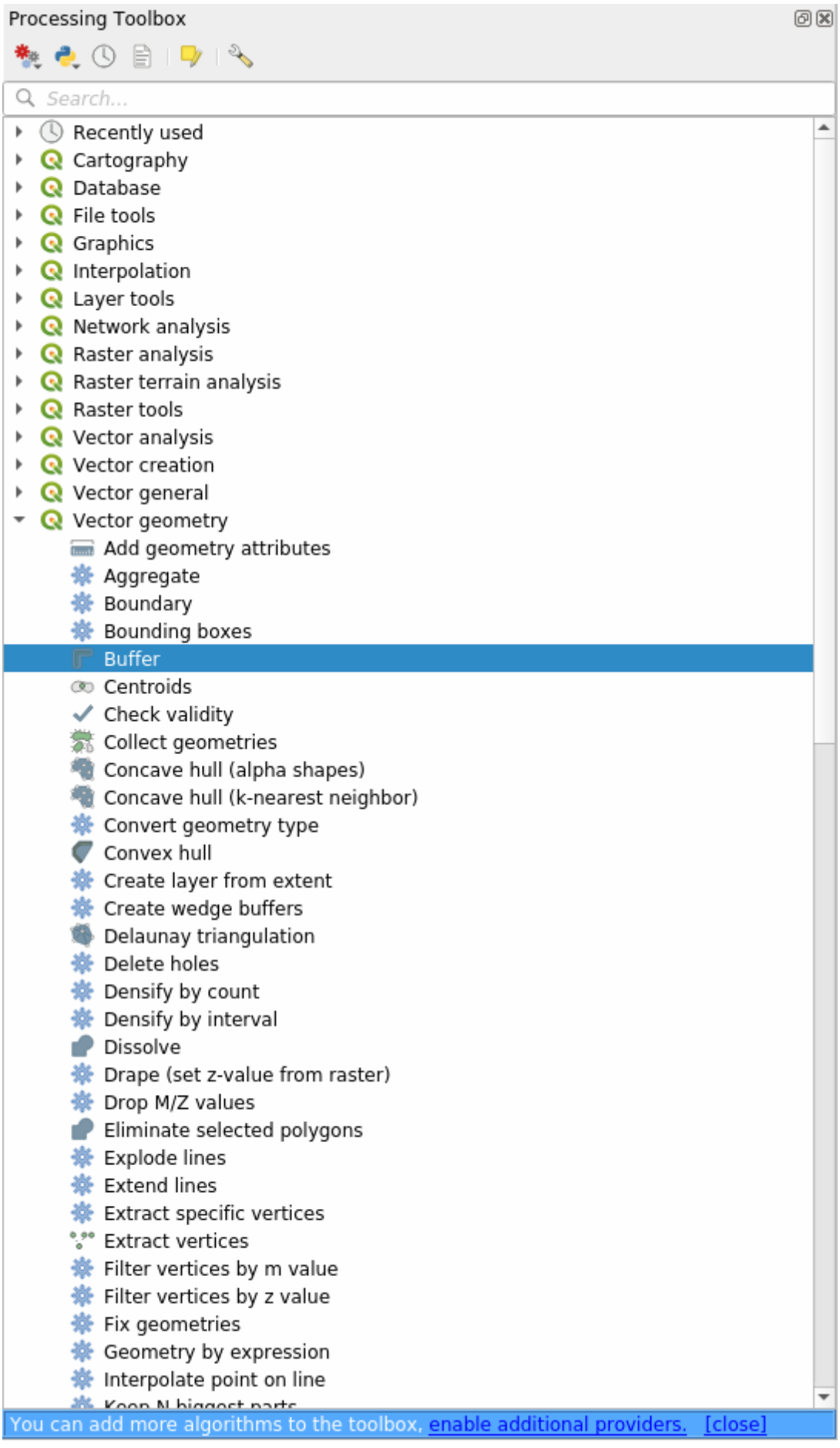
9. После того, как вы завершили процесс по всем слоям, вам надо кликнуть правой кнопкой мыши на любой слой и нажать *Zoom to layer extent*, чтобы сфокусировать карту на местность, которая вас интересует.

Теперь, когда мы преобразовали данные OSM в Универсальную поперечную проекцию Меркатора (УППМ), мы можем начать наши вычисления.

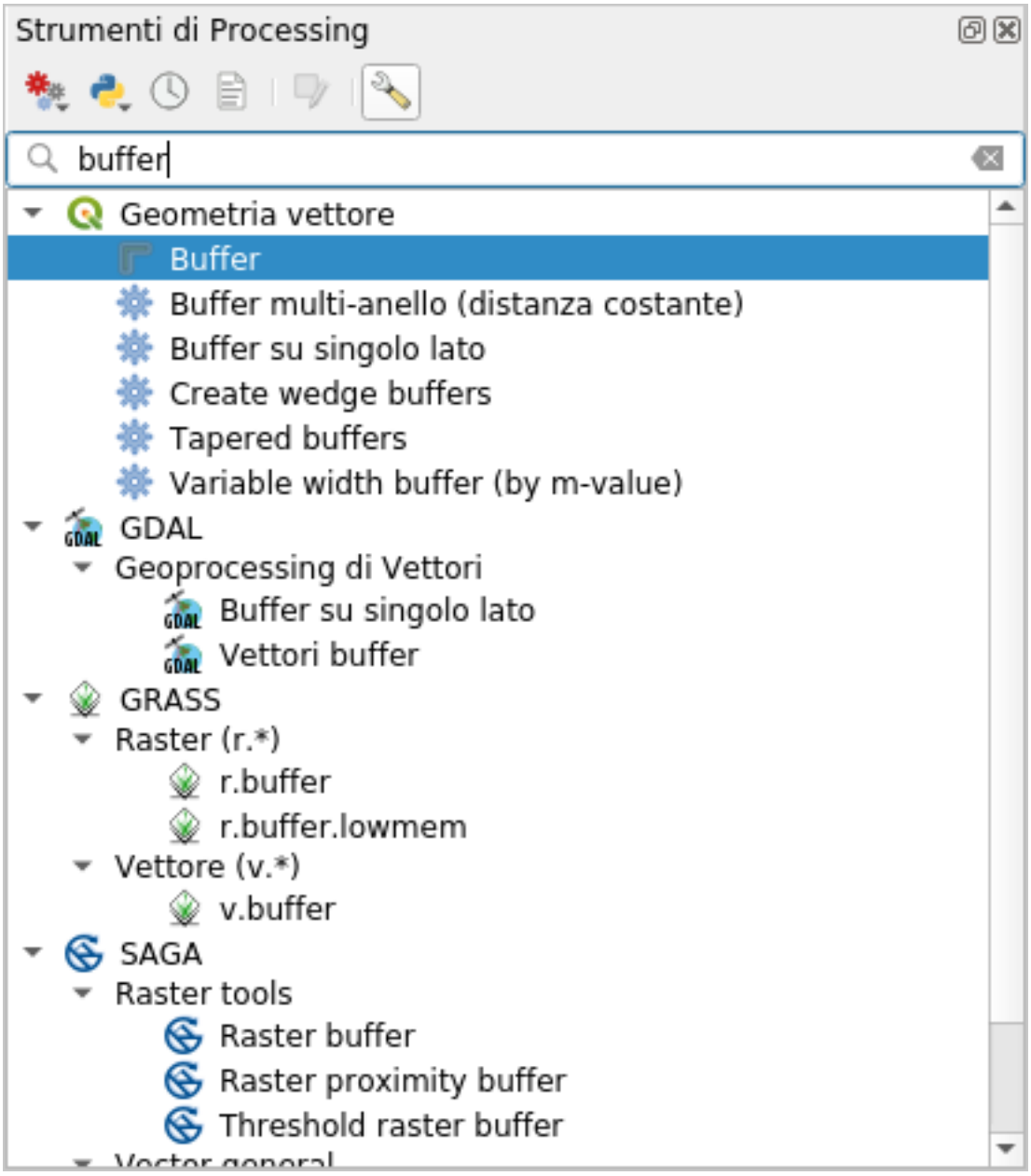
6.2.6  Идем дальше: Анализируем проблемы: расстояние от школ и дорог

QGIS позволяет рассчитывать расстояния между любыми векторными объектами.

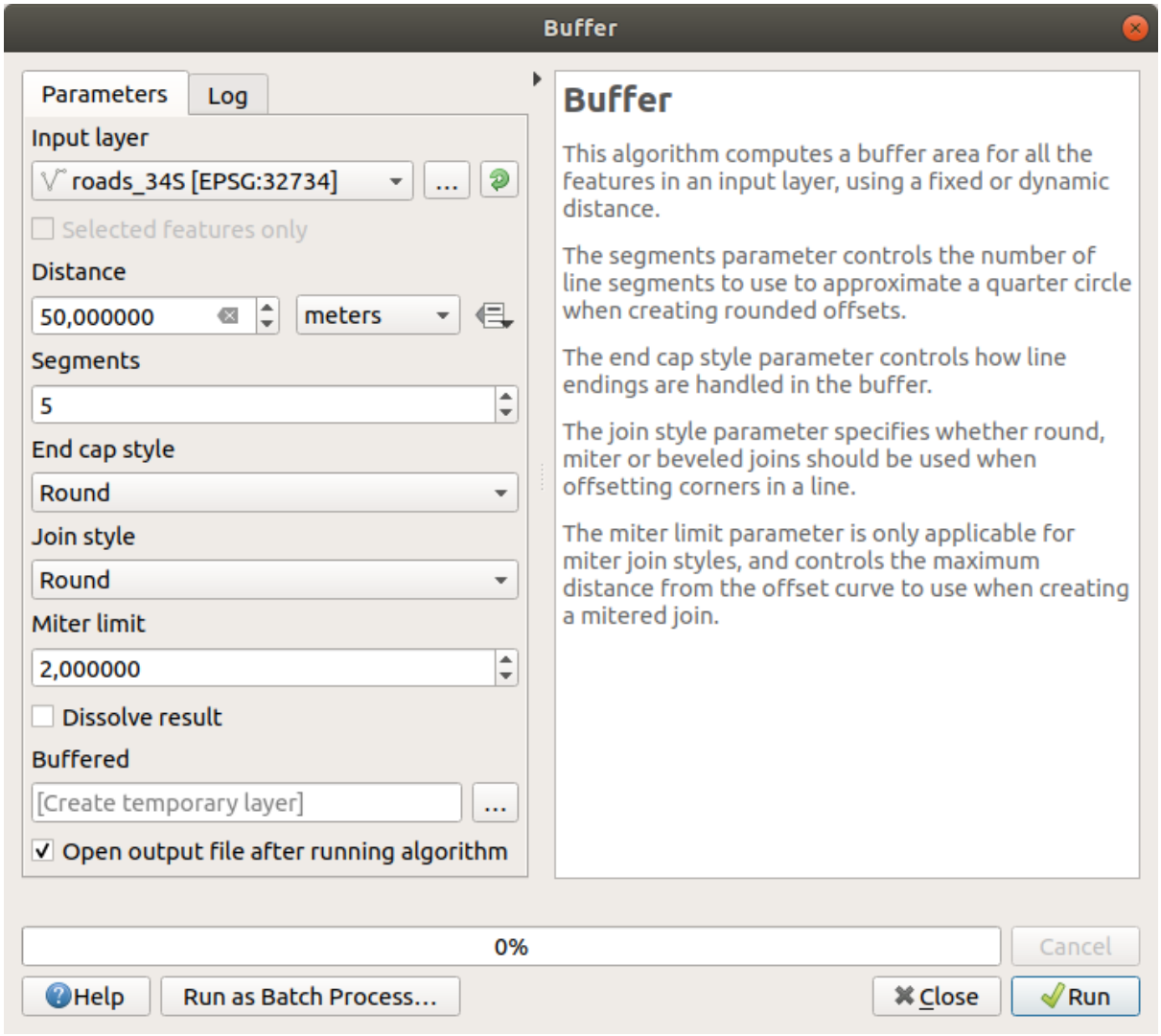
1. Удостоверьтесь в том, что только слои `roads_34S` и `buildings_34S` были видны (для упрощения карты во время работы).
2. Кликни на *Processing* ► *Toolbox* для того, чтобы открыть аналитическое **ядро** QGIS. В основном, в этом наборе инструментов имеются **все** алгоритмы (для векторного **и** растрового анализа).
3. Мы можем начать рассчитывать площади вокруг `roads_34S` используя алгоритм *Buffer*. Вы можете найти его в группе *Vector Geometry*.


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

Или вы можете набрать в меню поиска `buffer` в верхней части панели инструментов:



4. Два раза кликните по нему, чтобы открыть диалоговое окно алгоритма.
5. Выбираете `roads_34S` в качестве *Input layer*, указываете *Distance* до 50 и используйте значения по умолчанию для остальных параметров.



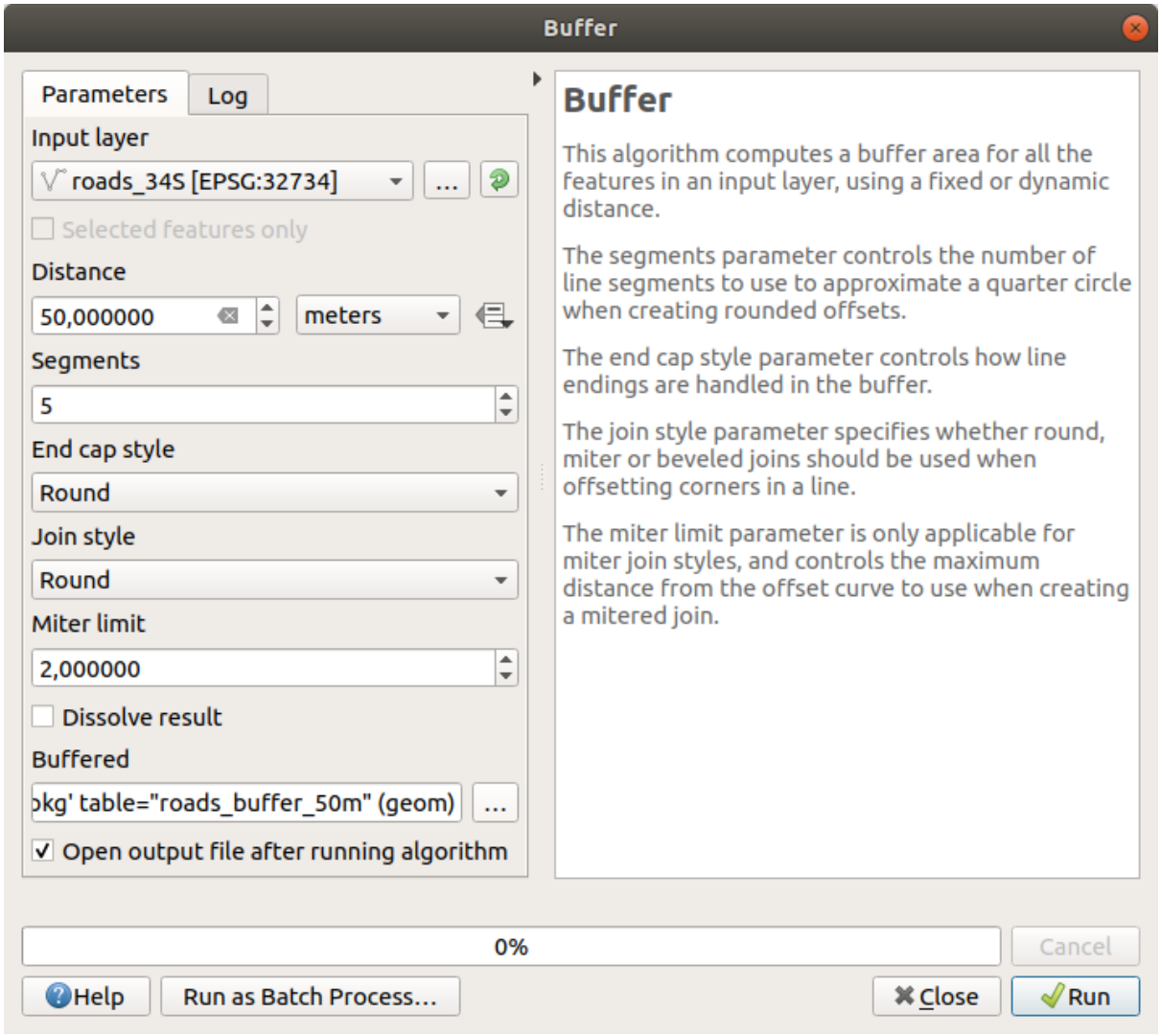
6. По умолчанию *Distance* измеряется в метрах, потому что наш входной набор данных находится в системе координат проекции, которая использует метр в качестве основной единицы измерения. Вы можете использовать поле со списком, чтобы выбрать другие проецируемые единицы, такие как километры, ярды и т.д.

Примечание: Если вы пытаетесь создать буфер на слое с географической системой координат, система при обработке предупредит вас и предложит перепроецировать слой в метрическую систему координат.

---

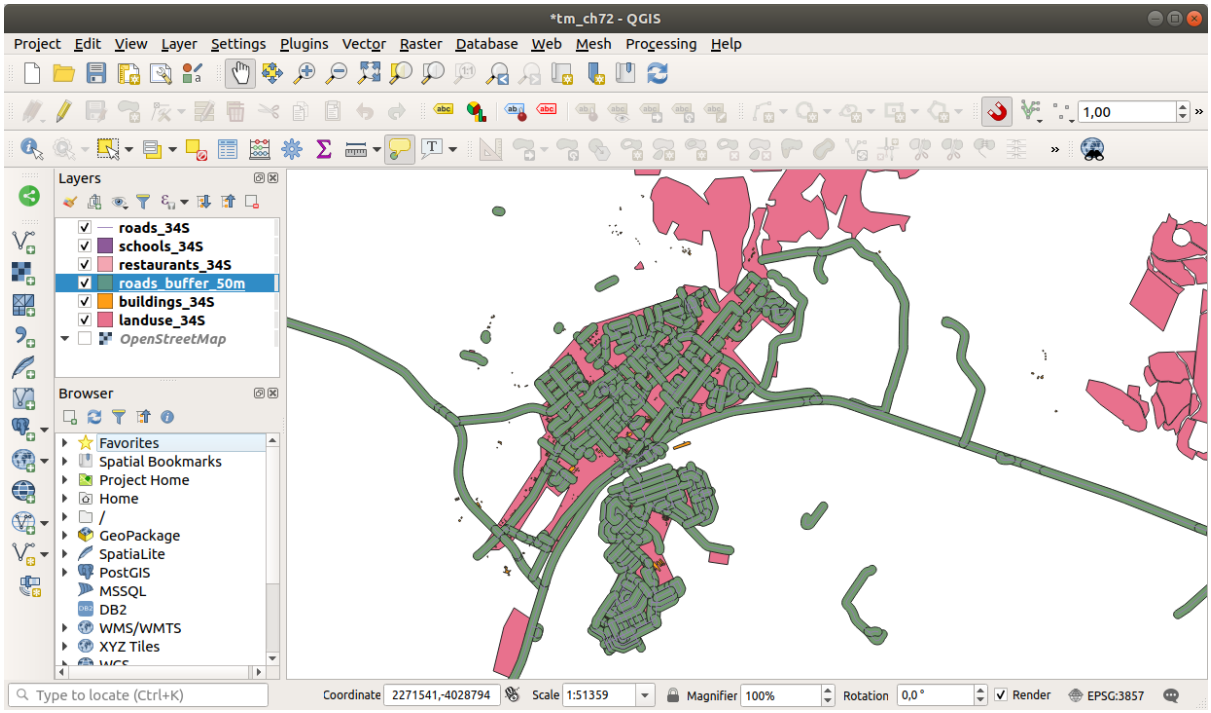
7. По умолчанию при обработке создаются временные слои и они добавляется в панель *Layers*. Вы также можете приложить результат в базу данных GeoPackage:

    1. Нажав на кнопку … и выбрав *Save to GeoPackage…*
    2. Назвав новый слой `roads_buffer_50m`.
    3. Сохранив его в файле `vector_analysis.gpkg`.



8. Кликните на *Run*, а затем закройте диалоговой окно *Buffer*.

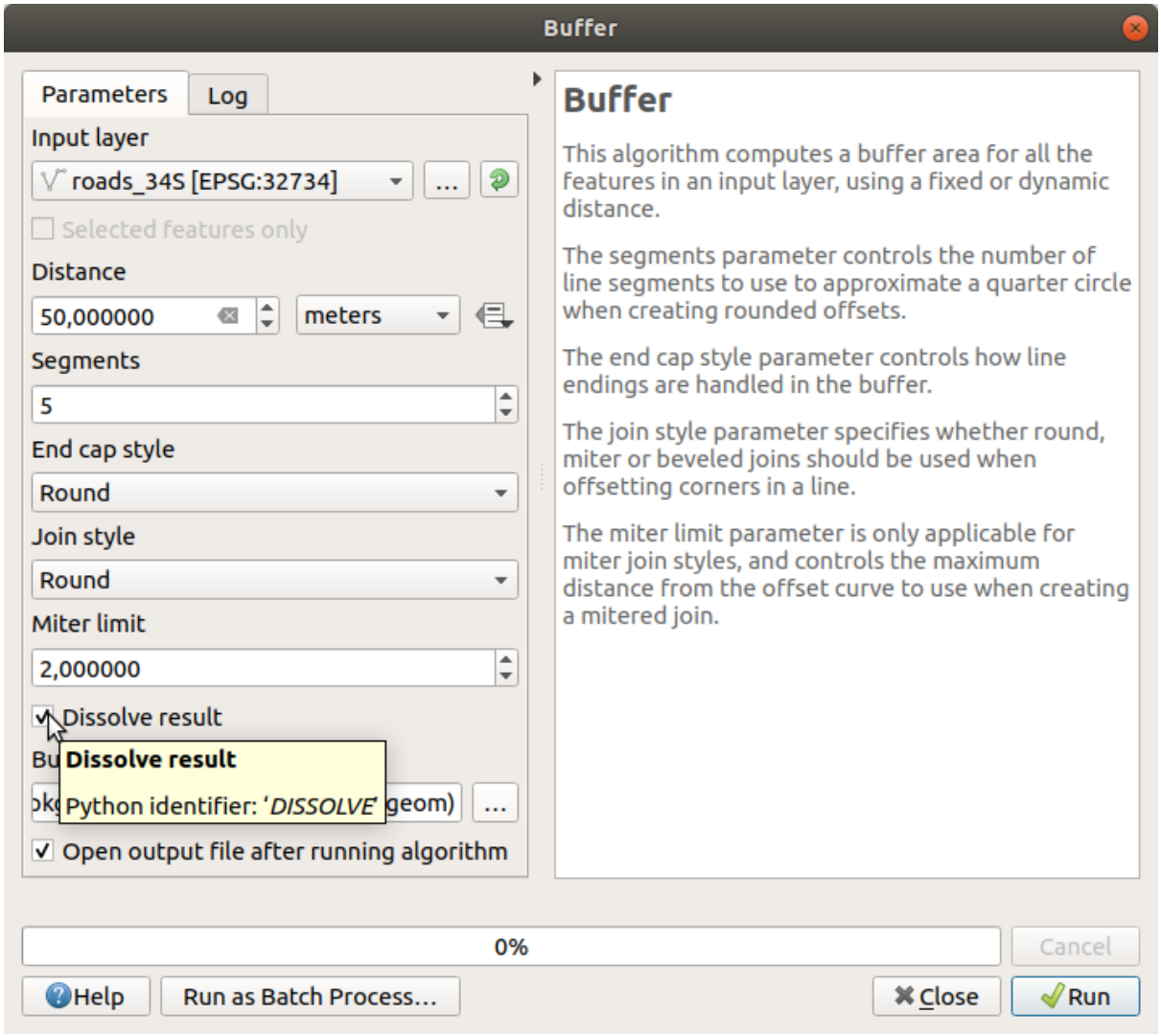
   Ваша карта теперь будет выглядеть примерно так:



Если ваш новый слой находится в верхней части списка *Layers*, он, скорее всего, закроет большую часть вашей карты, но вы сможете увидеть все местности в вашем регионе, которые находятся на расстоянии в 50 м от дороги.

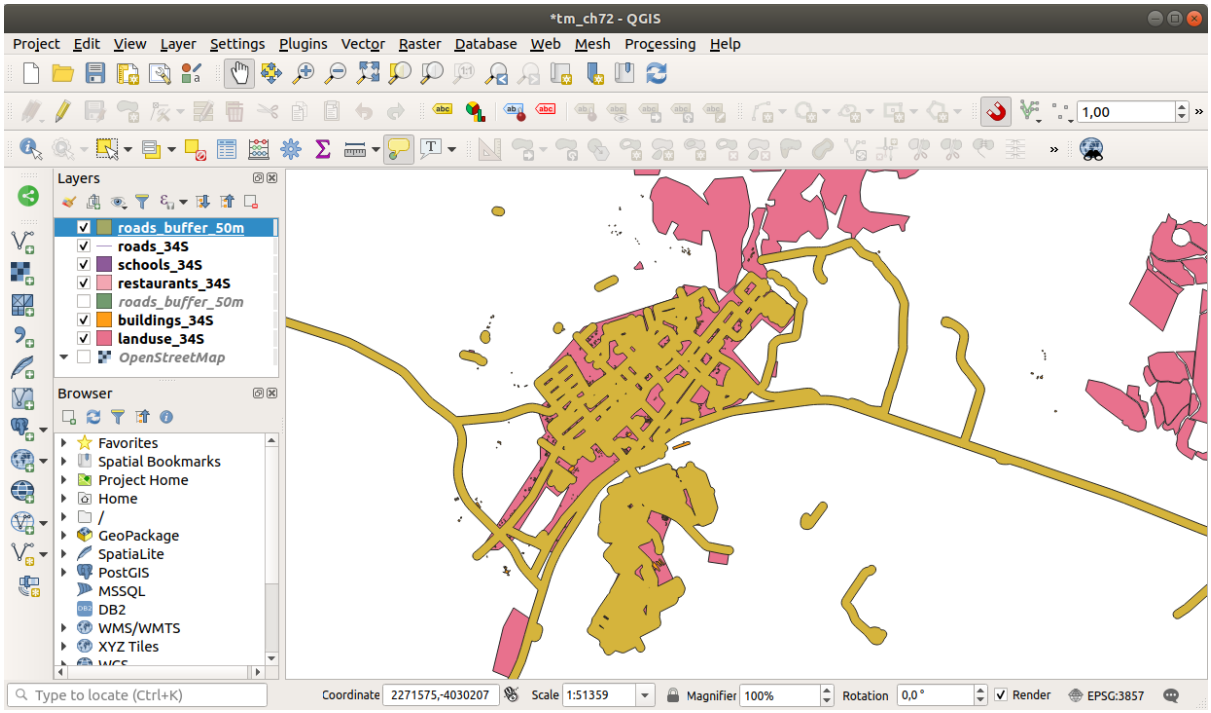
Обратите внимание, что в вашем буфере есть отдельные местности, соответствующие каждой отдельной дороге. Чтобы устранить эту проблему необходимо:

1. Убрать надпись *roads_buffer_50m* в слое и снова создать буфер с включённой *Dissolve results* командой.



2. Сохраните выход как *roads_buffer_50m_dissolved*.
3. Кликните на *Run* и закройте диалоговой окно *Buffer*.

После того, как вы добавили слой в панель *Layers*, она будет выглядеть так:



Теперь лишних делений уже нет.

Примечание: *Краткая справка (помощь)* в правой части диалогового окна объясняет, как работает алгоритм. Если вам нужна дополнительная информация, просто кликните на кнопку *Help* в нижней части, чтобы открыть более подробное руководство по алгоритму.

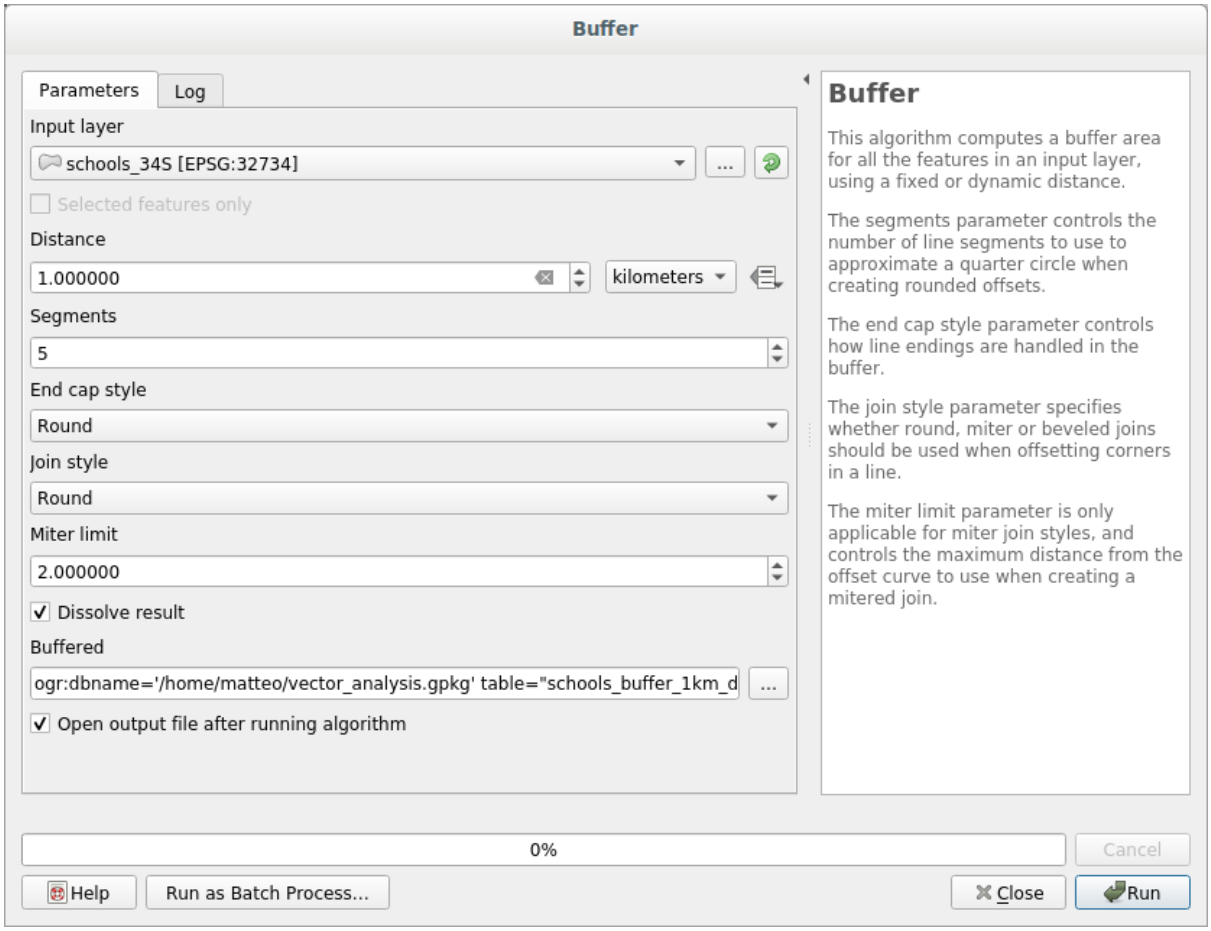
6.2.7  Попробуйте сами: Расстояние от школ

Следуйте тому же подходу, который описан выше, и создайте буфер для ваших школ.

It shall be `1 km` in radius. Save the new layer in the `vector_analysis.gpkg` file as `schools_buffer_1km_dissolved`.

Answer

- Your buffer dialog should look like this:

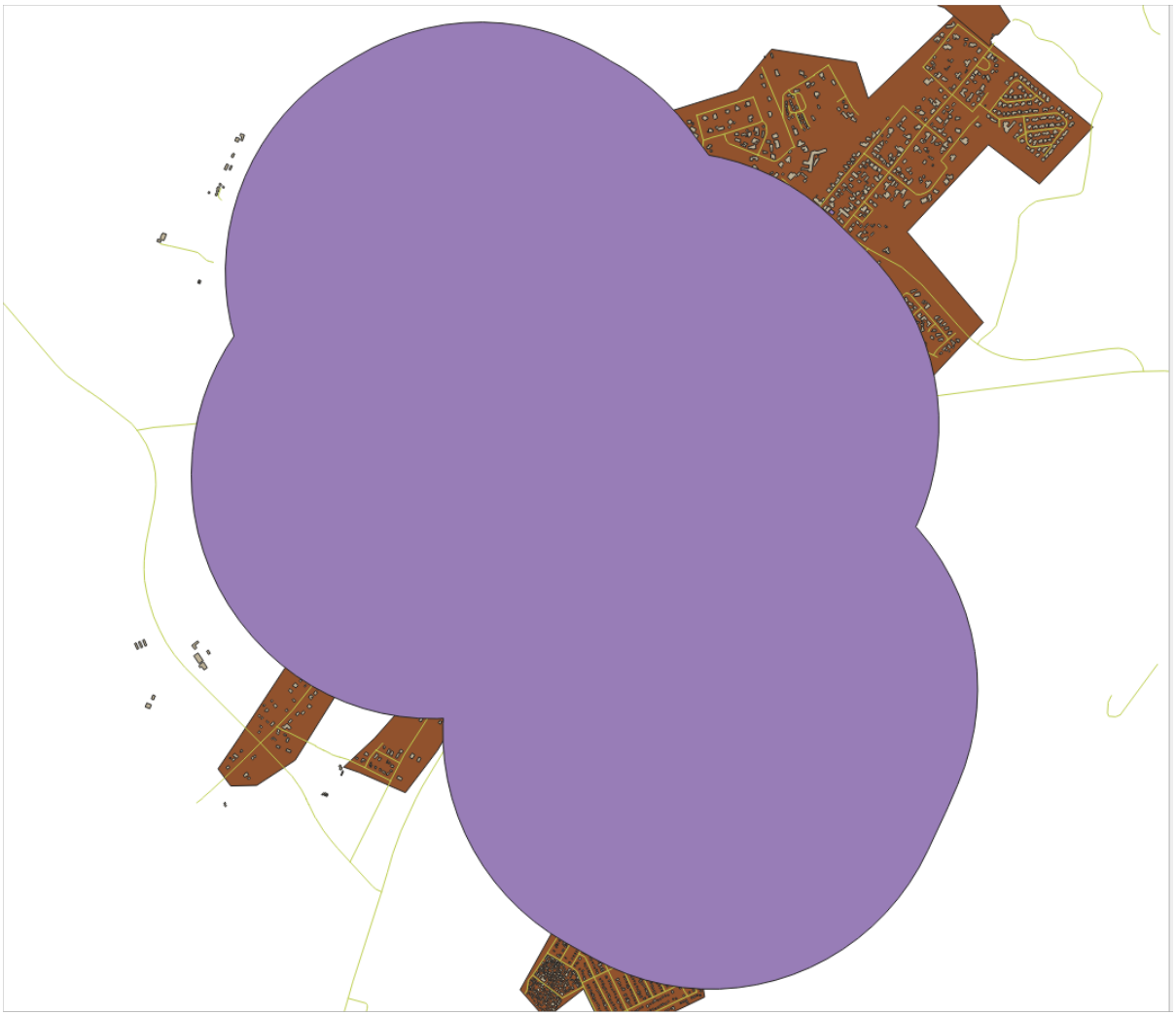


The *Buffer distance* is *1* kilometer.

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



To this:

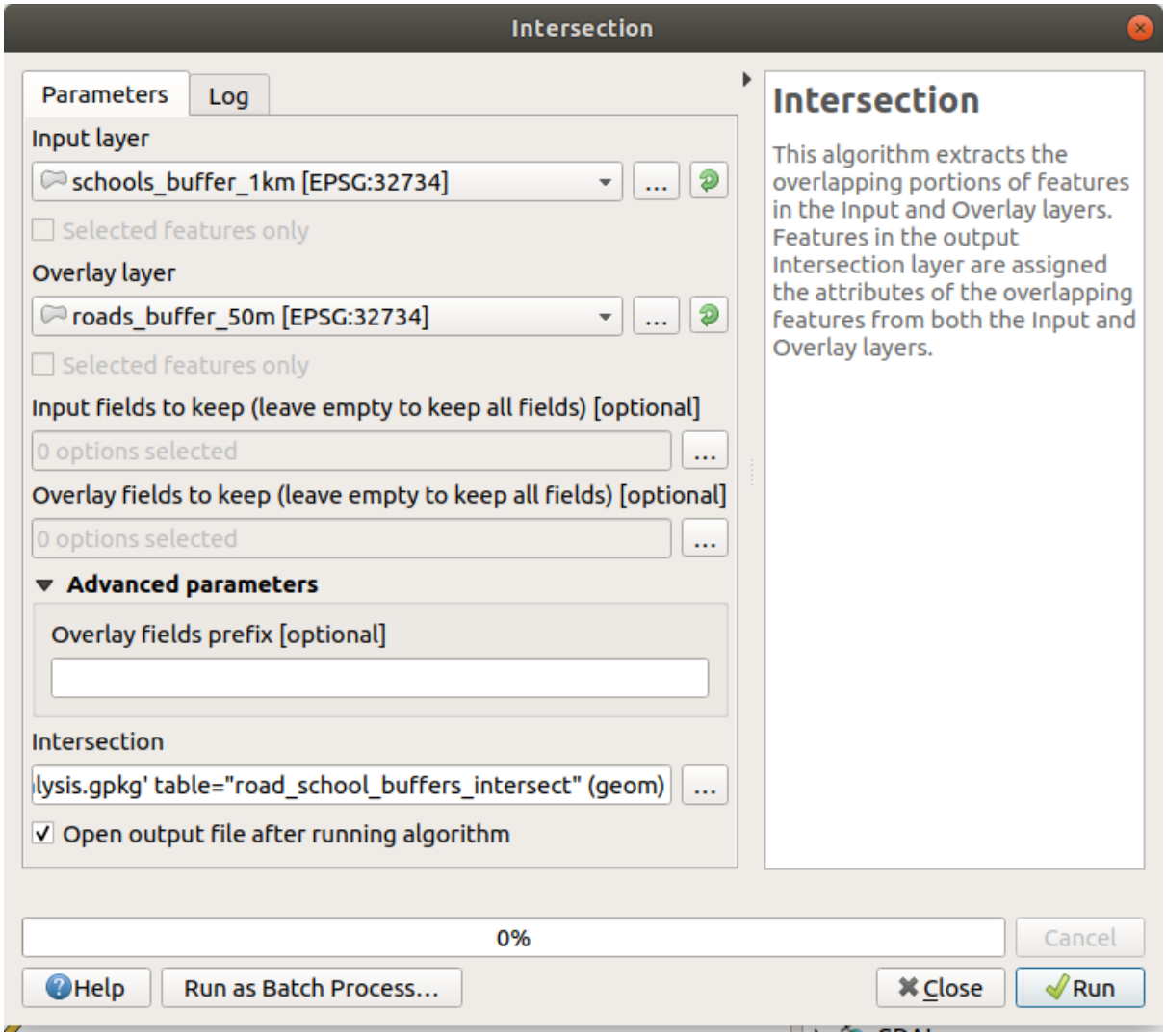


The first image shows the buffer with the *Segments to approximate* value set to *5* and the second shows the value set to *20*. In our example, the difference is subtle, but you can see that the buffer's edges are smoother with the higher value.

6.2.8 Идем дальше: Зоны перекрытия

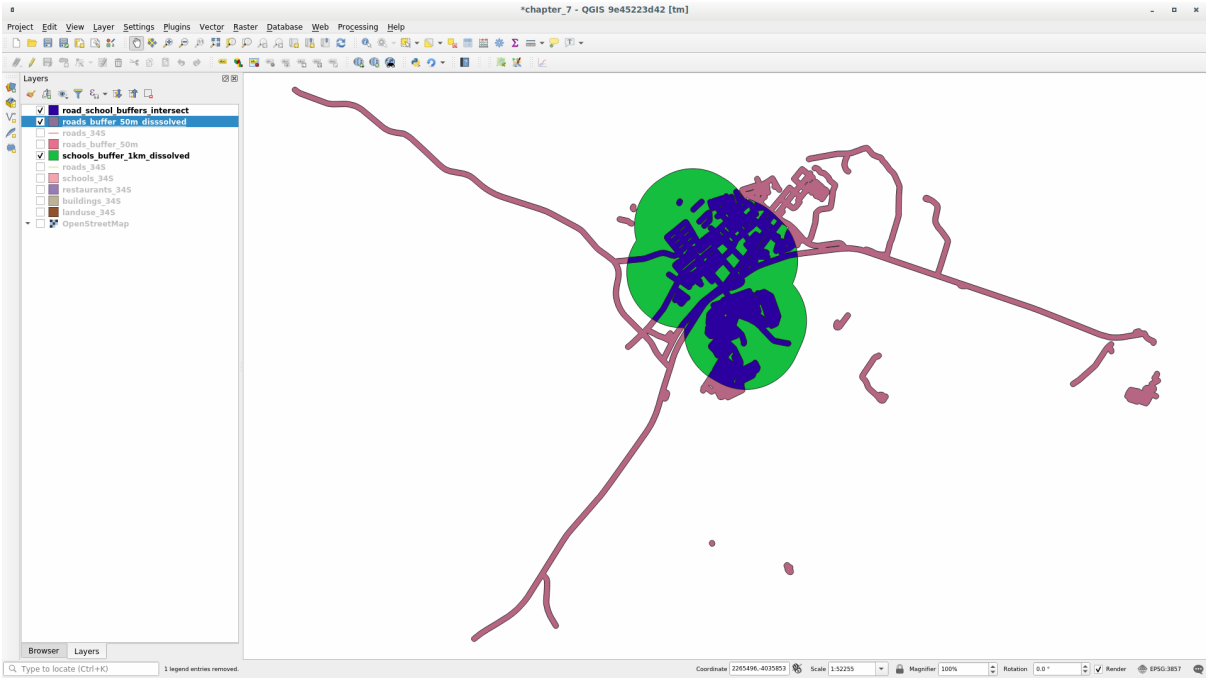
Теперь мы определили местности, где дорога находится на расстоянии менее чем в 50 метрах, а также местности, расположенные от школы в пределах 1 км (по прямой линии, а не по дороге). Естественно нам нужны только те местности, которые отвечают обоим этим критериям. Для этого нам нужно будет использовать инструмент *Intersect*. Вы сможете найти его в группе *Vector Overlay* в *Processing Toolbox*.

1. Используйте два буферных слоя как *Input layer* и *Overlay layer* выберите `vector_analysis.gpkg` GeoPackage в *Intersection* с *Layer name* `road_school_buffers_intersect`. Остальное оставьте как предлагается (по умолчанию).

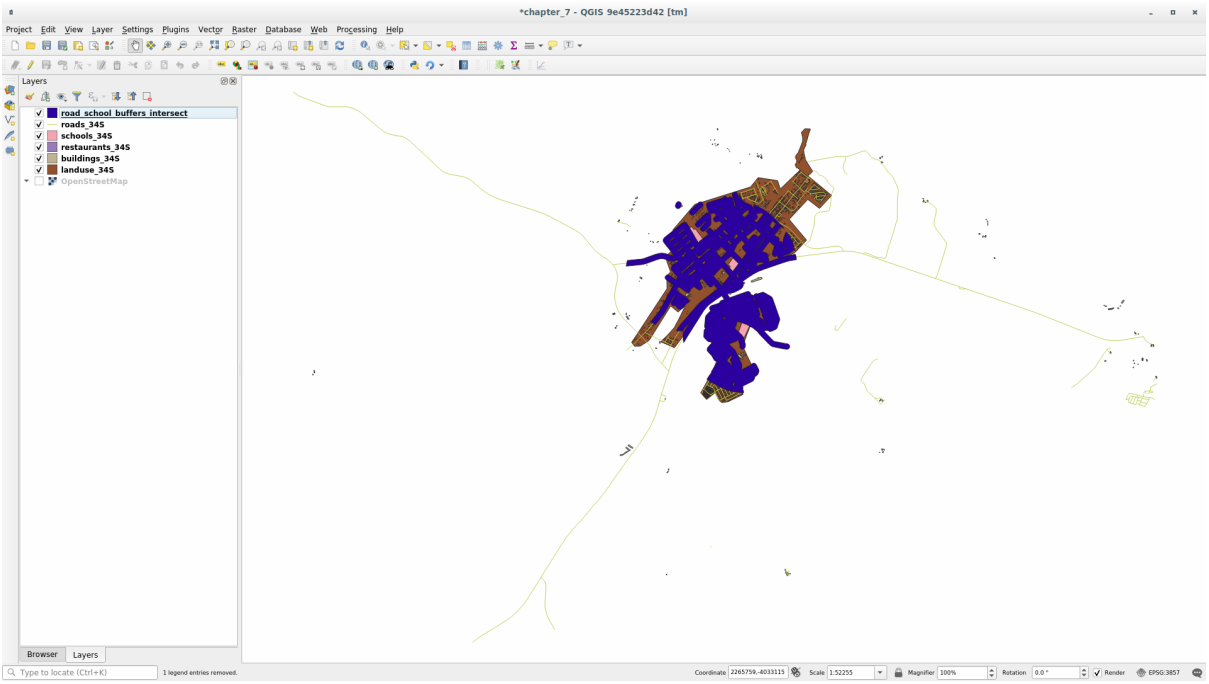


2. Кликните на кнопку *Run*.

   На изображении ниже местности, окрашенные синим цветом это те местности, где оба соответствия по критериям расстояния соблюдены.



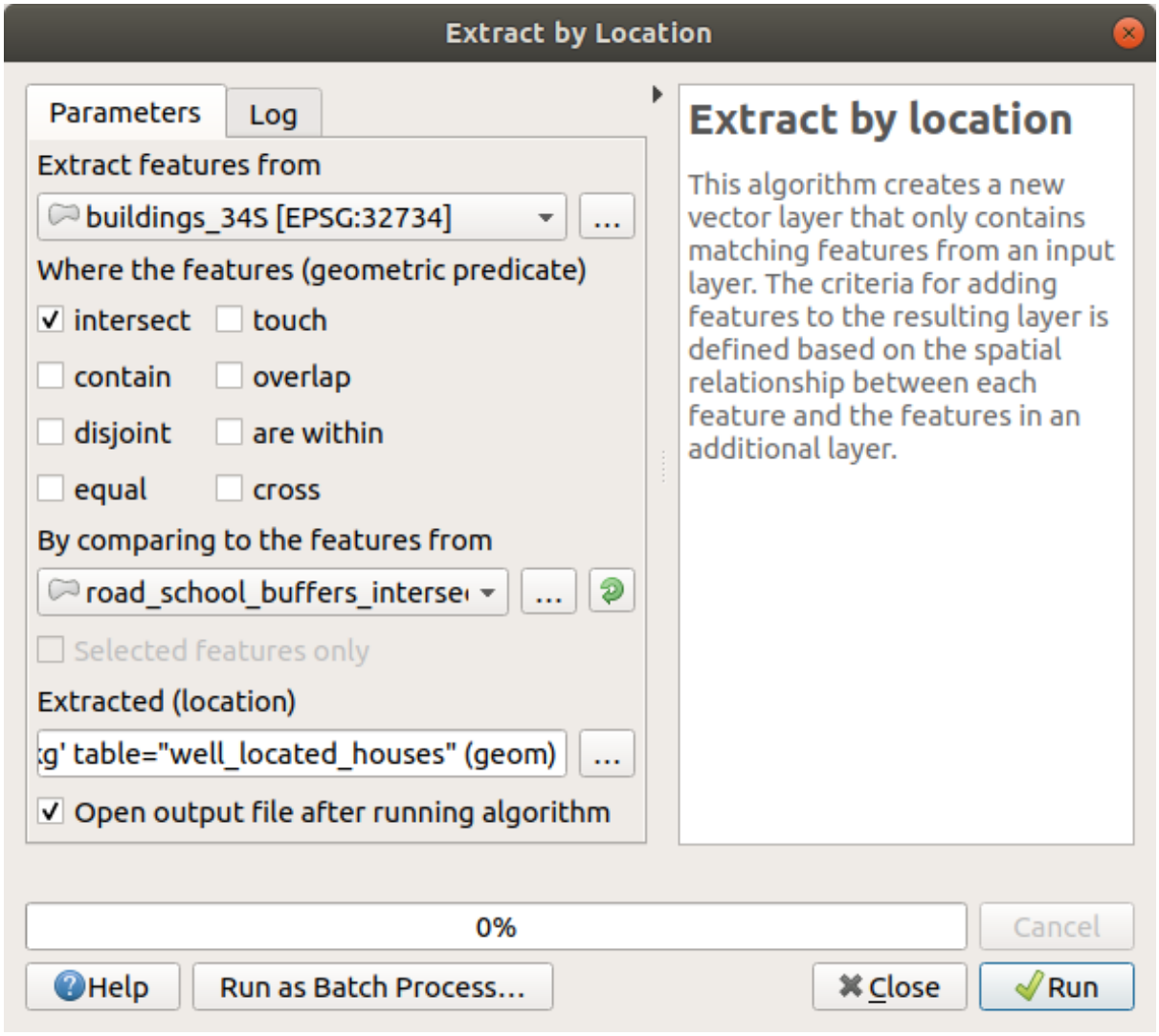
3. Вы можете убрать два буферных слоя и оставить только тот, который показывает, где они перекрываются, так как мы действительно хотели это выяснить в первую очередь:



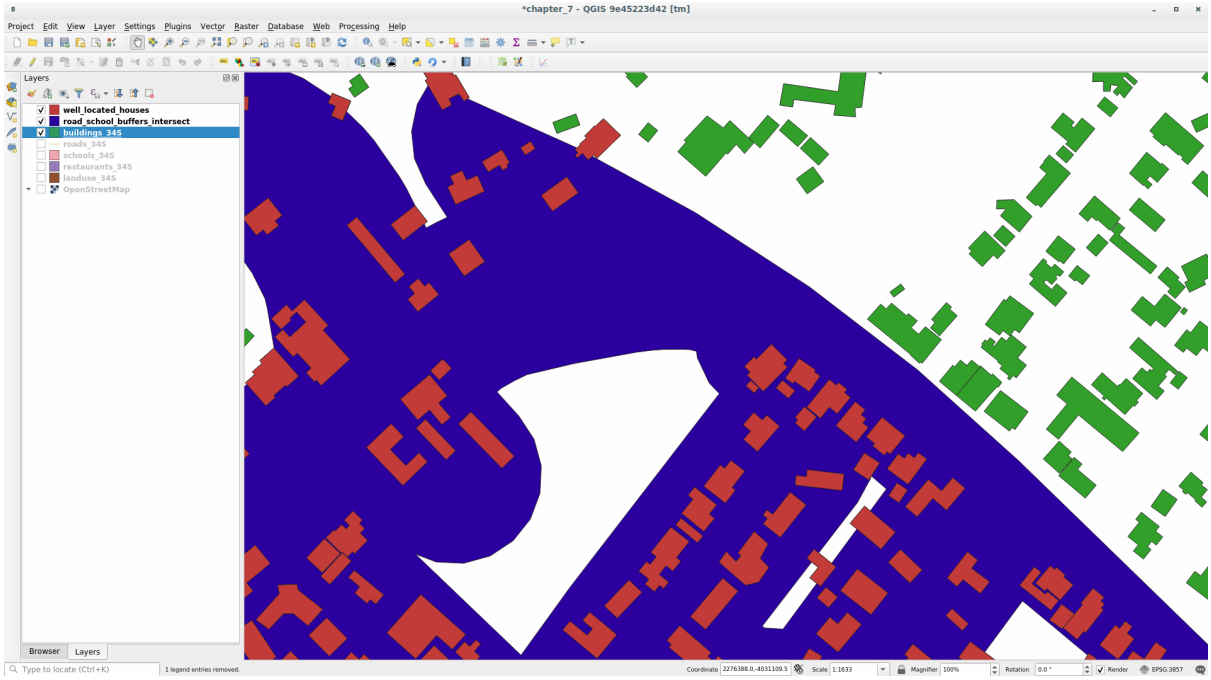
6.2.9 Идем дальше: Извлечение зданий

Теперь у вас есть местность, на которой здания должны перекрывать друг друга. Теперь вам надо извлечь здания в этой местности.

1. Ищите пункт в меню *Vector Selection* ► *Extract by location* в *Панели инструментов обработки*.
2. Выберите `buildings_34S` в *Extract features from*. Проверьте *intersect* в *Where the features (geometric predicate)*, выберите буферный слой пересечения в *By comparing to the features from*. Сохраните в `vector_analysis.gpkg` и назовите слой `well_located_houses`.



3. Кликните на кнопку *Run* и закройте диалоговое окно.
4. Вы, скорее всего, обнаружите, что мало что изменилось. Если так, переместите слой *well_located_houses* в верхнюю часть списка слоев, затем увеличьте масштаб.



Здания красным цветом отвечают нашим критериям, а зеленым - нет.

5. Теперь у вас два отдельных слоя и вы сможете убрать `buildings_34S` из списка слоев.

6.2.10  Попробуйте сами: Дальше отфильтровать наши здания

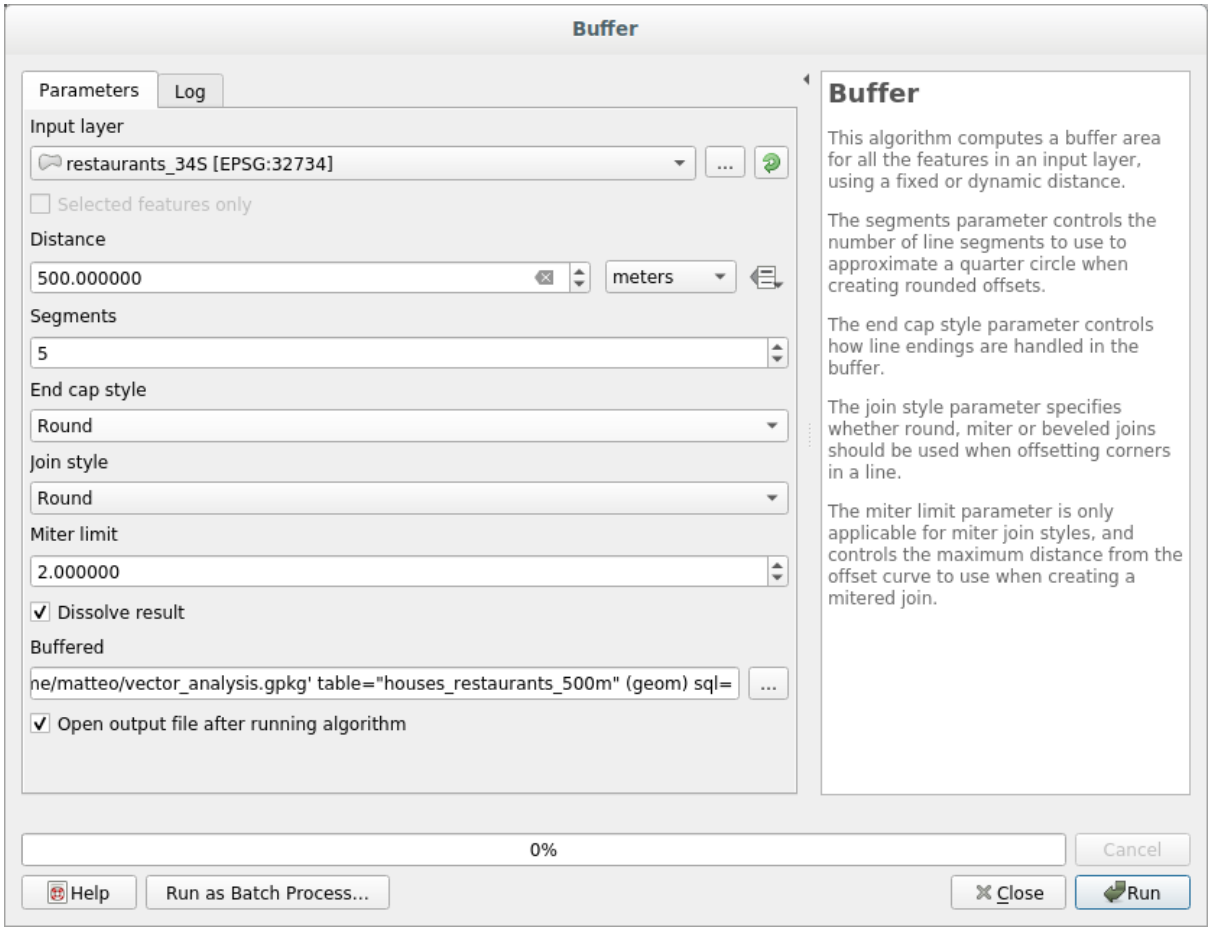
Теперь у нас есть слой, который показывает нам все здания в пределах 1 км от школы и в пределах 50 м от дороги. Нам теперь надо сократить этот отбор и показать только те здания, которые находятся в пределах 500 м от ресторана.

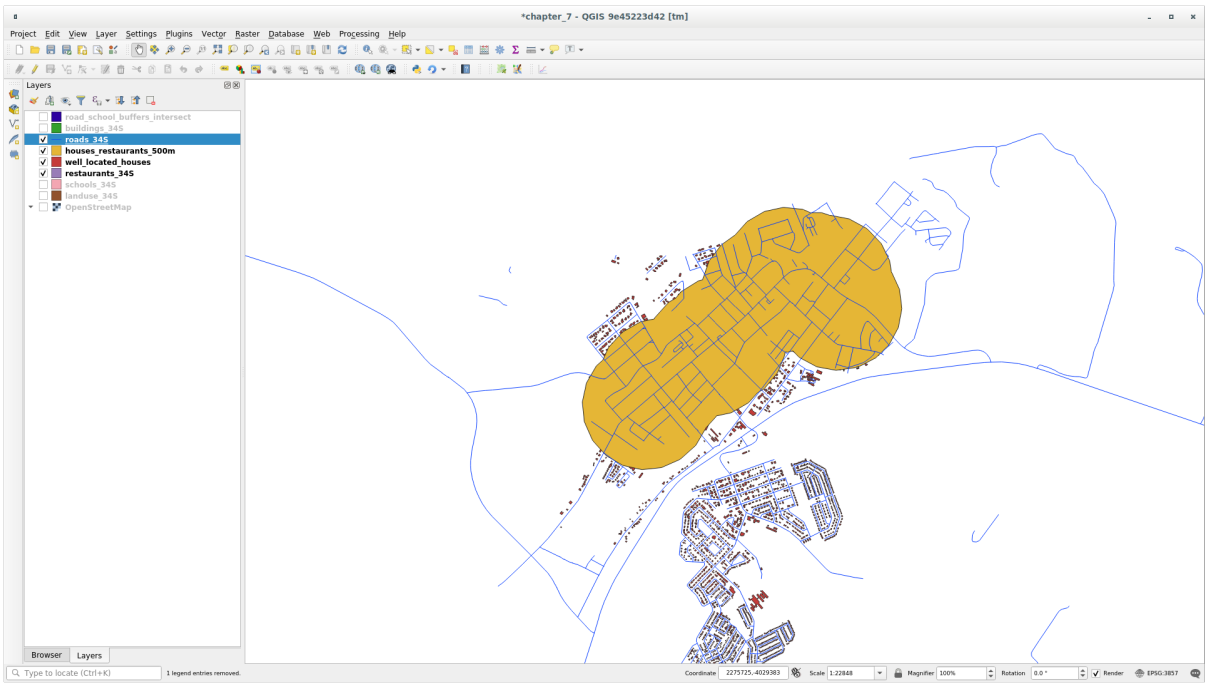
Следуя процессам, описанным выше, вам надо создать новый слой, который будет называться *houses_restaurants_500m* и который дополнительно профильтрует ваш слой *well_located_houses* для того, чтобы показать только те, которые находятся в пределах 500 м от ресторана.

Answer

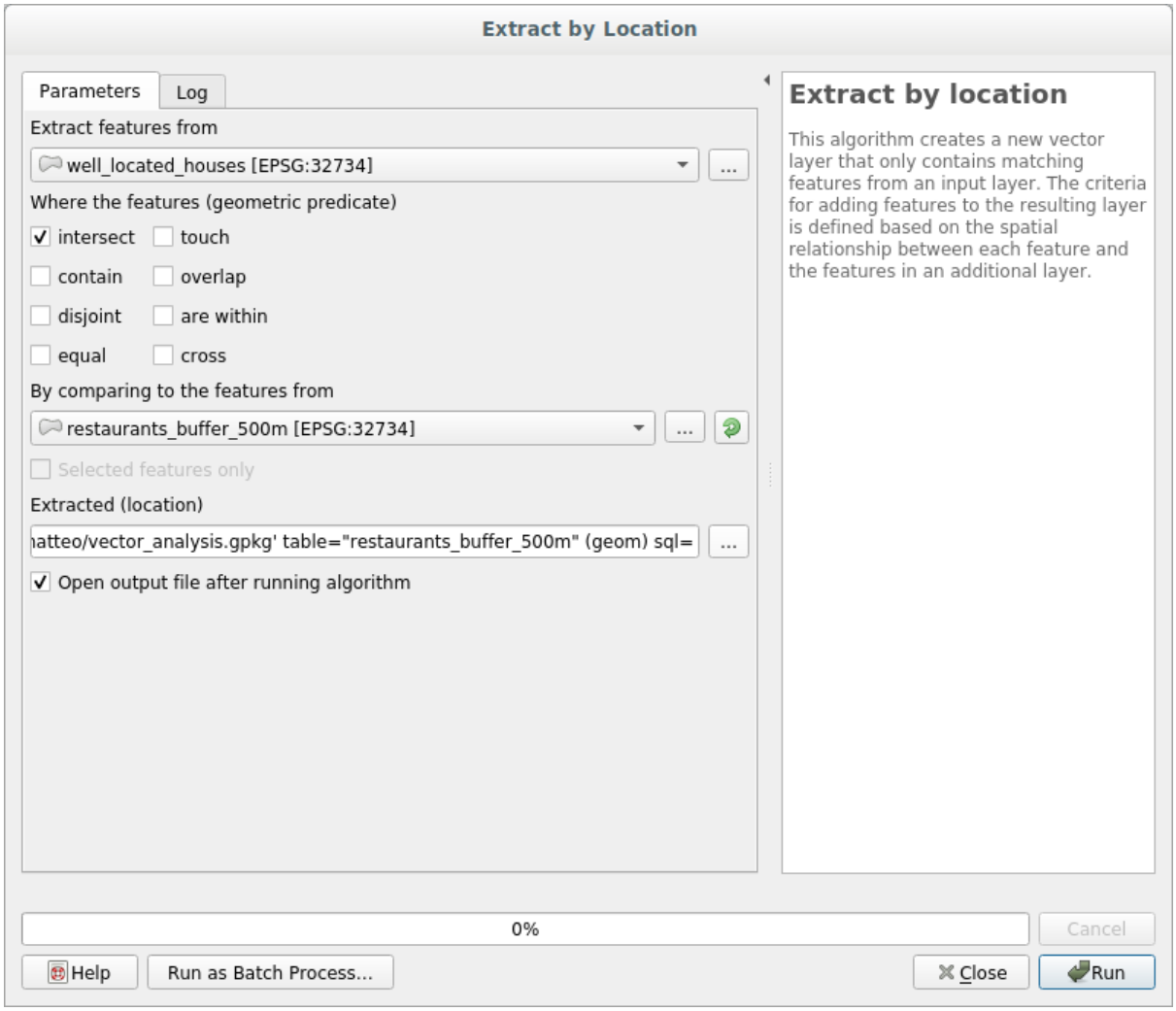
To create the new *houses_restaurants_500m* layer, we go through a two step process:

1. First, create a buffer of 500m around the restaurants and add the layer to the map:

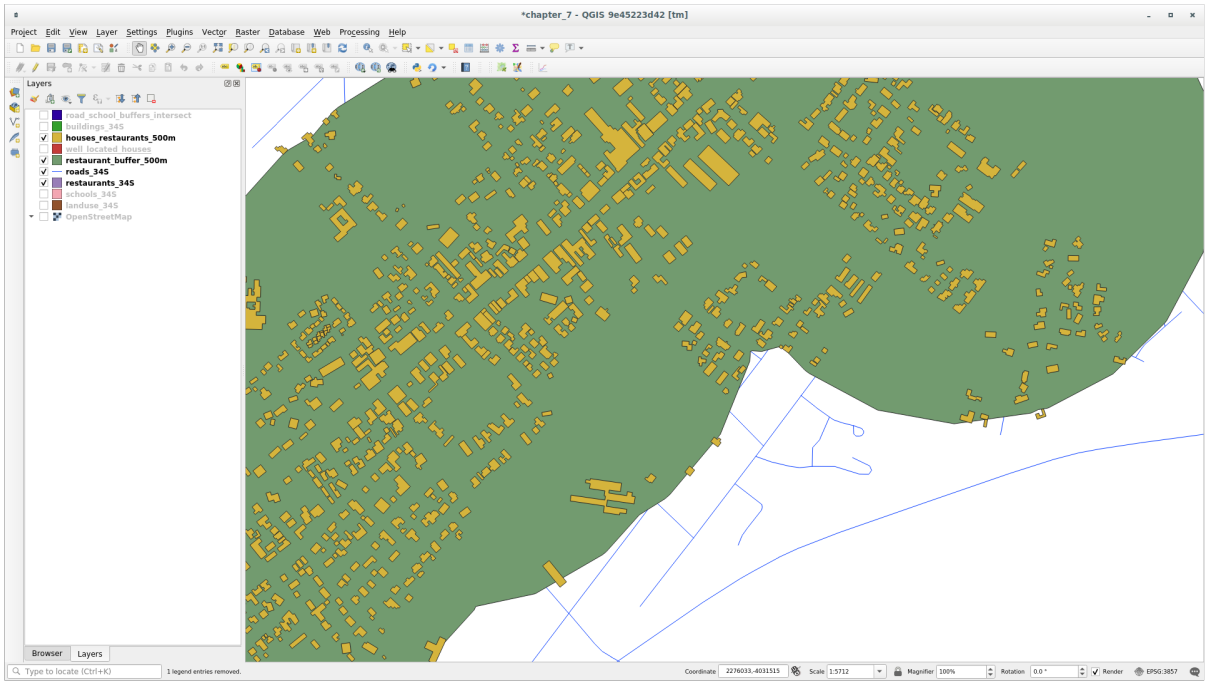




2. Next, extract buildings within that buffer area:



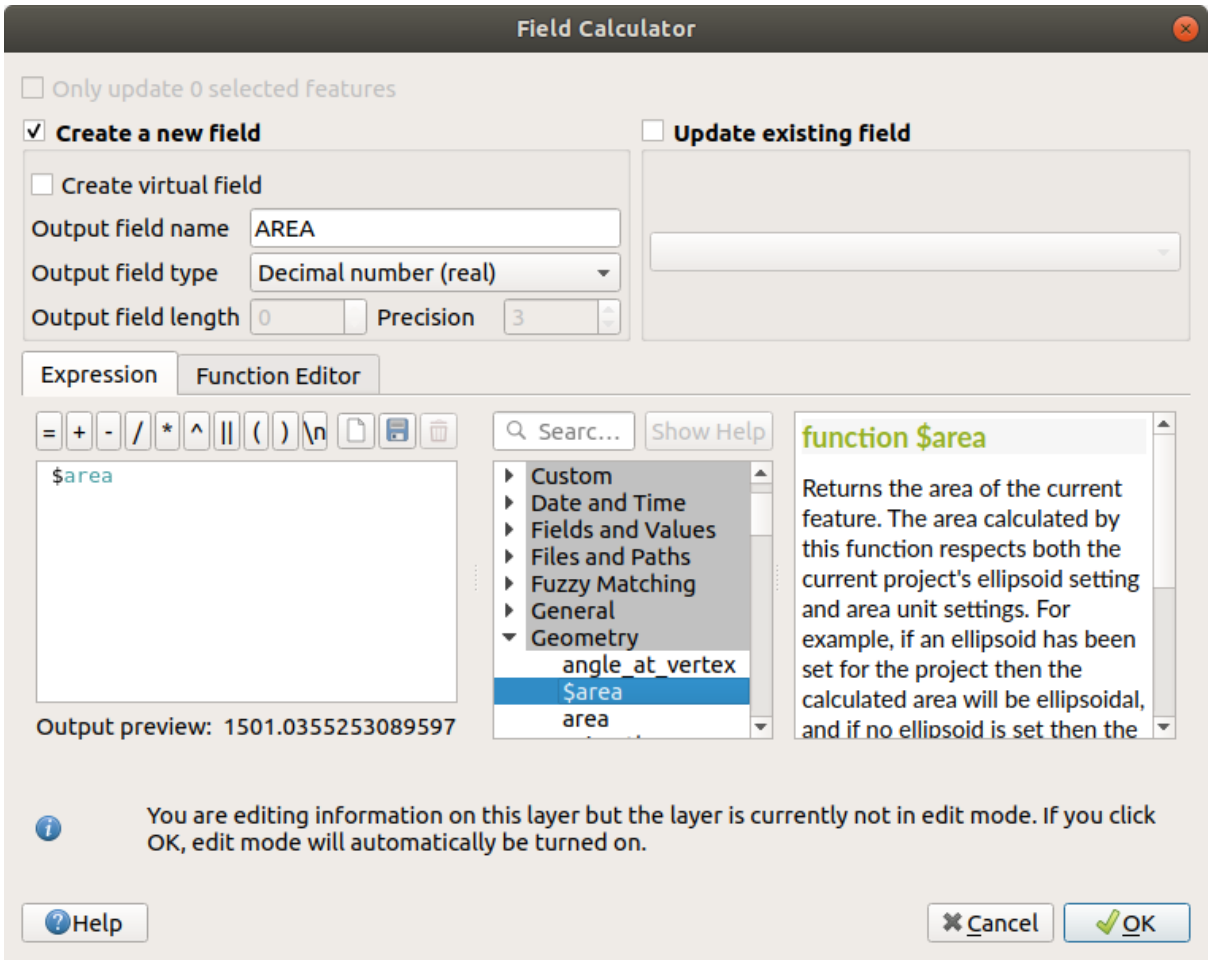
Your map should now show only those buildings which are within 50m of a road, 1km of a school and 500m of a restaurant:



6.2.11 Идем дальше: Выбираем здания подходящего размера

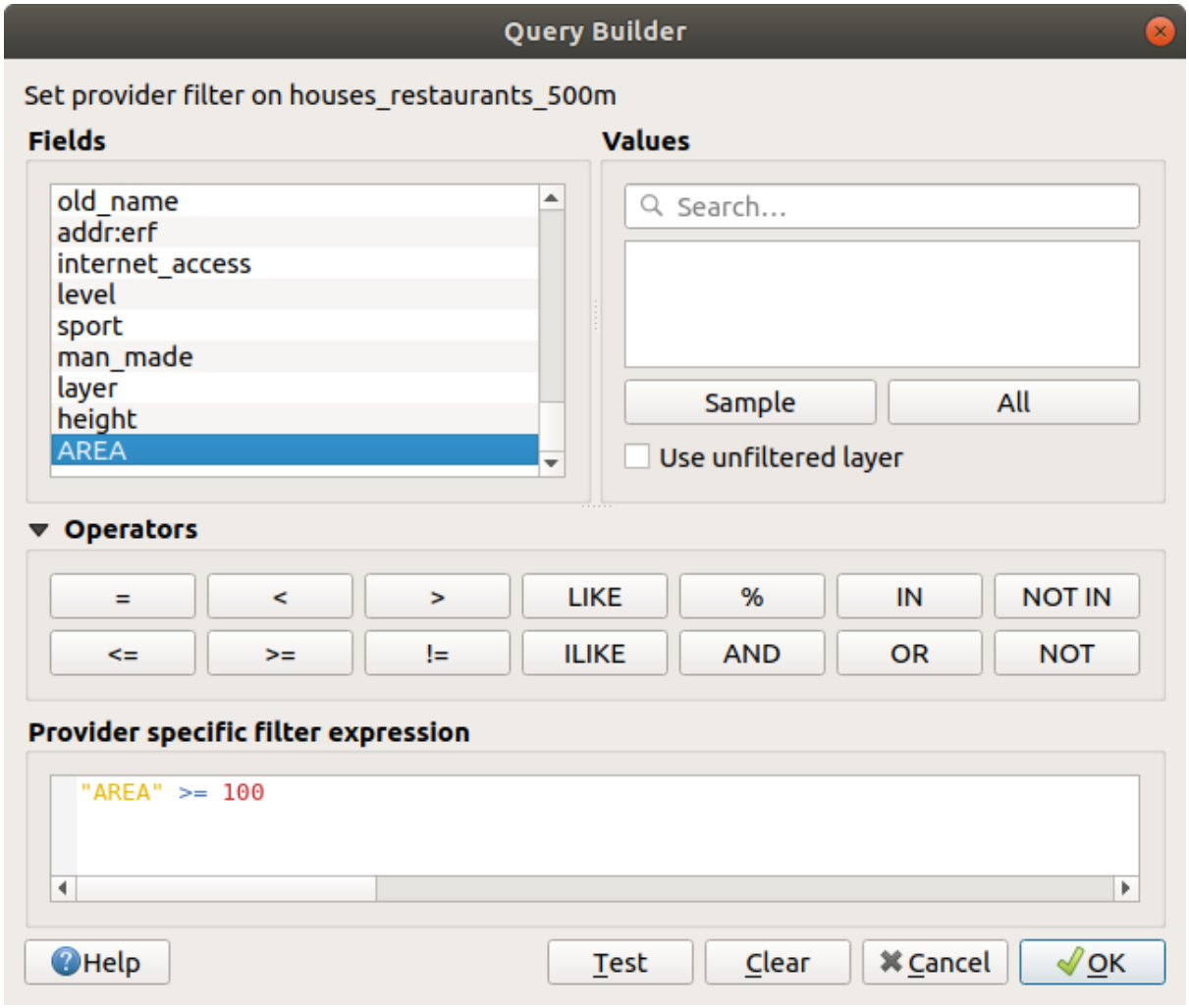
Чтобы увидеть, какие здания имеют правильный размер (более 100 квадратных метров), нам надо рассчитать их размер.

1. Выберите слой *houses_restaurants_500m* и откройте *Калькулятор поля*, нажав на кнопку Open Field Calculator на главной панели инструментов или в окне таблицы.
2. Выберите *Create a new field*, установите *Output field name* на `AREA`, выберите *Decimal number (real)* как *Output field type*, и выберите `$area` из группы *Geometry*.



   В новом поле `AREA` будет площадь каждого здания в квадратных метрах.

3. Кликните на *OK*. Поле `AREA` добавлено в конце таблицы атрибутов.
4. Кликните мышкой на кнопку Toggle Editing для того, чтобы завершить редактирование и сохраните ваши изменения при появлении запроса.
5. Во вкладке *Source* в свойствах слоя установите *Provider Feature Filter* на `"AREA >= 100`.



6. Кликните на кнопку *OK*.

Ваша карта теперь должна показывать только те здания, которые соответствуют нашим первоначальным критериям и имеют размер более 100 квадратных метров.

6.2.12  Попробуйте сами

Сохраните ваш вариант как новый слой, следуя подходу, которому вы научились выше. Файл должен быть сохранен внутри той же базы данных GeoPackage с названием `solution`.

6.2.13 В заключении

Следуя подходу решения проблем GIS вместе с инструментами векторного анализа QGIS, вы смогли быстро и легко решить проблему с несколькими критериями.

6.2.14 Что дальше?

На следующем уроке мы изучим, как рассчитать наикратчайшее расстояние по дорогам от одной точки до другой.

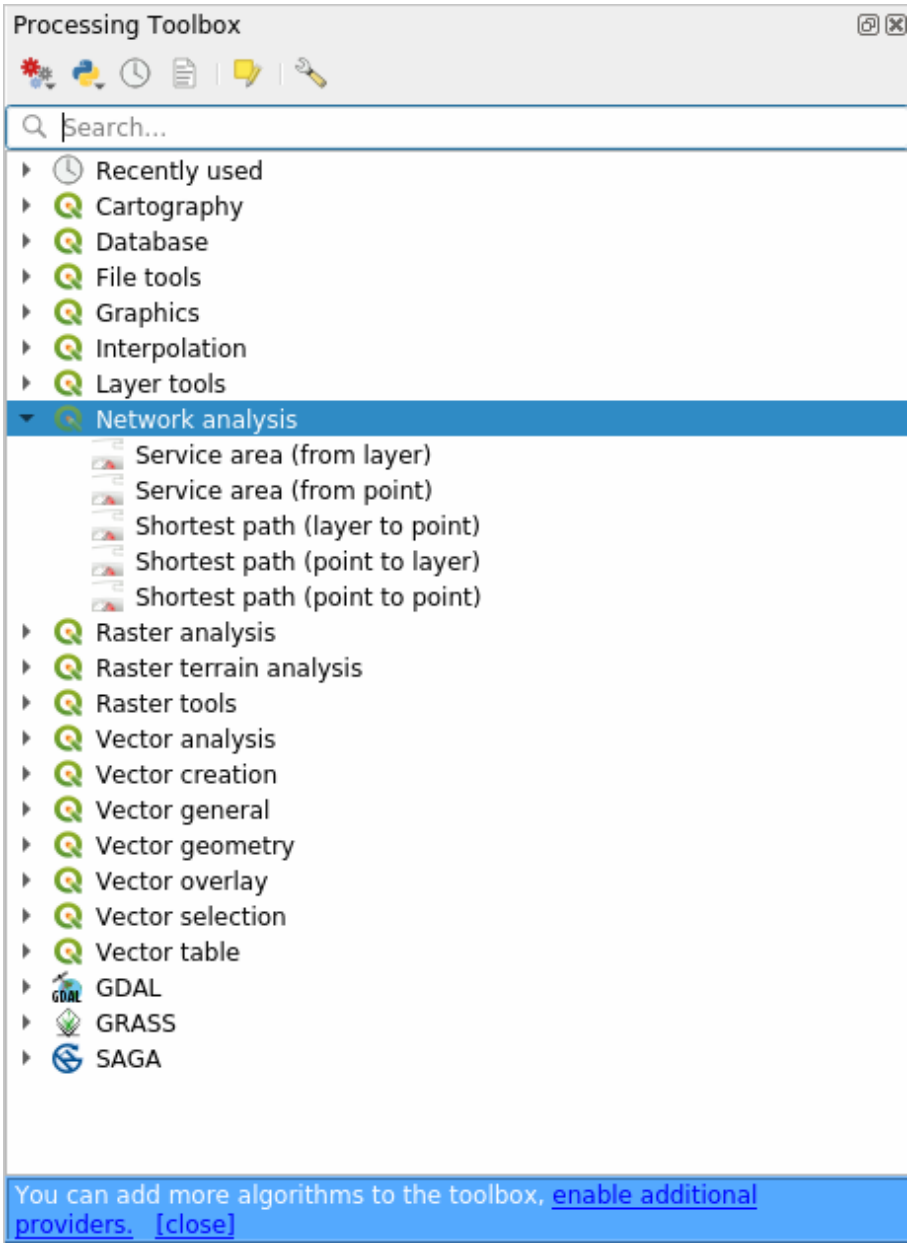
6.3 Занятие: Сетевой анализ

Расчет кратчайшего расстояния между двумя точками это обычная задача GIS. Инструменты для решения этой задачи можно найти в *Processing Toolbox*.

Цель этого урока: научиться использовать алгоритмы *Network analysis*.

6.3.1  Идем дальше: Инструменты и данные

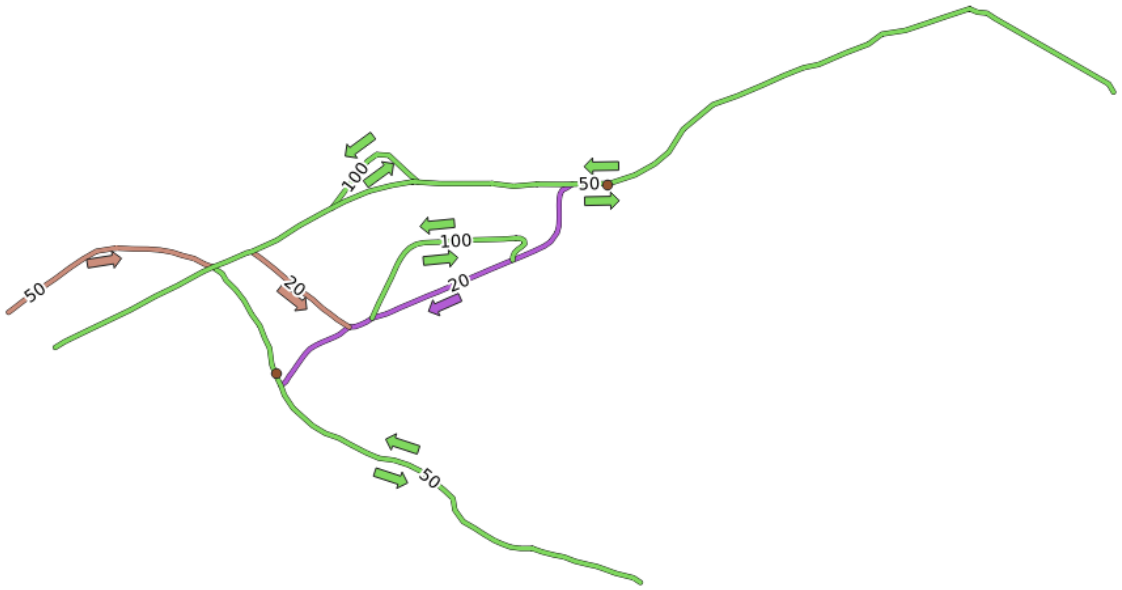
Вы можете найти все алгоритмы сетевого анализа в меню *Processing* ► *Network Analysis*. Как видите, там есть множество инструментов:



Откройте проект `exercise_data/network_analysis/network.qgz`. Он состоит из двух слоев:

- `network_points`.
- `network_lines`.

Слой *network_lines* уже показывает направления, которые помогают понять дорожную сеть.



Инструменты поиска кратчайшего пути позволяют рассчитать кратчайший или самый быстрый путь между двумя точками сети при условии, что:

- на карте выбраны начальная и конечная точки,
- начальная точка выбрана на карте, а конечные точки взяты из слоя точек,
- начальные точки взяты из слоя точки, конечная точка отобрана на карте.

Давайте начнем.

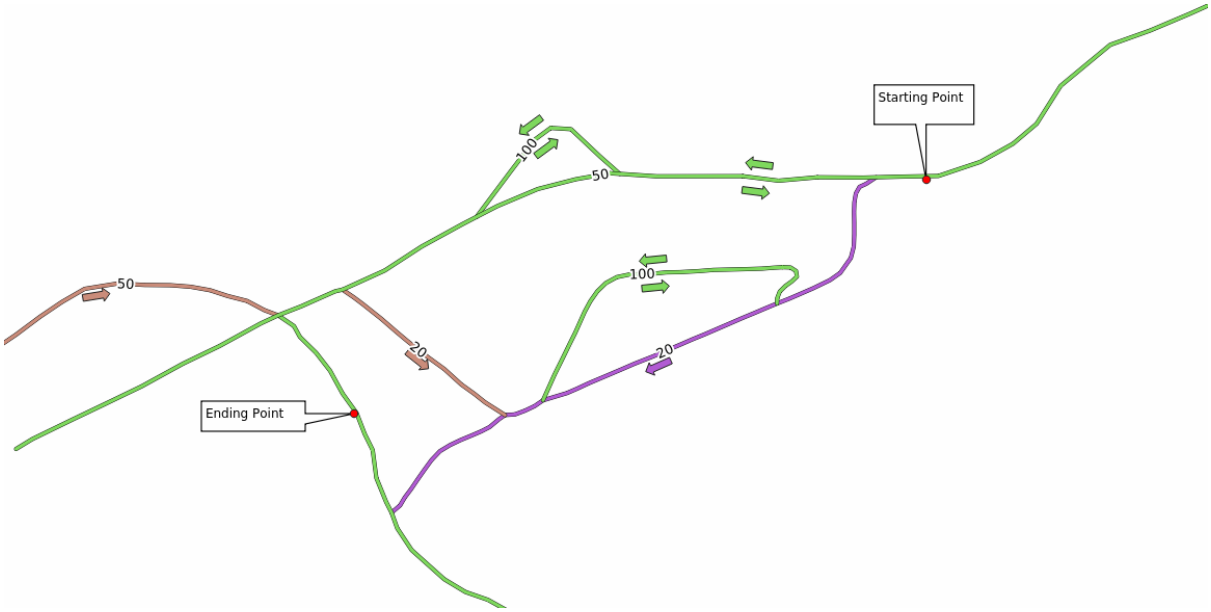
6.3.2 Рассчитать кратчайший путь (от точки к точке)

Network analysis ► *Shortest path (point to point)* позволяет вам рассчитать кратчайшее расстояние между двумя вручную выбранными точками на карте.

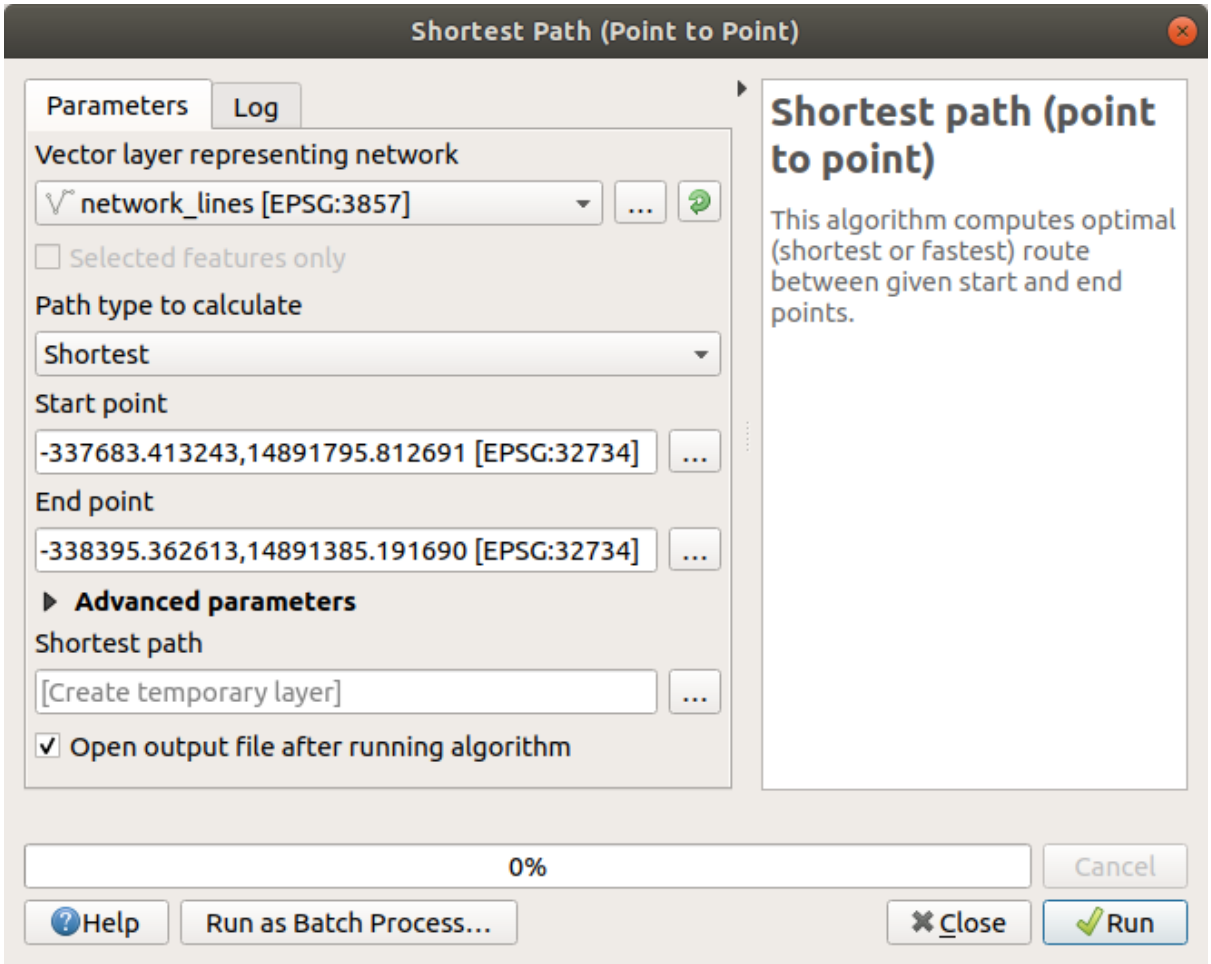
На данном примере мы рассчитаем **кратчайший** (не самый быстрый) путь между двумя точками.

1. Откройте алгоритм *Shortest path (point to point)*.
2. Выберите *network_lines* в *Vector layer representing network*.
3. Используйте `Shortest` в *Path type to calculate*.

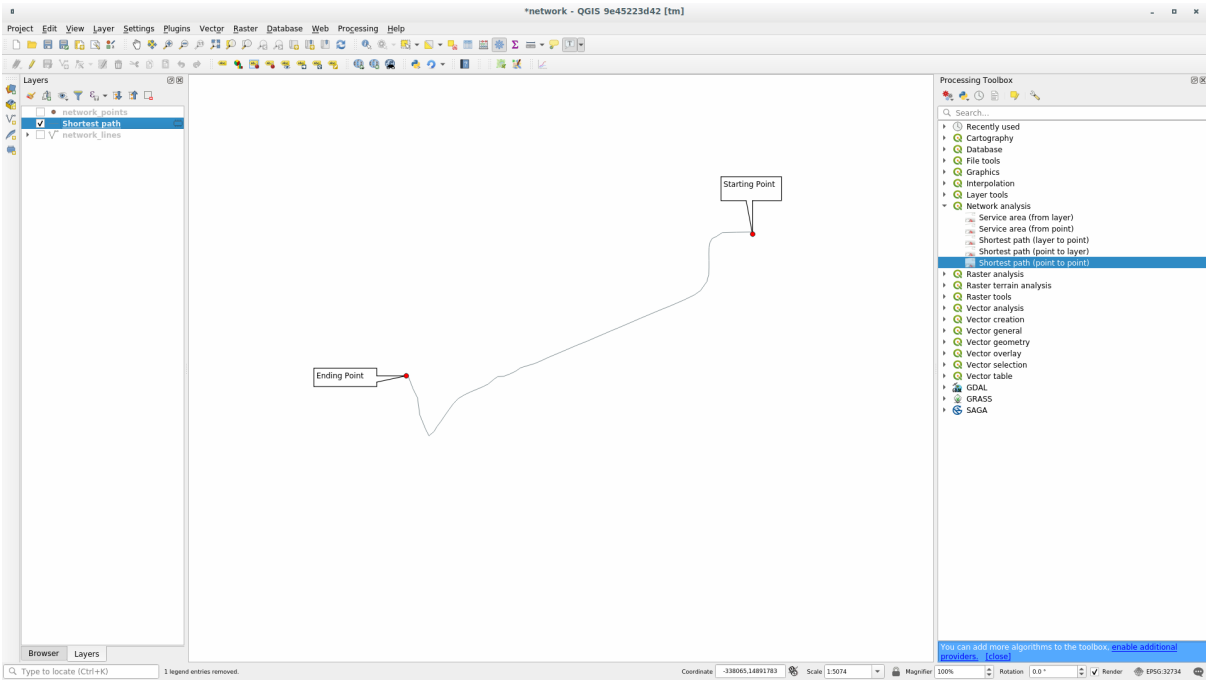
   Используйте эти две точки как начальную и конечную точки для анализа:



4. Кликни на кнопку … рядом с *Start point (x, y)* и выберите место и пометьте `Starting Point` на картинке. Координаты отобранной точки добавлены.
5. Сделайте то же самое, но выбрав место, помеченное `Ending point` для *End point (x, y)*.
6. Кликните на кнопку *Run*:



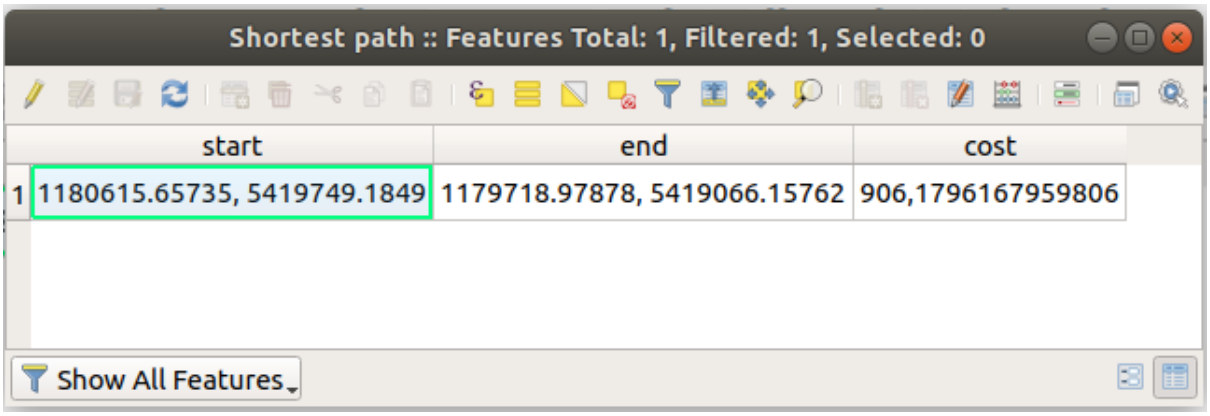
7. Новый линейный слой создан, который представляет кратчайший путь между выбранными точками. Уберите надпись в слое `network_lines`, чтобы лучше увидеть результат:



8. Откройте таблицу атрибутов выходного слоя. Там есть три поля, представляющие координаты начальной и конечной точек и **стоимость/затраты**.

   Мы выбираем `Shortest` в виде *Path type to calculate*, чтобы **стоимость** представляла собой **расстояние** в единицах слоя между двумя местоположениямиучастками.

   В нашем случае *кратчайшее* расстояние между выбранными точками составляет около `1000` метров:



Теперь, когда вы знаете, как пользоваться этим инструментом, вы можете попробовать с другими участками.

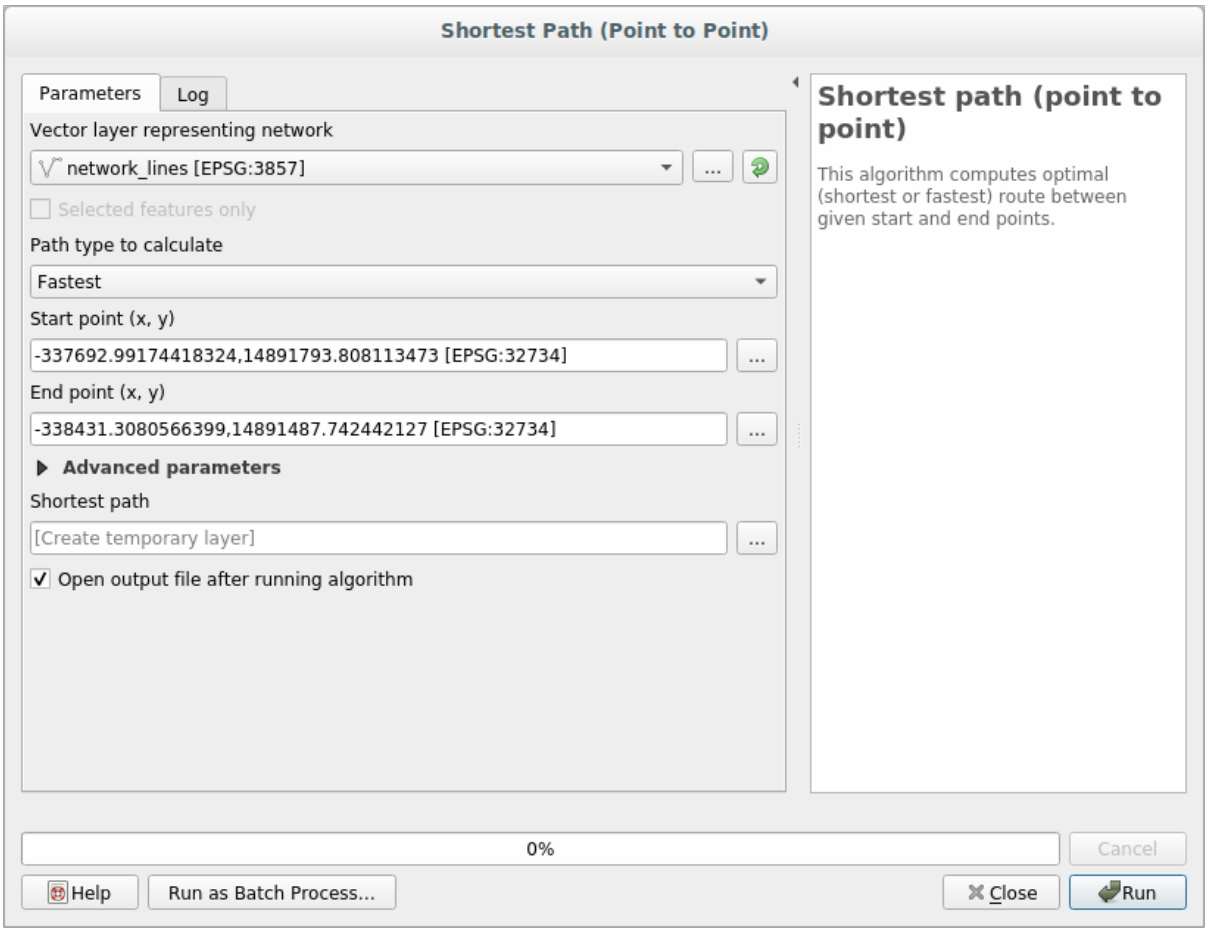
6.3.3  Попробуйте сами: Самый быстрый путь

Используя те же данные, что и в предыдущем упражнении, попробуйте вычислить самый быстрый путь между двумя точками.

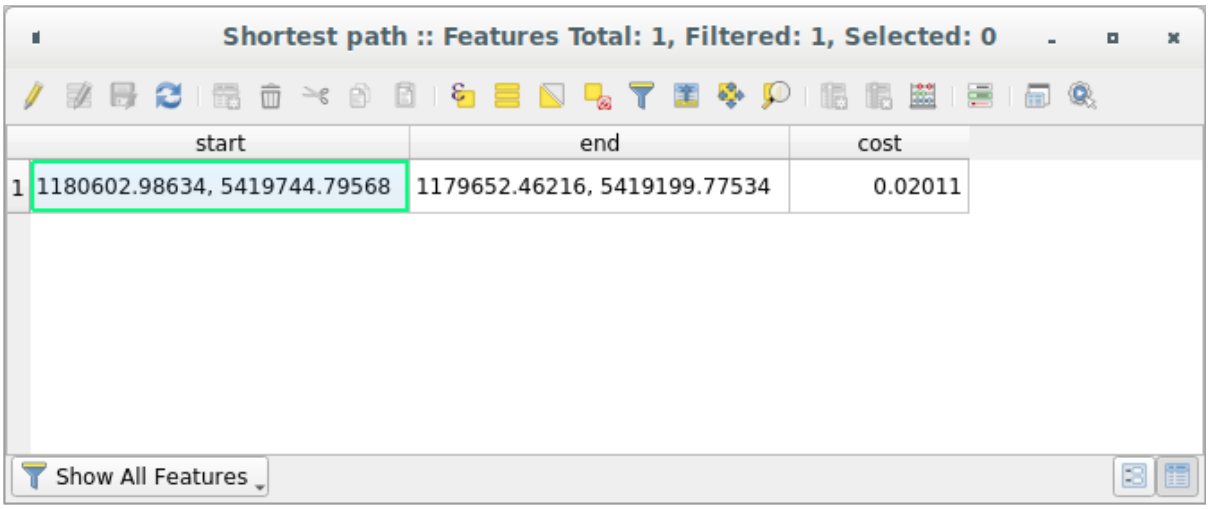
Сколько времени вам нужно, чтобы пройти от начальной до конечной точки?

Answer

1. Open *Network Analysis* ► *Shortest Path (Point to Point)* and fill the dialog as:



2. Make sure that the *Path type to calculate* is `Fastest`.
3. Click on *Run* and close the dialog.
4. Open now the attribute table of the output layer. The *cost* field contains the travel time between the two points (as fraction of hours):

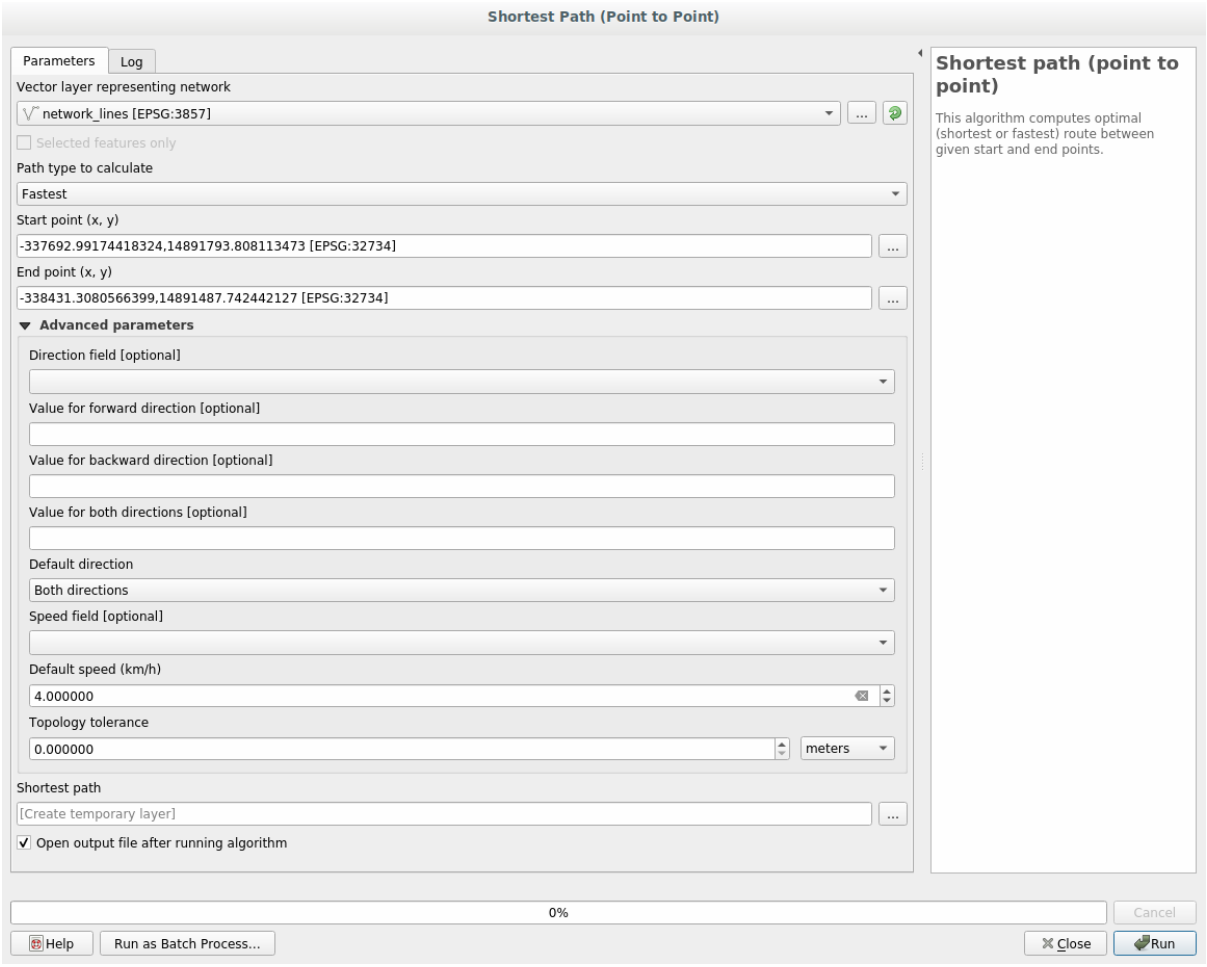


6.3.4 Идем дальше: Дополнительные опции

Let us explore some more options of the Network Analysis tools. In the *previous exercise* we calculated the **fastest** route between two points. As you can imagine, the time depends on the travel **speed**.

Мы используем те же слои, начальные и конечные точки предыдущих упражнений.

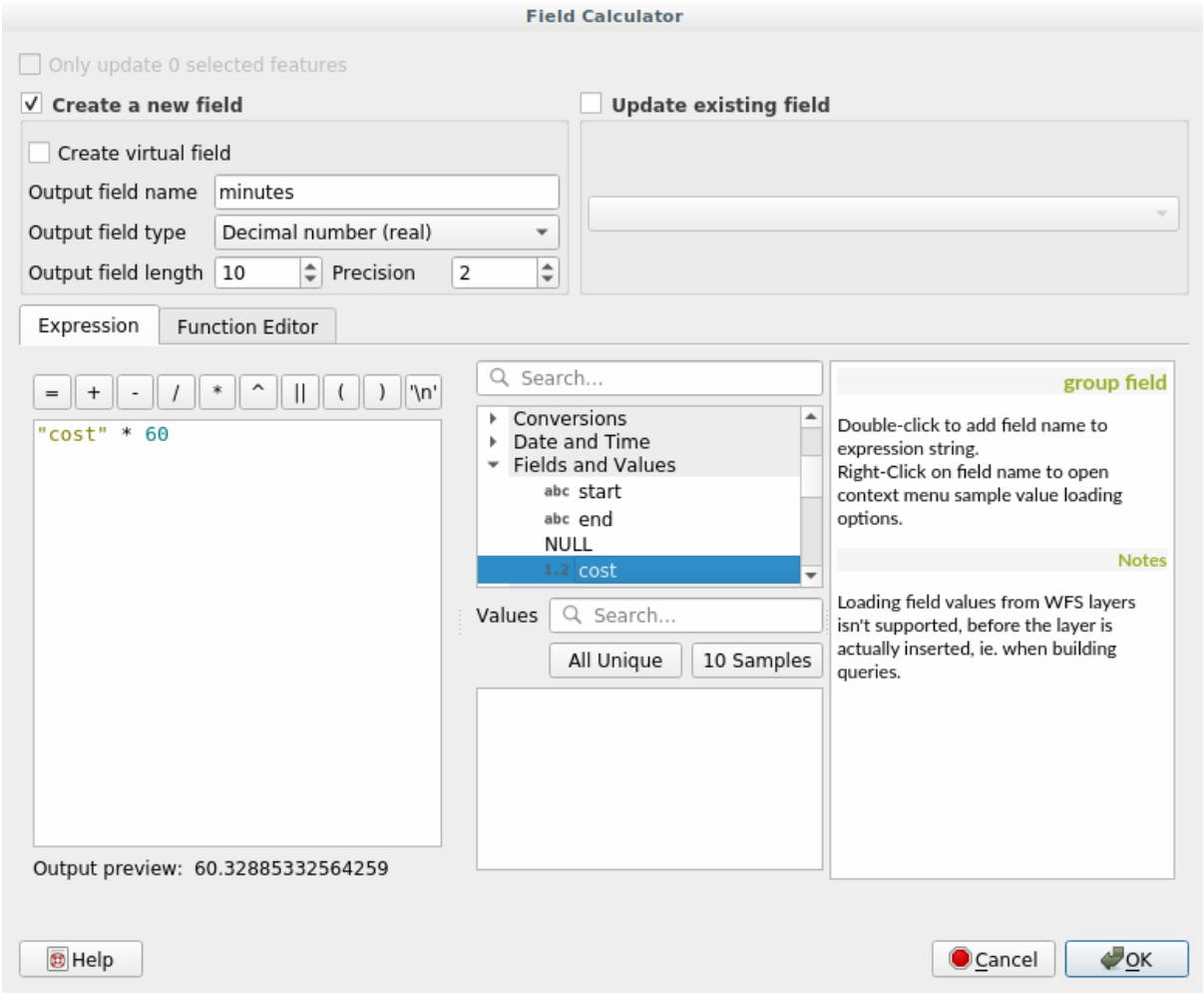
1. Откройте алгоритм *Shortest path (point to point)*.
2. Заполните *Input layer*, *Start point (x, y)* а также *End point (x, y)* как мы делали ранее.
3. Выберите `Fastest` в *Path type to calculate*.
4. Открой меню *Advanced parameter*.
5. Измените *Default speed (km/h)* значение по умолчанию `50` на `4`.



6. Кликните на кнопку *Run*.
7. После завершения алгоритма закройте диалоговое окно и откройте таблицу с атрибутами выходного слоя.

   Поле стоимости содержит значение в соответствии с выбранным вами параметром скорости. Мы можем преобразовать поле стоимости из часов с дробями в более удобочитаемые значения - *минуты*.

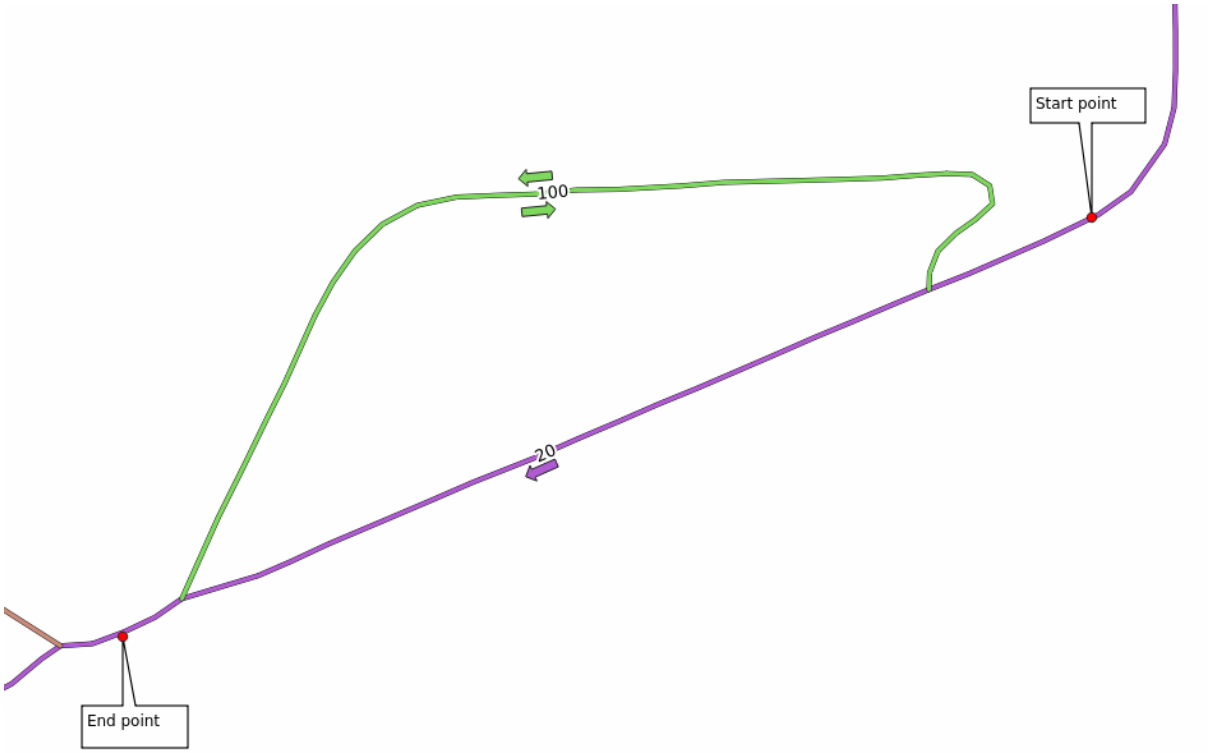
8. Откройте поле калькулятора, нажав на значок  и добавьте новое поле *minutes*, умножив поле *cost* на 60:



Вот и все! Теперь вы знаете, сколько минут потребуется, чтобы добраться из одной точки в другую.

6.3.5 Кратчайший путь с ограничением скорости

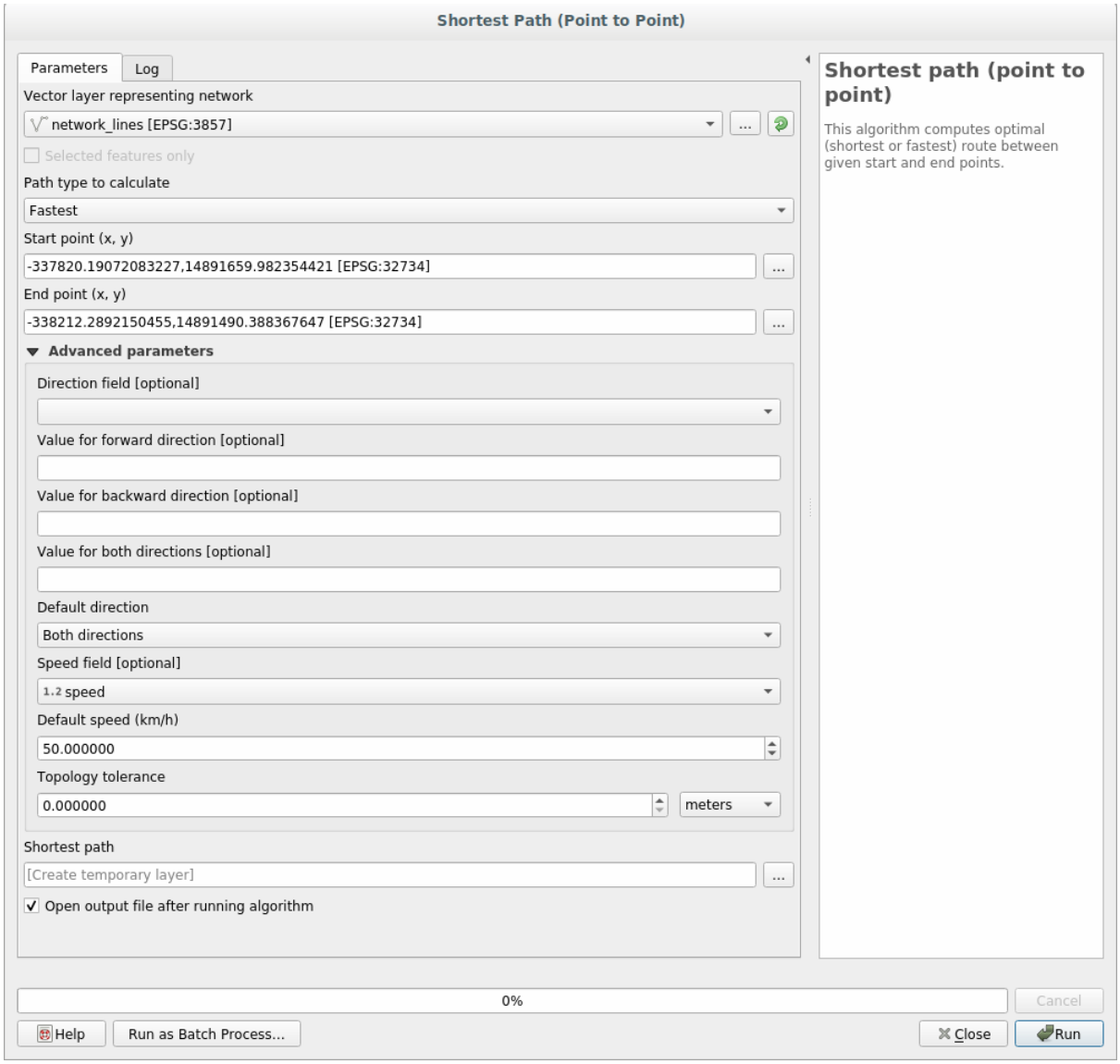
На панели инструментов сетевого анализа имеются и другие интересные опции. Посмотрите на эту карту:



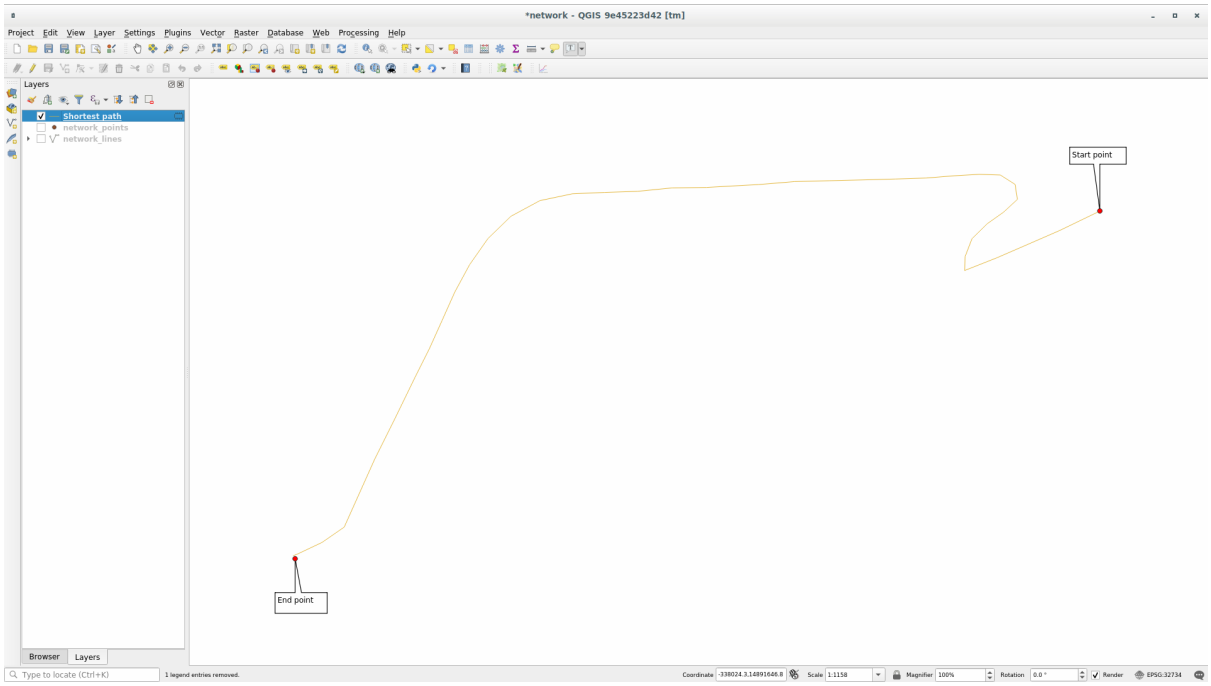
Мы хотели бы узнать **самый быстрый** маршрут с учетом **ограничений скорости** на каждом участке дороги (надписи обозначают ограничения скорости в км / ч). Наикратчайший путь без учета ограничений скорости, конечно же, фиолетовый маршрут. Но на этом участке дороги ограничение скорости составляет 20 км / ч, а по зеленой дороге можно ехать со скоростью 100 км/ч!

Как и в первом упражнении, мы используем *Network analysis* ► *Shortest path (point to point)* и вручную выберем начальную и конечную точки.

1. Откройте алгоритм *Network analysis* ► *Shortest path (point to point)*.
2. Выберите *network_lines* в параметрах *Vector layer representing network*.
3. Выберите `Fastest` в *Path type to calculate*.
4. Кликни на кнопку … рядом с *Start point (x, y)* и выберите начальную точку.
5. То же самое проделайте по *End point (x, y)*.
6. Откройте меню *Advanced parameters*.
7. Выберите поле *скорости* в качестве параметра *Speed Field*. С этой опцией алгоритм будет учитывать ограничения скорости для каждой дороги.



8. Нажмите на кнопку *Run*.
9. Выключите слой `network_lines`, чтобы лучше увидеть результат.



Как видите, самый быстрый маршрут не соответствует самому короткому.

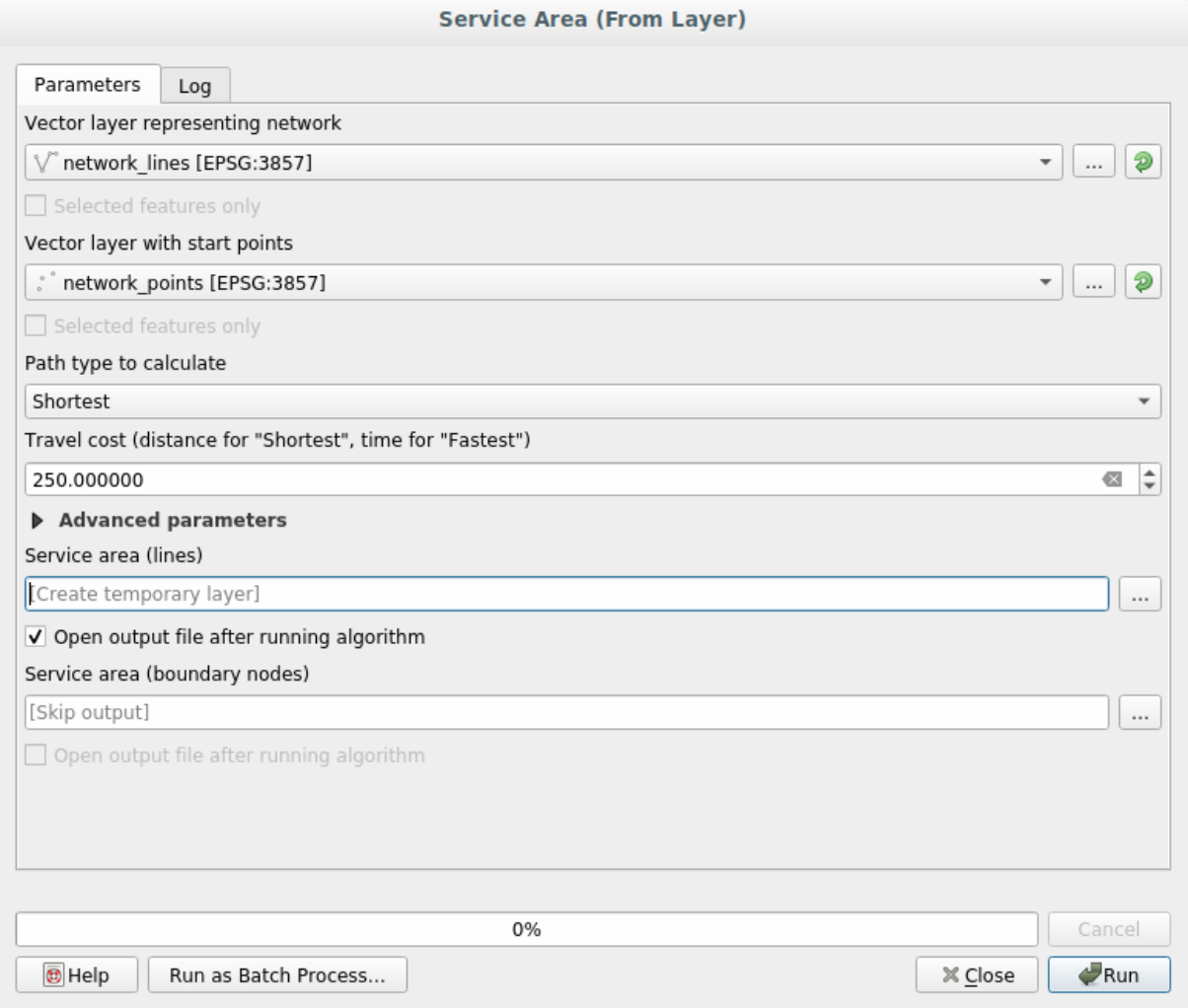
6.3.6 Зона обслуживания (из слоя)

Алгоритм *Network Analysis* ► *Service area (from layer)* может ответить на вопрос: какие местности доступны с учетом значения расстояния или времени?

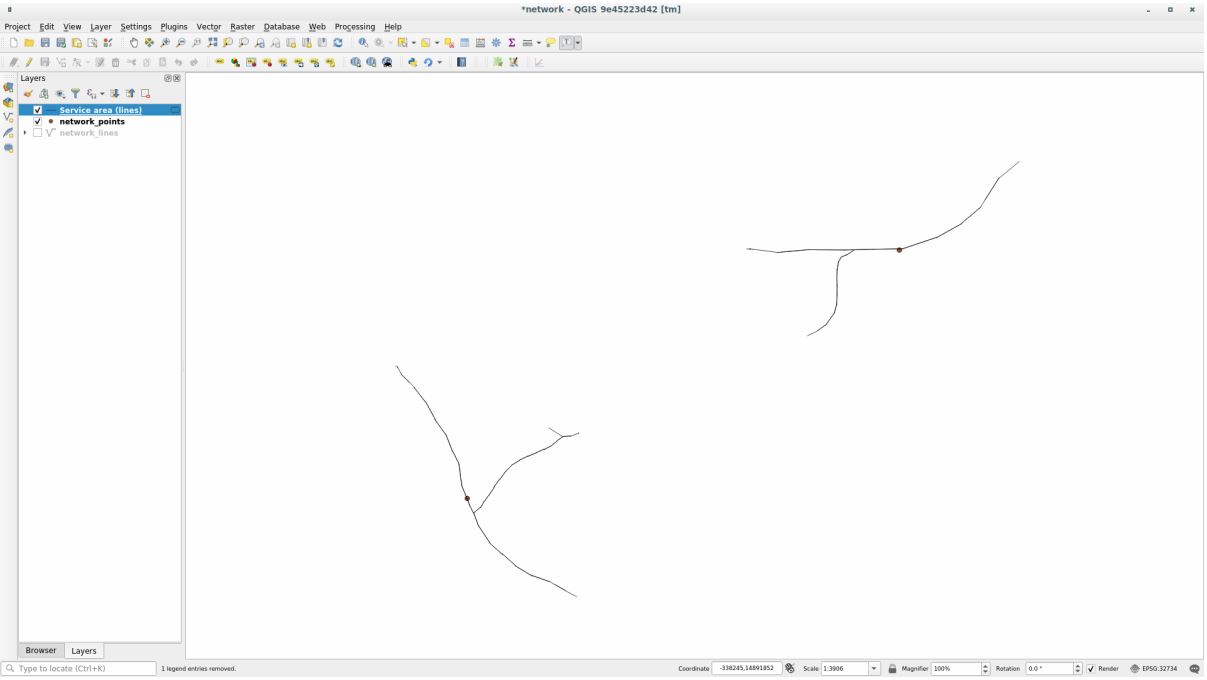
Примечание: *Network Analysis* ► *Service area (from point)* является тем же алгоритмом, но он позволяет вручную выбрать точку на карте.

С учетом расстояние `250` метров, мы хотим узнать, как далеко мы можем пройти в сети от каждой точки слоя *network_points*.

1. Снимите флажок со всех слоев, кроме `network_points`.
2. Откройте алгоритм *Network Analysis* ► *Service area (from layer)*.
3. Выберите `network_lines` для *Vector layer representing network*.
4. Выберите `network_points` для *Vector layer with start points*.
5. Выберите `Shortest` в *Path type to calculate*.
6. Наберите `250` для параметра *Travel cost*.
7. Кликните на кнопку *Run* и закройте диалоговое окно.



Выходной слой представляет собой максимальный путь, который вы можете пройти от точечных объектов на расстоянии 250 метров:



Круто, не правда ли?

6.3.7 В заключении

Теперь вы узнали, как пользоваться алгоритмом *Network analysis* для решения задач, связанных с кратчайшим и самыми быстрым маршрутом.

Мы теперь сможем выполнить некоторые действия по пространственной статистике с данными векторного слоя. Поехали!

6.3.8 Что дальше?

Мы покажем Вам, как запускать алгоритмы пространственной статистики для векторных наборов данных.

6.4 Занятие: Пространственная статистика

Примечание: Урок подготовлен Linfiniti и S Motala (Технологический университет Кейп-Пенинсула).

Пространственная статистика позволяет вам сделать анализ и понять, что происходит в данном наборе векторных данных. QGIS имеет множество полезных инструментов для статистического анализа.

Цель этого урока: Узнать, как использовать инструменты пространственной статистики QGIS внутри *Processing Toolbox*.

6.4.1  Идем дальше: Создаем тестовый набор данных

Мы создадим случайный набор точек и получим набор данных для работы.

Чтобы это сделать вам понадобится набор данных полигона, чтобы определить местность, в которой вы хотите создать точки.

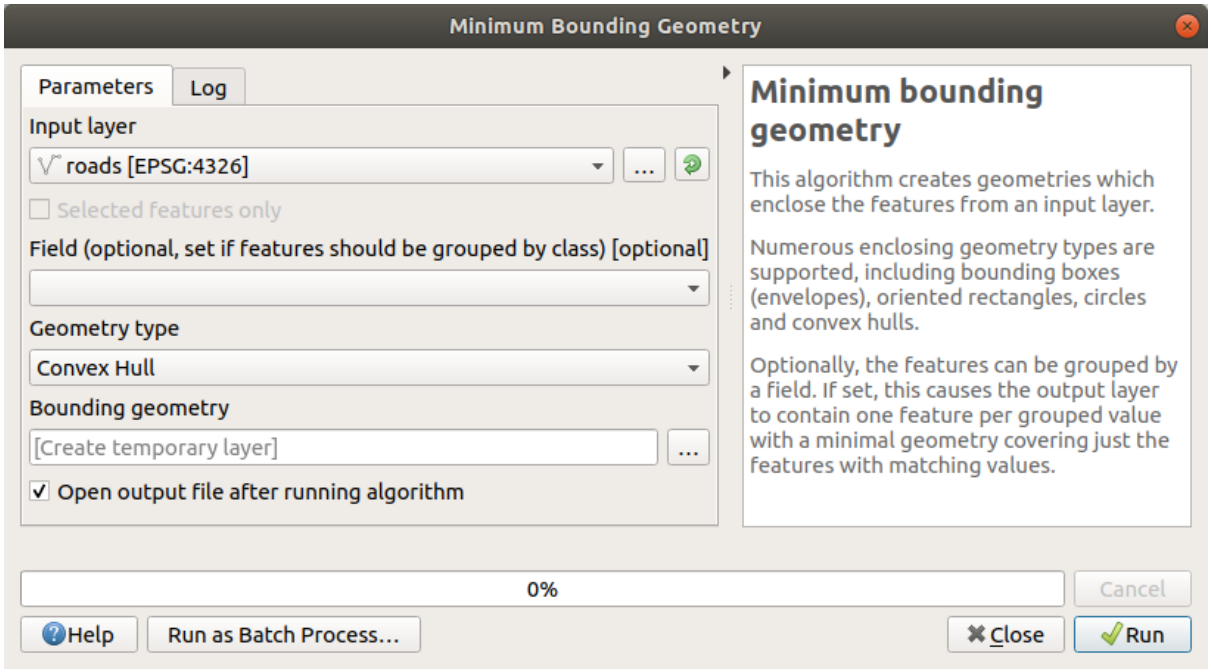
Мы будем использовать площадь, покрытую улицами.

1. Начните новый проект.
2. Добавьте свой набор данных `roads`, а также `srtm_41_19` (данные о высоте) в `exercise_data/raster/SRTM/`.

   Примечание: Вы возможно обнаружите, что слой SRTM DEM имеет другую ССК, которая отличается от слоя дорог. QGIS пере-проецирует оба слоя в одной ССК. Для следующих упражнений эта разница не имеет значения, но вы можете пере-проецировать (как было показано ранее в этом модуле).

   ---

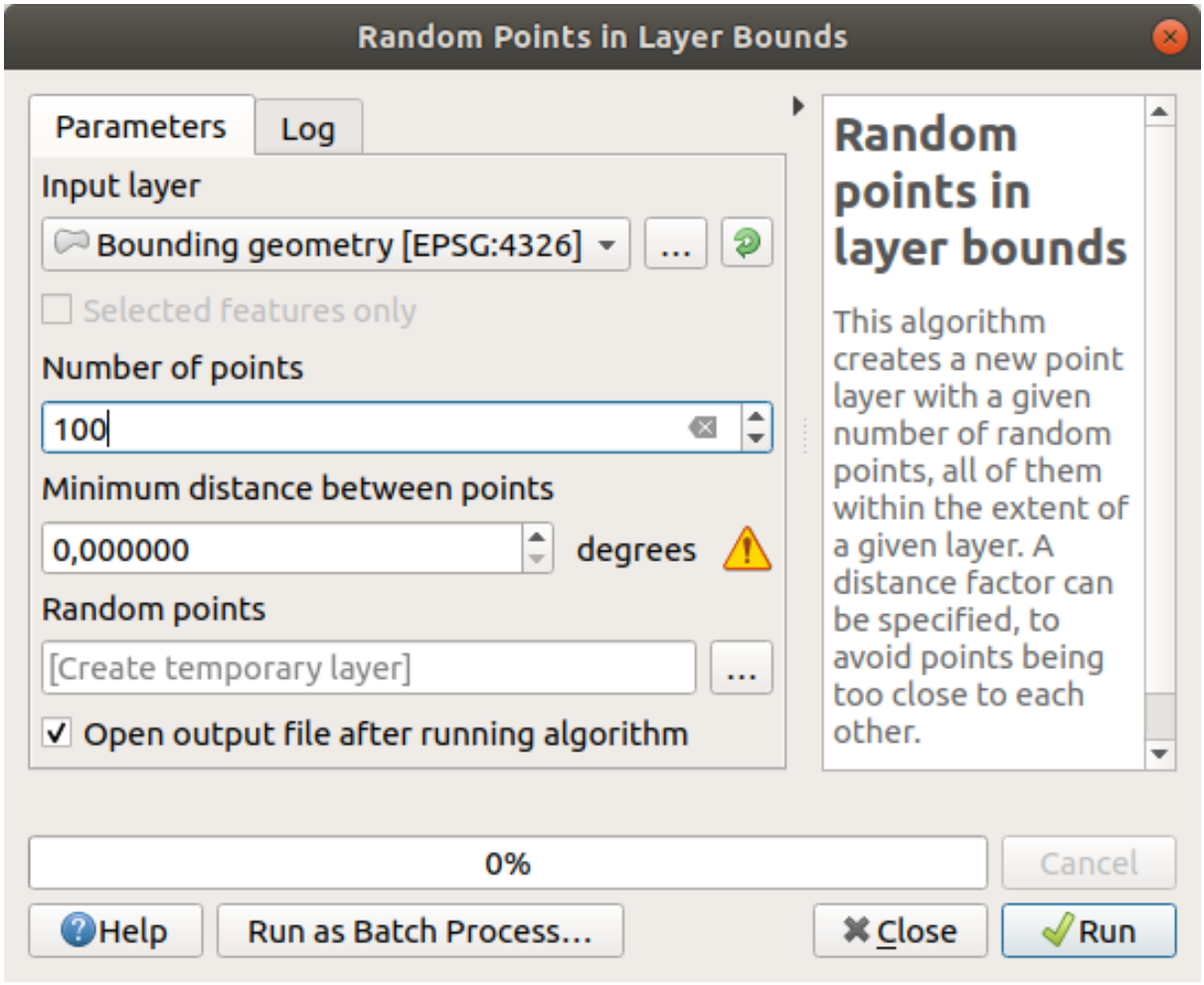
3. Откройте ящик для инструментов *Processing*.
4. Используйте инструмент *Vector Geometry* ► *Minimum bounding geometry* для создания местности, привязав все дороги, выбрав `Convex Hull` в качестве *Geometry Type*:



Как известно, если вы не укажете выход, *Обработка* создаст временные слои. Вы можете сразу сохранить слои или сделать это позже.

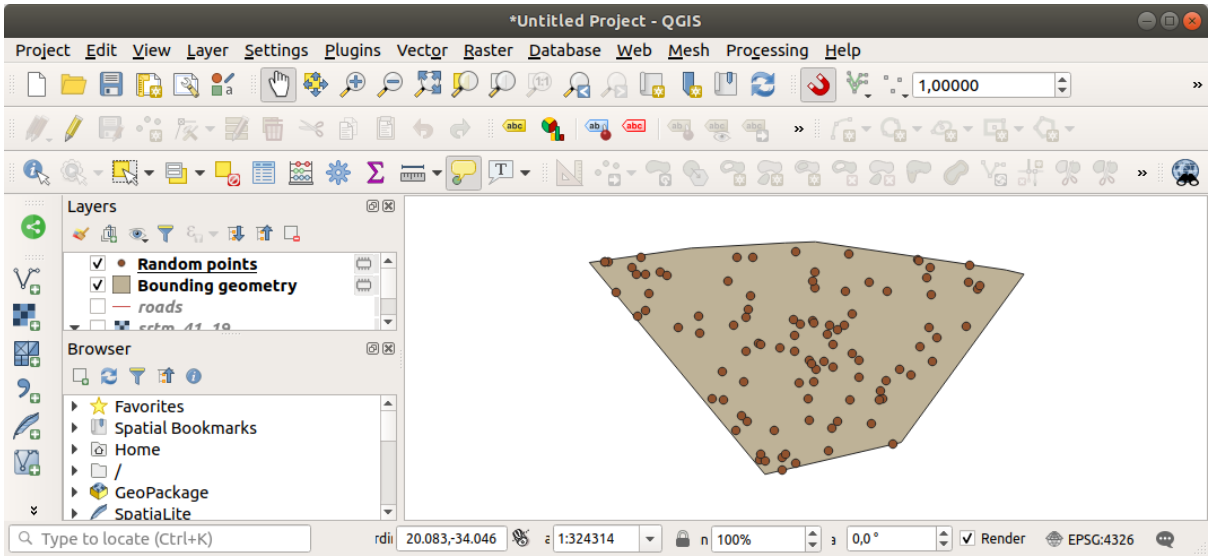
Создание случайных точек

- Создайте 100 случайных точек в этой местности, используя инструмент в *Vector Creation* ► *Random points in layer bounds* с минимальным расстоянием `0.0`:



Примечание: Желтый предупреждающий знак говорит вам, что этот параметр относится к расстояниям. Слой *Bounding geometry* находится в географической системе координат и алгоритм просто напоминает вам об этом. В этом примере мы не будем использовать этот параметр, поэтому вы можете его проигнорировать.

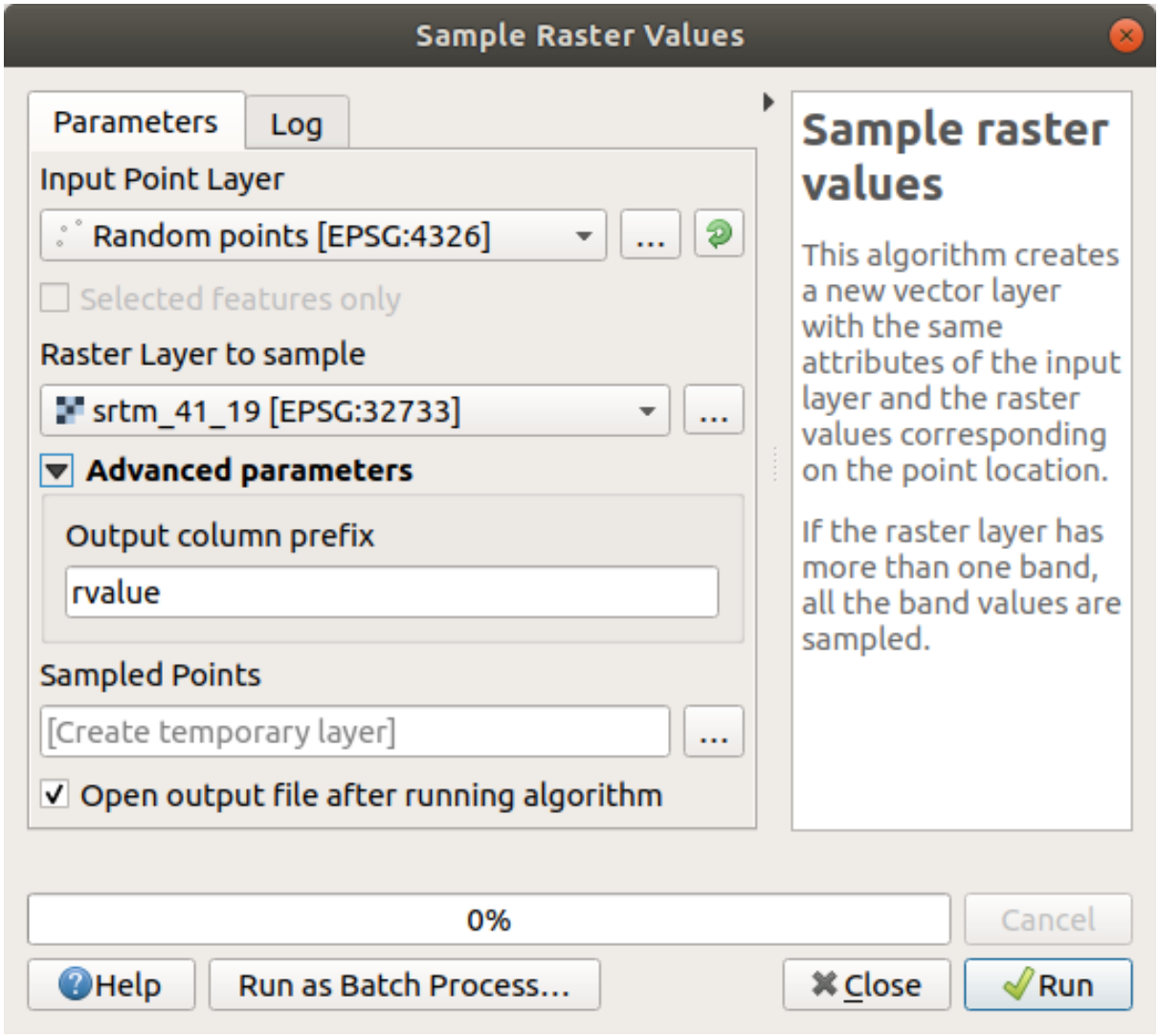
Если надо, переместите сгенерированные случайные точки в верхнюю часть условного обозначения, чтобы лучше их увидеть:



Выборка данных

Для создания выборочного набора данных из растра, вам надо использовать алгоритм *Raster Analysis* ► *Sample raster values*. Этот инструмент производит выборку растра в местах расположения точек и добавляет значения растра в новое поле (поля) в зависимости от количества каналов в растре.

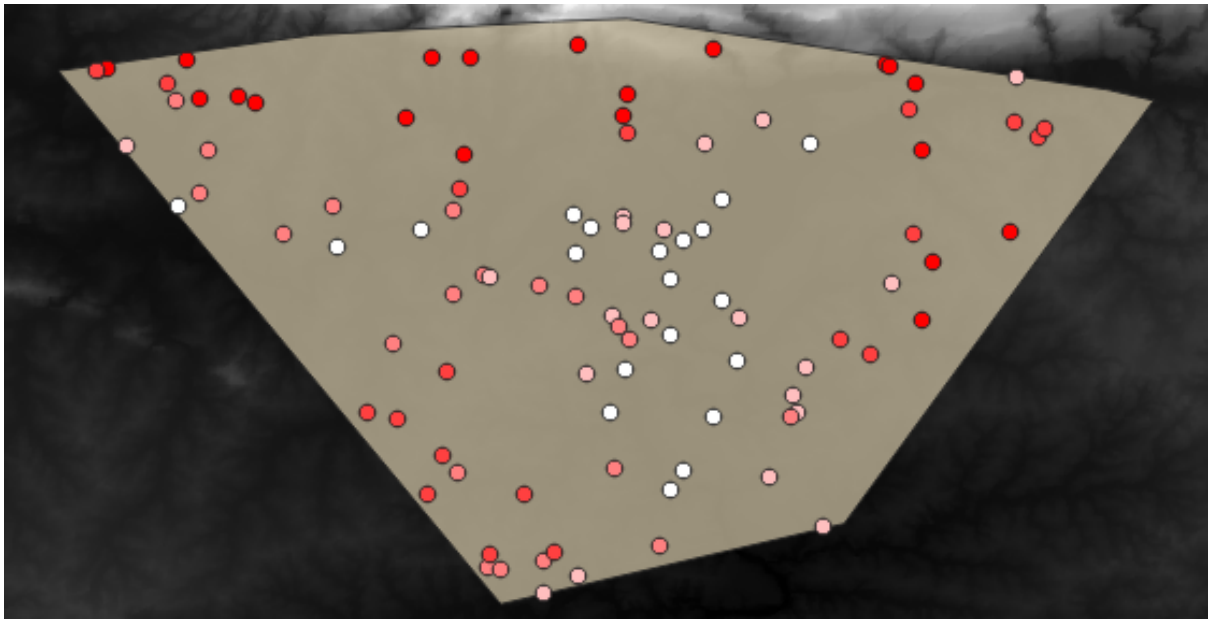
1. Откройте диалоговое окно алгоритма *Sample raster values*.
2. Выберите `Random_points` как слой, содержащий точки выборки, и растр SRTM как диапазон, из которого нужно получить значения. Название нового поля по умолчанию: `rvalue_N`, где `N` - номер полосы растра. Вы можете изменить название приставки, если хотите.



3. Кликните на кнопку *Run*.

Вы теперь сможете проверить выборочные данные из растрового файла в таблице атрибутов слоя `Sampled Points`. Они появятся в новом поле с названием, которое вы выбрали.

Возможный образец слоя показан здесь:



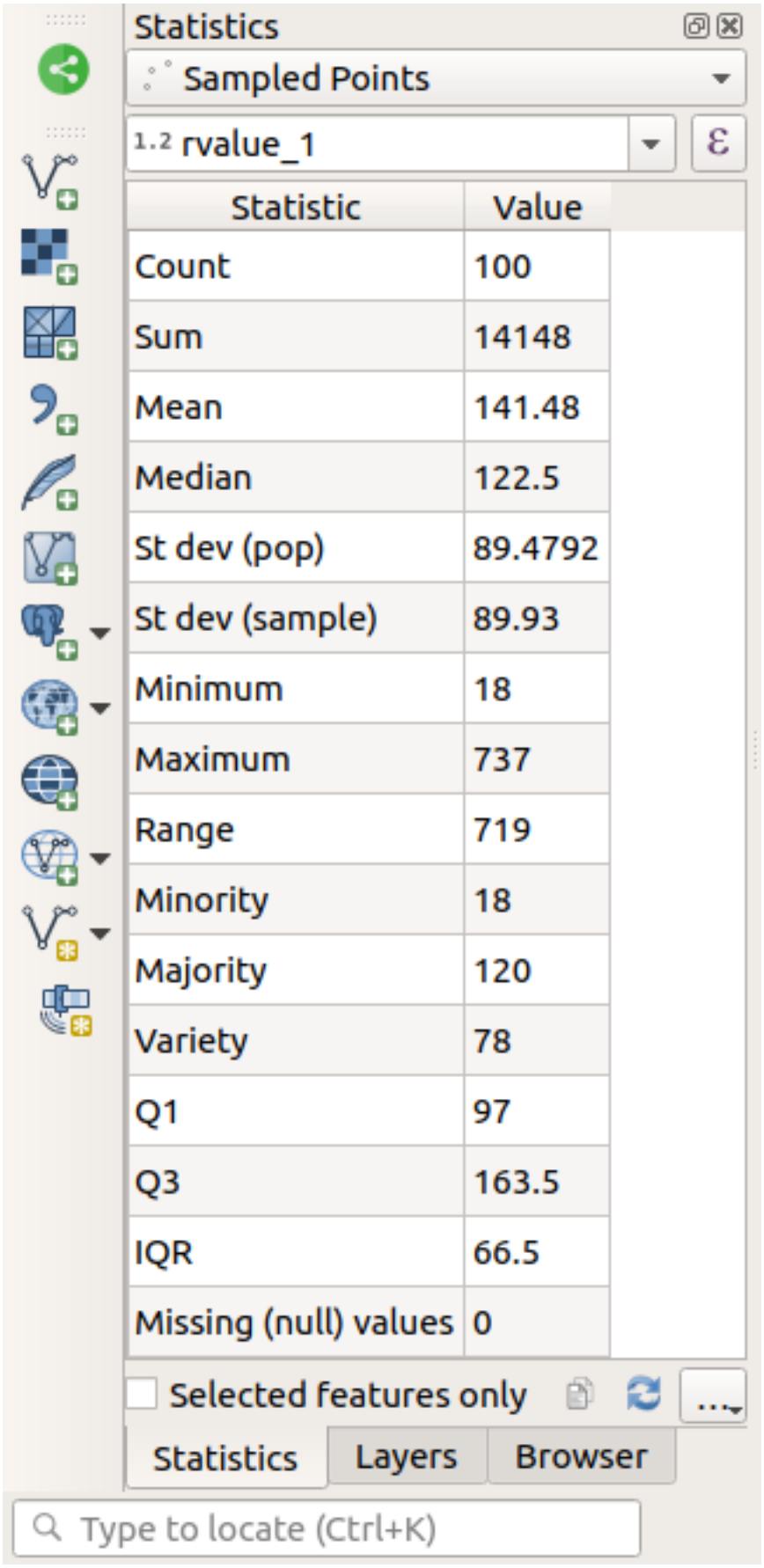
Выборочные точки классифицируются с использованием поля `rvalue_1` так, чтобы красные точки находились на большей высоте.

Вы будете использовать этот слой для остальных статистических упражнений.

6.4.2  Идем дальше: Базовая статистика

Вам теперь надо получить базовую статистику для этого слоя.

1. Кликни на значок Показать статистку  Show statistical summary в *Attributes Toolbar*. Появится новая панель.
2. В появившемся диалоговом окне укажите слой `Sampled Points` в качестве источника.
3. Выберите поле *rvalue_1* в поле со списком полей. Это поле, для которого вы будете рассчитывать статистику.
4. Панель *Statistics* автоматически обновится с расчетными статданными:


Statistics
Sampled Points
1.2 rvalue_1
Statistic
Value
Count
100
Sum
14148
Mean
141.48
Median
122.5
St dev (pop)
89.4792
St dev (sample)
89.93
Minimum
18
Maximum
737
Range
719
Minority
18
Majority
120
Variety
78
Q1
97
Q3
163.5
IQR
66.5
Missing (null) values
0
Selected features only
Statistics
Layers
Browser
Type to locate (Ctrl+K)

Примечание: Вы можете скопировать значения, нажав на кнопку Copy Statistics To Clipboard и вставив результаты в таблицу.

5. Закройте панель *Statistics* когда закончите.

Появятся много команд, связанных со статистикой:

Count/Считать Количество выборок /значений.

Sum/Сумма Сложение значений.

Mean/Среднее значение Среднее (среднее) значение - это просто сумма значений, разделенная на количество значений.

Median/Медиана Если вы расположите все значения от наименьшего к наибольшему, значение в середине (или среднее из двух средних значений, если N - четное число) будет медианным значением.

St Dev (pop)/Стандартное отклонение Стандартное отклонение. Указывает, насколько близко значения сгруппированы вокруг среднего. Чем меньше стандартное отклонение, тем ближе значения к среднему.

Minimum/Минимум Минимальное значение.

Maximum/Максимум Максимальное значение.

Range/Диапазон Разница между минимальным и максимальным значениями.

Q1 Первый квартиль данных.

Q3 Третий квартиль данных.

Missing (null) values/Пропущенные (нулевые) значения Количество пропущенных значений.

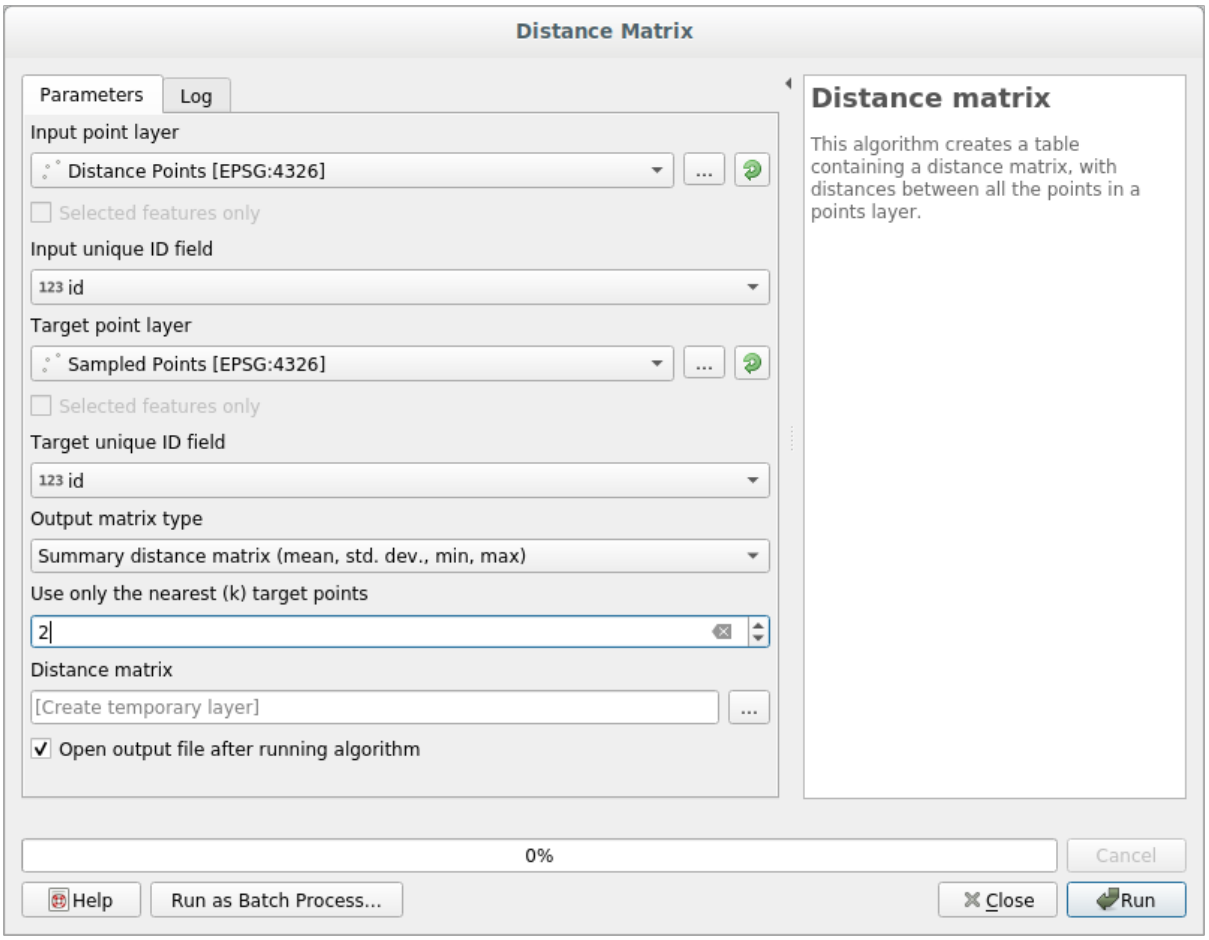
6.4.3  Идем дальше: Вычисляем статистические данные относительно расстояний между точками

1. Создайте новый временный точечный слой.
2. Зайдите в режим редактирования и оцифруйте три точки где-нибудь среди других точек.

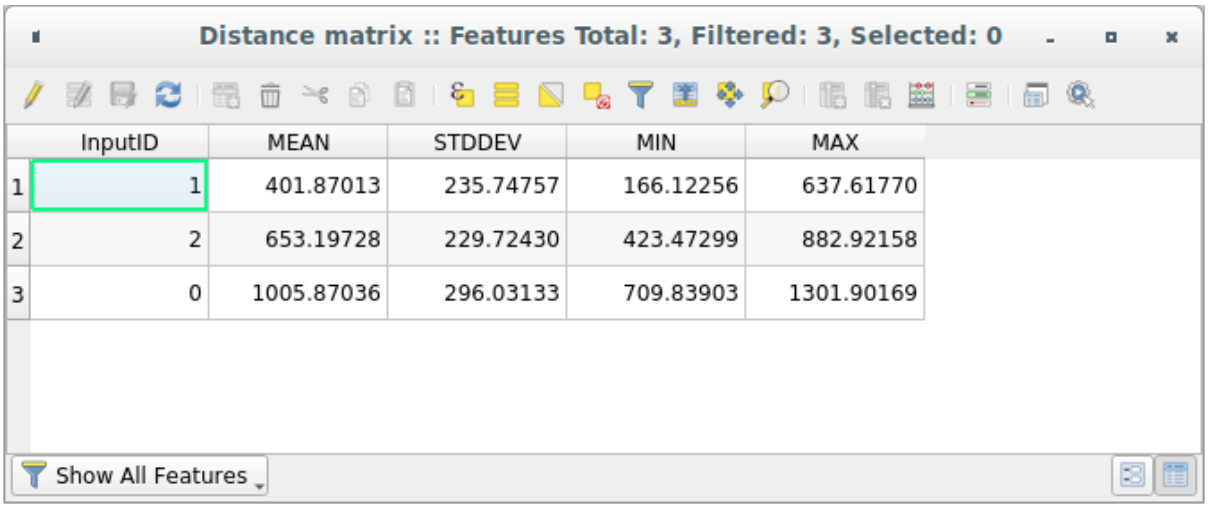
   В качестве альтернативы вы можете использовать тот же метод генерации случайных точек, как ранее мы делали, но надо указать только **три** точки.
3. Сохраните ваш новый слой как *distance_points* в том формате, к котором вам хочется.

Чтобы получить статистику расстояний между точками в двух слоях необходимо сделать следующее:

1. Откройте инструмент *Vector Analysis* ► *Distance matrix*.
2. Выберите слой `distance_points` в качестве входного слоя, а слой `Sampled Points` - в качестве целевого слоя.
3. Сделайте это так:



4. Если хотите, вы можете сохранить выходной слой как файл или просто можете запустить алгоритм и сохранить временный выходной слой позже.
5. Кликните на кнопку *Run* чтобы получить слой матрицы расстояний.
6. Откройте таблицу с атрибутами сгенерированного слоя: значения относятся к расстояниям между объектами *distance_points* и их двумя ближайшими точками в слое *Sampled Points*:



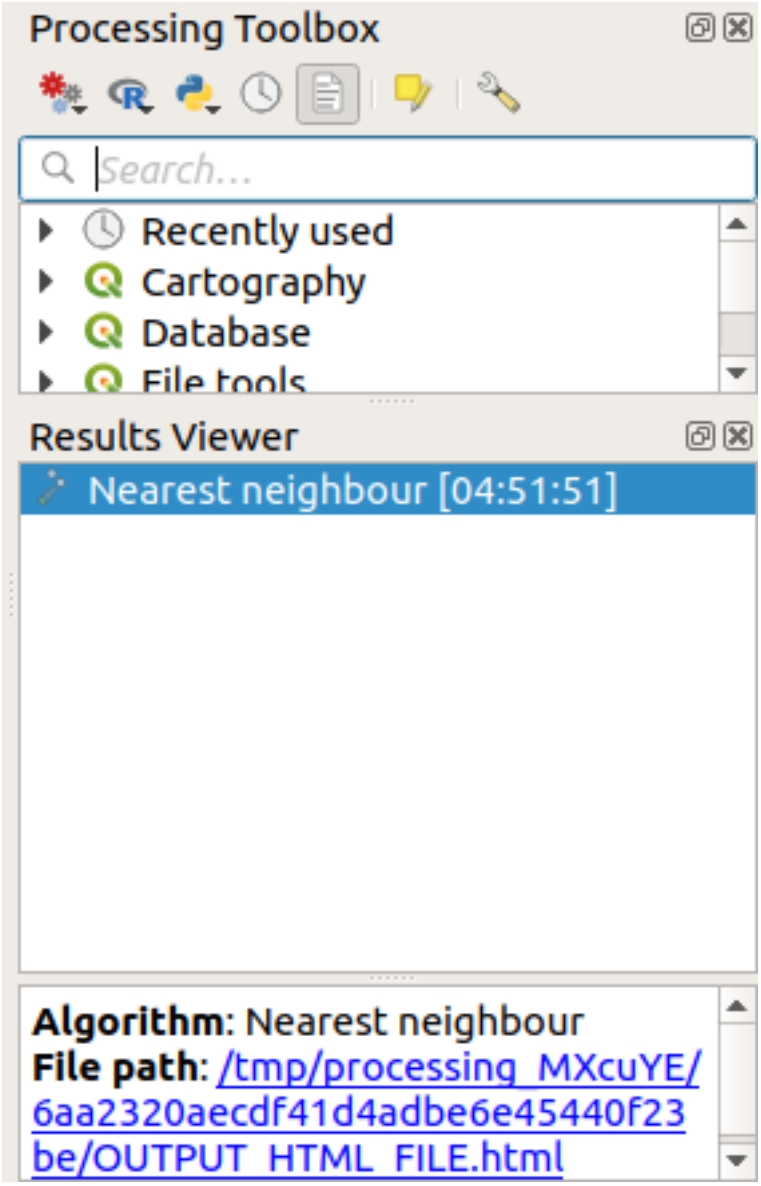
С помощью этих параметров инструмент *Матрица расстояний* рассчитает статистику расстояний для каждой точки входного слоя по отношению к ближайшим точкам целевого слоя. Поля выходного слоя содержат

среднее значение, стандартное отклонение, минимальные и максимальные значения в отношении расстояний до ближайших соседних точек входного слоя.

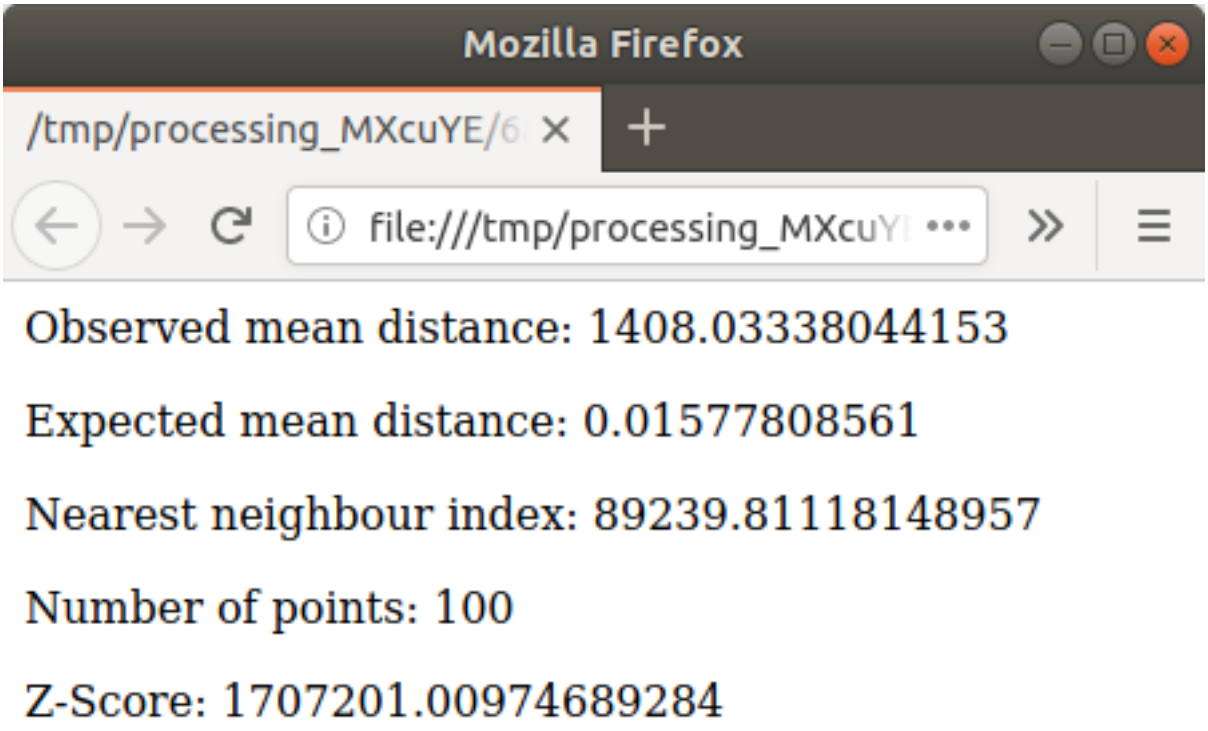
6.4.4  Идем дальше: Анализ методом «ближайший сосед» (внутри слоя)

Чтобы выполнить анализ методом «ближайший сосед» точечного слоя, необходимо сделать следующее:

1. Выберите *Vector analysis* ► *Nearest neighbor analysis*.
2. В появившемся диалоговом окне выберите слой *Random points* и кликните *Run*.
3. Результаты появятся в панели «Обработка» *Result Viewer*.



4. Кликните синюю ссылку, чтобы открыть страницу `html` с результатами:



6.4.5  Идем дальше: Средние координаты

Для того, чтобы получить средние координаты набора данных необходимо сделать следующее:

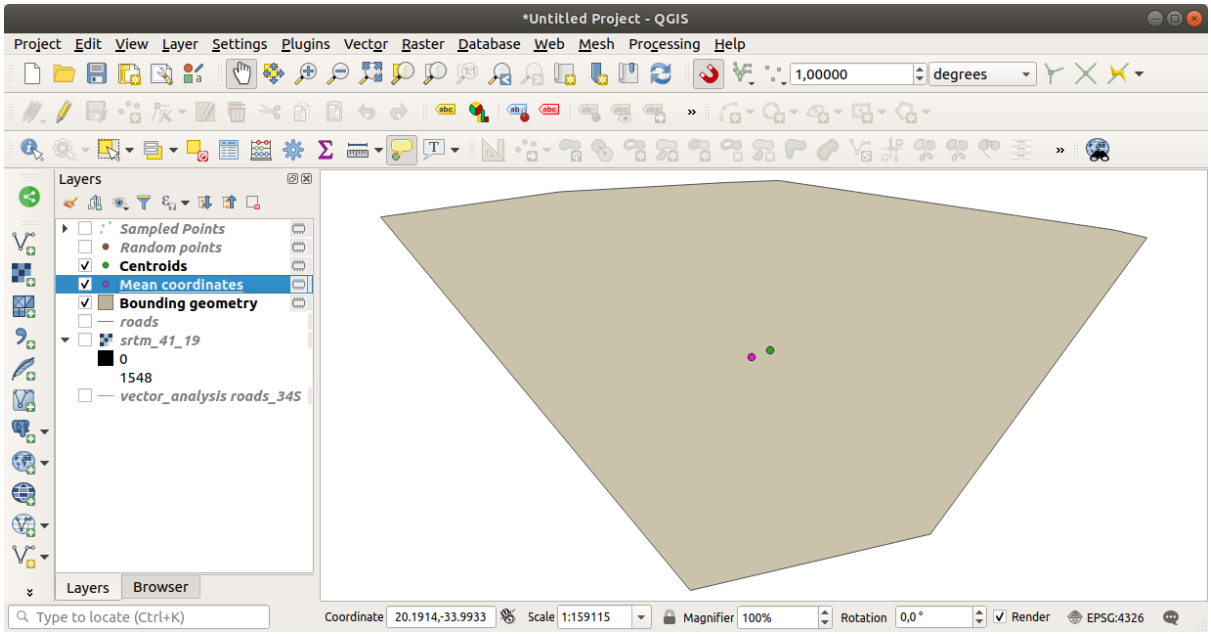
1. Запустите *Vector analysis* ► *Mean coordinate(s)*.
2. В появившемся диалоговом окне укажите *Random points* как *Input layer*, и оставьте необязательные варианты без изменений.
3. Кликните на кнопку *Run*.

Давайте сравним это с центральной координатой полигона, который был использован для создания случайной выборки.

1. Запустите *Vector geometry* ► *Centroids*.
2. В появившемся диалоговом окне выберите `Bounding geometry` в качестве входного слоя.

Как видите, средние координаты (розовая точка) и центр исследуемой местности (зеленый) не обязательно совпадают.

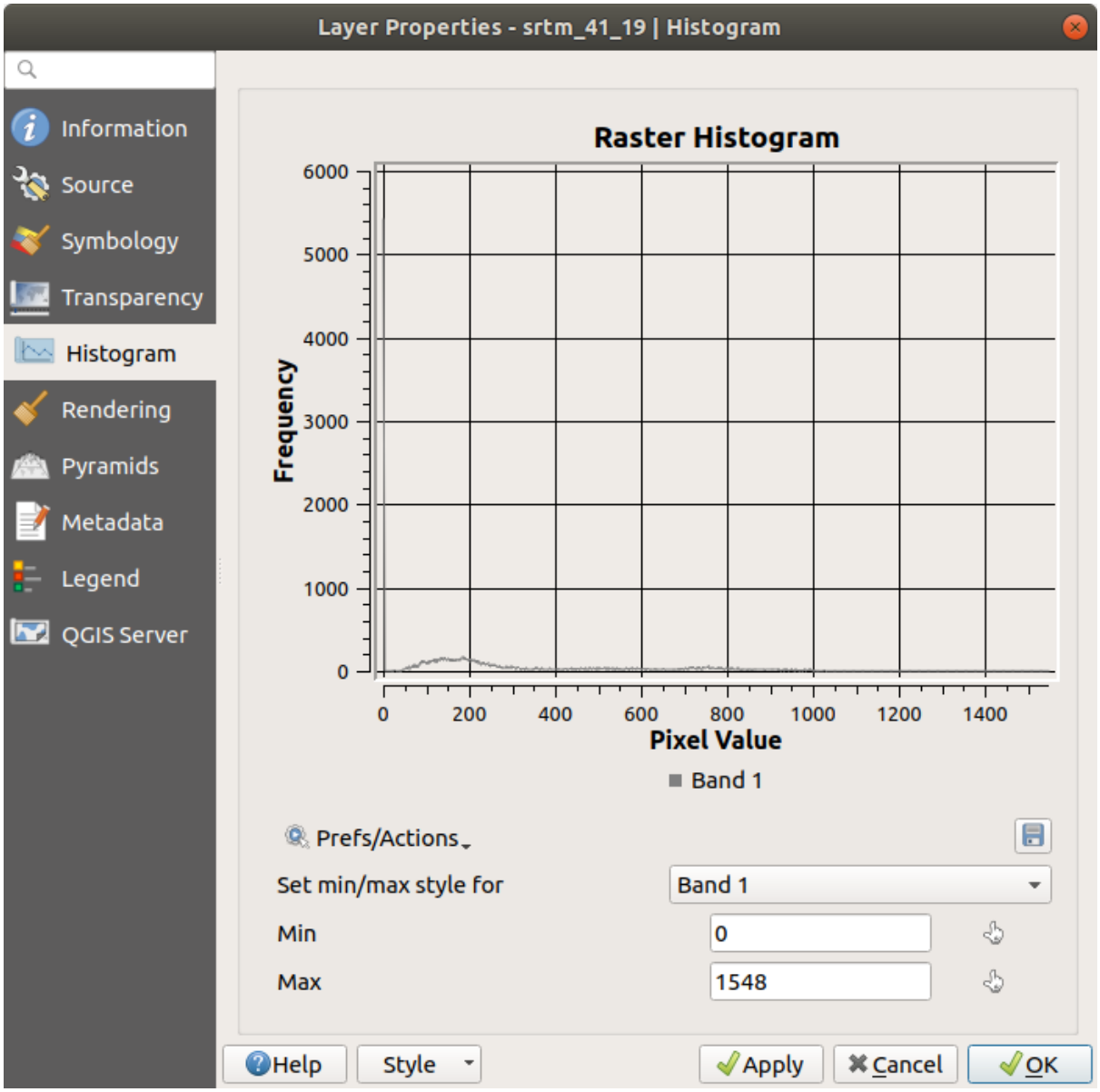
Центроид - это барицентр слоя (барицентр квадрата - это центр квадрата), а средние координаты представляют собой среднее значение всех координат узлов.



6.4.6  Идем дальше: Гистограммы изображений

Гистограмма набора данных показывает распределение его значений. Самый простой способ показать это в QGIS – через использование гистограммы изображения, которая имеется в диалоговом окне *Layer Properties* любого слоя изображения (набор растровых данных).

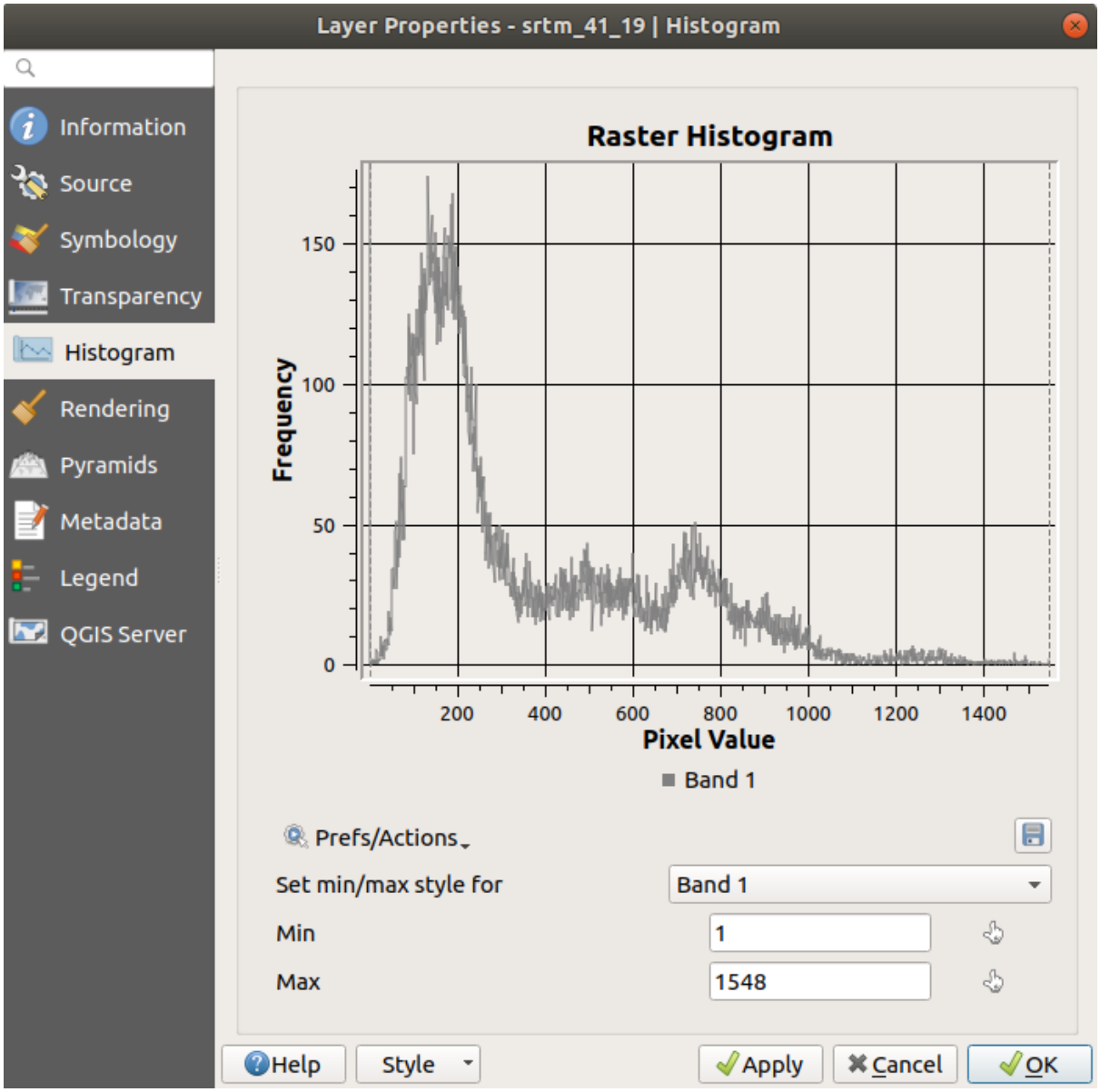
1. В вашей панели *Layers* кликните правой кнопкой мыши на слой `srtm_41_19`.
2. Выберите *Properties*.
3. Выберите *Histogram*. Вам надо кликнуть на кнопку *Compute Histogram* для того, чтобы создать график. Вы увидите график, показывающий частотное распределение значений растра.



4. График можно экспортировать как изображение с помощью кнопки с  Save plot.

5. Более подробную информацию можно получить о слое нажав на вкладку *Information* (среднее и максимальное значения являются расчетными и могут быть неточными).

Среднее значение `332.8` (расчетная 324,3), а максимальное значение равно `1699` (расчетная 1548)! Вы можете увеличить гистограмму. Поскольку там много пикселей со значением `0`, гистограмма выглядит сжатой по вертикали. Увеличьте масштаб чтобы охватить все, кроме пика на `0`, и вы увидите более подробную информацию:



Примечание: Если средние и максимальные значения не такие, как выше, это может быть из-за вычисления минимального/максимального значения. Откройте вкладку *Symbology* и расширьте меню *Min / Max Value Settings*. Выберите *Min / max* и кликните на кнопку *Apply*.

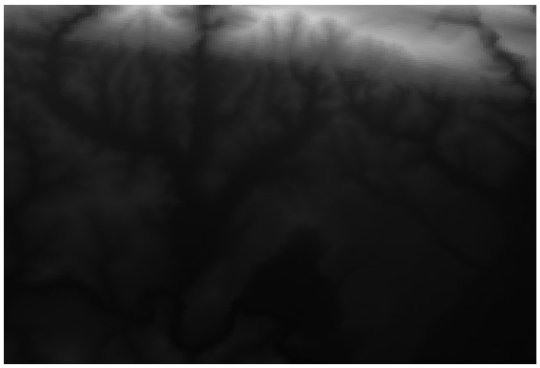
Помните, что гистограмма показывает вам распределение значений и не все значения обязательно должны отображаться на графике.

6.4.7  Идем дальше: Пространственная интерполяция

Предположим, что у вас набор точек выборки, на базе которого вы хотите экстраполировать данные. Например, у вас может быть доступ к набору данных *Sampled points*, который мы создали ранее, и вы хотели бы иметь некоторое представление о том, как выглядит местность.

1. Для начала вам надо запустить инструмент *GDAL* ► *Raster analysis* ► *Grid (IDW with nearest neighbor searching)*, который находится в *Processing Toolbox*.
2. В качестве *Point layer* выберите `Sampled points`.
3. Установите *Weighting power* на `5.0`.
4. Внутри *Advanced parameters*, установите *Z value from field* на `rvalue_1`.
5. И наконец, кликните на кнопку *Run* и подождите до окончания обработки.
6. Закройте диалоговое окно.

Вы сможете сравнить исходный набор данных (слева) с тем, который был построен на базе наших точек выборки (справа). Ваш набор данных может выглядеть иначе из-за случайного расположения точек.





Как вы можете видеть, 100 выборочных точек недостаточно, чтобы получить подробное представление о местности. Вы получите только очень общее представление, но оно может вводить в заблуждение.

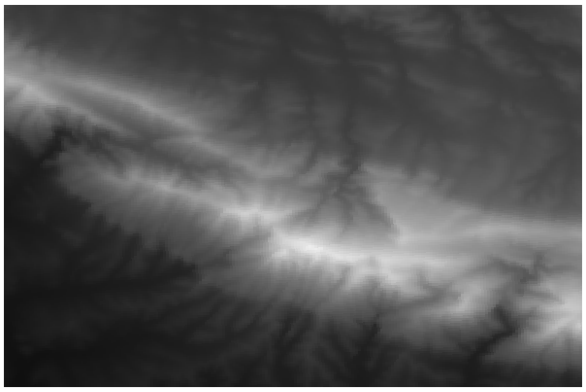
6.4.8  Попробуйте сами: Различные методы интерполяции

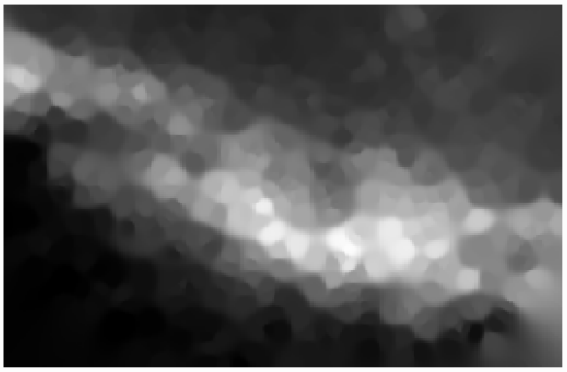
1. Следуйте описанному выше процессу для того, чтобы создать набор из 10 000 случайных точек.

   Примечание: Если точек действительно много, обработка может занять много времени.

2. Используйте эти точки для того сделать выборку исходной DEM / Цифровой модели рельефа.
3. Используйте инструмент *Grid (IDW with nearest neighbor searching)* в этом наборе данных.
4. Установите *Power* и *Smoothing* на `5.0` и `2.0`, соответственно.

Результаты (в зависимости от расположения ваших случайных точек) будут примерно такими:





Это более улучшенное представление ландшафта, так как плотность выборочных точек выше. Помните, что чем больше выборка, тем лучше результат.

6.4.9 В заключении

У QGIS есть ряд инструментов для анализа пространственных статистических свойств набора данных.

6.4.10 Что дальше?

Теперь, после того как мы рассмотрели векторный анализ, почему бы не посмотреть, что можно сделать с растрами? Этим мы и займемся в следующем модуле!

ГЛАВА 7

Модуль: Растры

Мы раньше использовали растры для оцифровки, но растровые данные можно также использовать напрямую. Вы узнаете как это делается в QGIS в этом модуле.

7.1 Занятие: Работа с растровыми данными

Растровые данные существенно отличаются от векторных данных. Векторные данные имеют дискретные характеристики с геометрией, которые построены из вершин и, скорее всего, связаны с линиями и/или местностями. Тем не менее растровые данные похожи на любое изображение. Хотя они могут отображать различные свойства объектов в реальном мире, эти объекты не существуют как отдельные объекты, а представлены с помощью пикселей с разными значениями.

В этом модуле вы будете использовать растровые данные как дополнение к существующему анализу GIS.

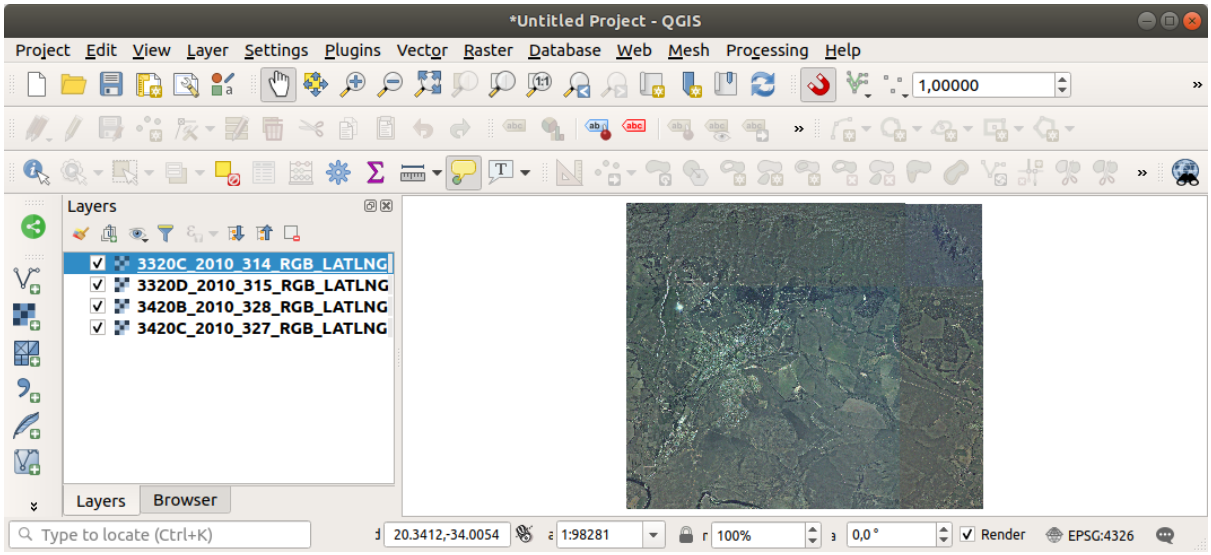
Цель этого урока: Научиться работать с растровыми данными в QGIS.

7.1.1  Идем дальше: Загружаем растровые данные

Растровые данные можно загрузить, используя те же методы, которые мы использовали для векторных данных. Тем не менее предлагается использовать панель *Browser*.

1. Откройте панель *Browser* и расширьте папку `exercise_data/raster`.
2. Загрузите все данные в эту папку:
   - `3320C_2010_314_RGB_LATLNG.tif`
   - `3320D_2010_315_RGB_LATLNG.tif`
   - `3420B_2010_328_RGB_LATLNG.tif`
   - `3420C_2010_327_RGB_LATLNG.tif`

Вы увидите следующую карту:



Вот и все - четыре аэрофотоснимка местности, которую мы будем изучать.

7.1.2 Идем дальше: Создаем виртуальный растр

Как вы видите теперь, ваш слой расположился на всех четырех изображениях. Значит вам все время придется работать с четырьмя растрами. Это конечно не идеальный вариант. Было бы хорошо работать лишь с одним файлом.

К счастью, QGIS дает вам возможность сделать именно это, без необходимости создавать новый растровый файл. Вы можете создать **виртуальный растр**, который также часто называют *каталогом*, что и объясняет его функцию. Это не совсем новый растр, а способ организации ваших существующих растров в одном каталоге: один файл для облегчения доступа.

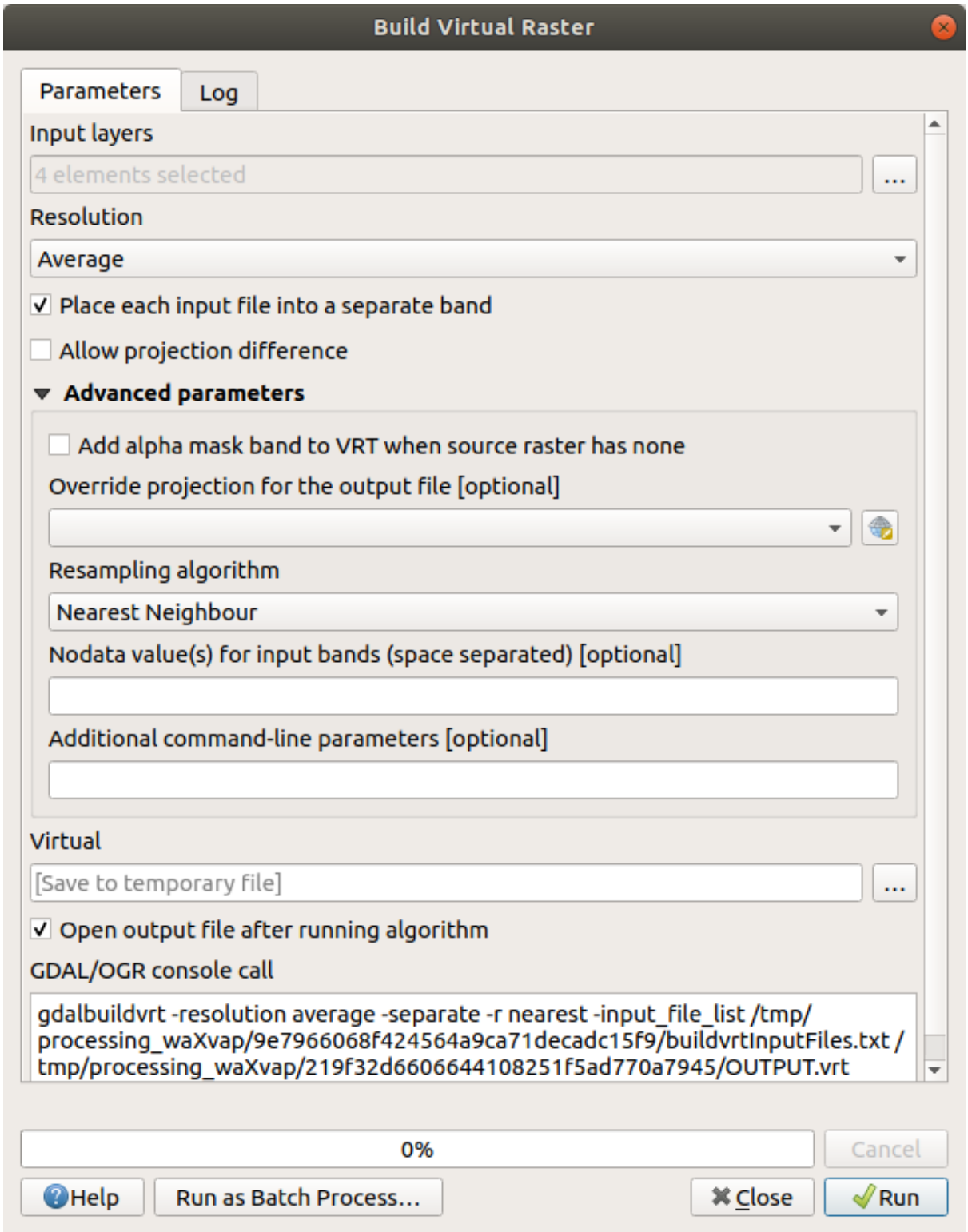
Для создания каталога мы использует *Processing* ► *Toolbox*.

1. Откройте алгоритм *Build virtual raster* из *GDAL* ► *Raster miscellaneous*;
2. В появившемся диалоговом окне кликните на кнопку … рядом с параметром *Input layers* и проверьте все слои или используйте кнопку *Select All*;
3. Снимите флажок с *Place each input file into a separate band*. Обратите внимание на текстовое поле ниже. Это диалоговое окно на самом деле пишет этот текст за вас. Это длинная команда, которую QGIS собирается запустить.

   Примечание: Помните, что вы сможете скопировать и вставить текст в `OSGeo Shell` (пользователь Windows) или `Terminal` (пользователь Linux и OSX) для того, чтобы запустить команду. Вы также можете создать запрос для каждой команды GDAL. Это очень удобно в тех случаях, когда процедура занимает много времени или когда вы хотите запланировать определенные задачи. Используйте кнопку *Help* для того, чтобы получить дополнительную справку по синтаксису команды.

4. И кликните на кнопку *Run*.

Примечание: Как известно из предыдущих модулей, *Processing* создает временные слои по умолчанию. Чтобы сохранить файл, кликните на кнопку ….



Теперь вы можете убрать четыре исходных растра из панели *Layers* и оставить только выходной растр виртуального каталога.

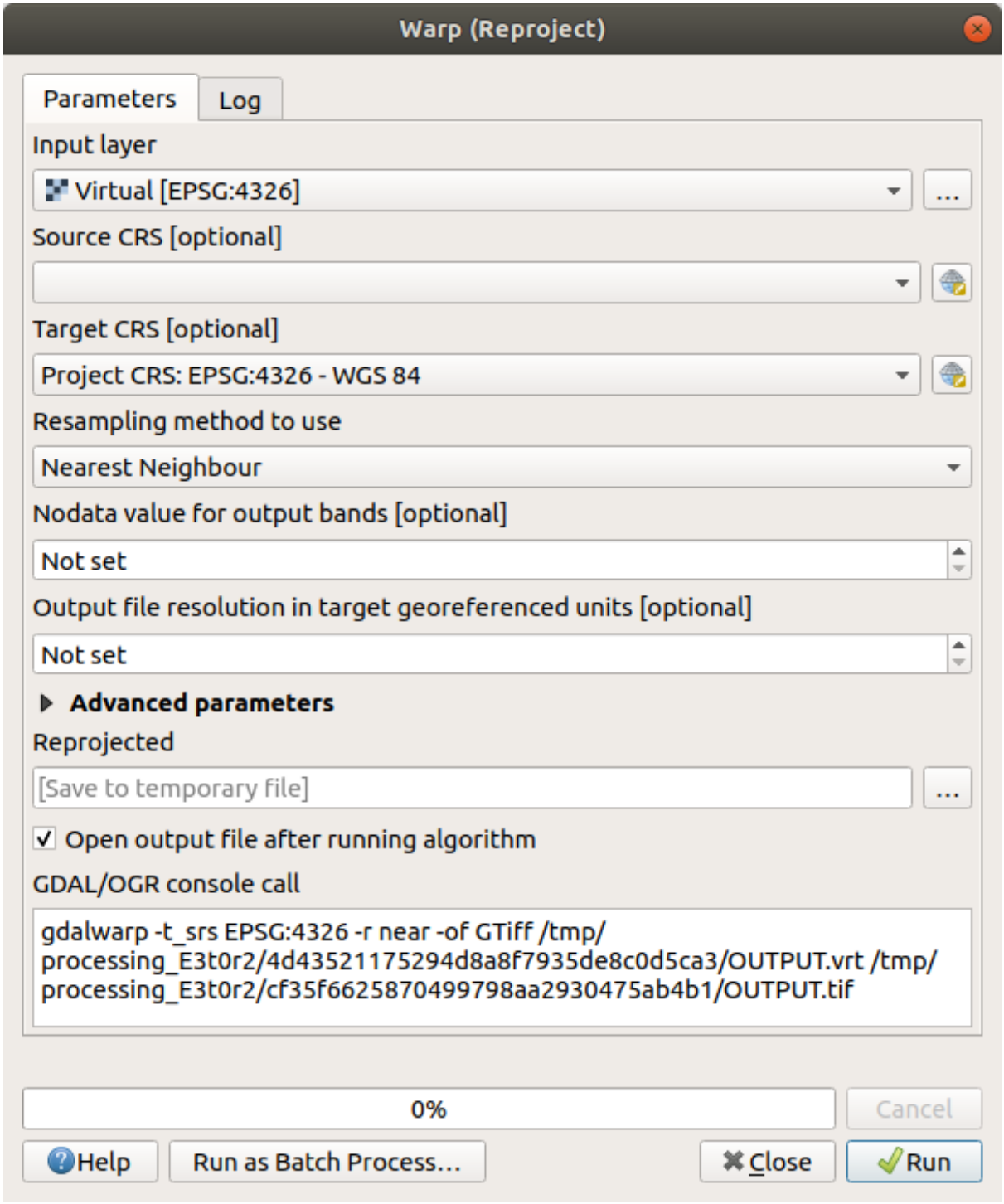
7.1.3  Преобразование растровых данных

Методы, упомянутые в выше, позволят виртуально объединять наборы данных с помощью каталога и пере-проецировать их «на лету». Тем не менее, если вы настраиваете данные, которые собираетесь использовать надолго, вам лучше создать новые растры, которые уже объединены и пере-проецированы, но для первоначальной настройки может потребоваться некоторое время.

Пере-проецирование растров

Надо открыть *Warp (reproject)* в *GDAL* ► *Raster projections*.

Вы также можете пере-проецировать виртуальные растры (каталоги), активировать многопоточную обработку и многое другое.



Объединение растров

Если вам надо создать новый растровый слой и сохранить его на диске, вы можете использовать алгоритм слияния.

Примечание: В зависимости от того, сколько растровых файлов вы объединяете и какие у них разрешения, новый растровый файл, который вы создадите, может быть довольно большого размера. Вам лучше вместо этого создать каталог растра, как описано в разделе *Создать виртуальный растр* .

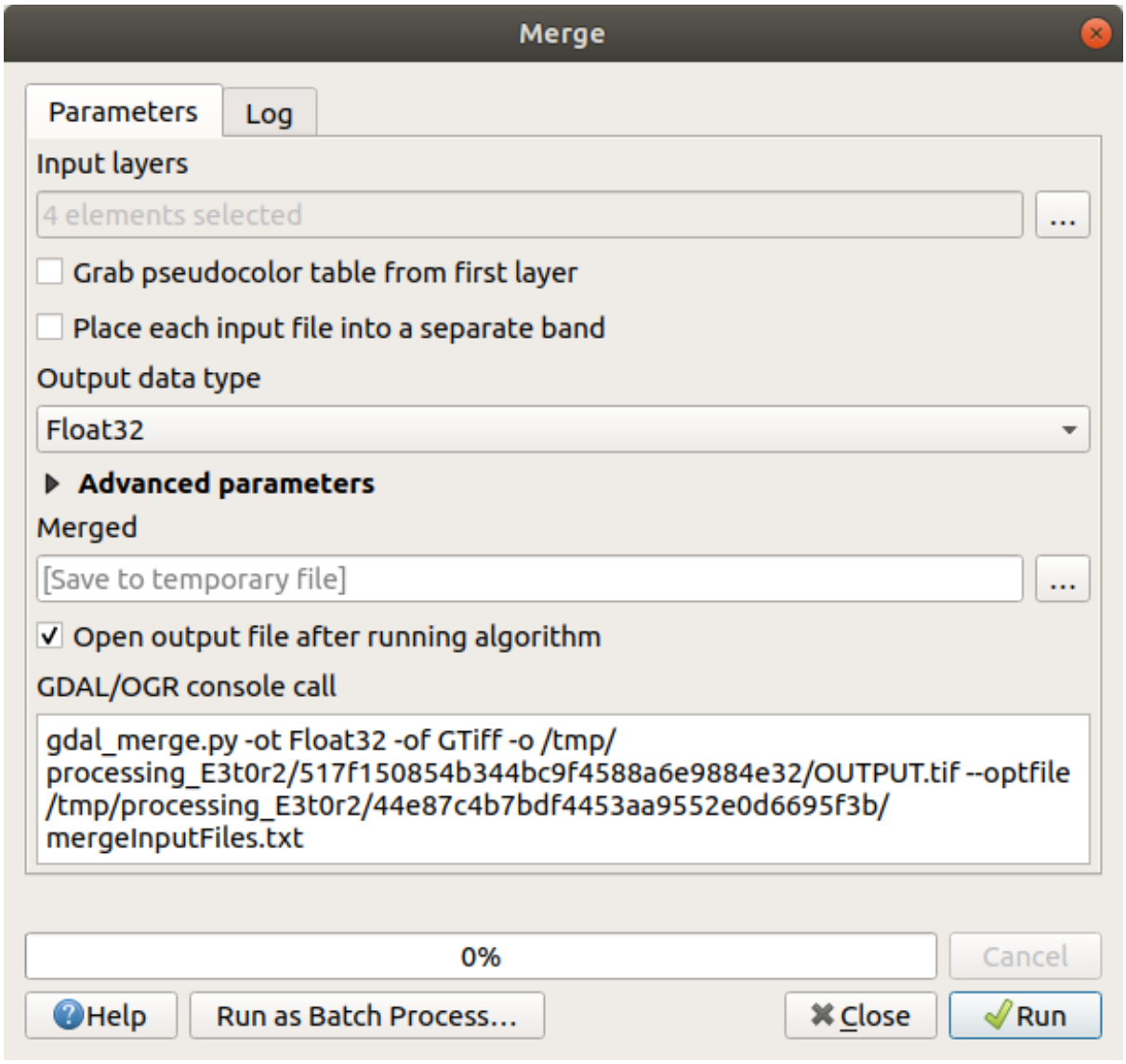
---

1. Вам надо кликнуть на алгоритм СлитьОбъединить *Merge* в меню *GDAL* ► *Raster miscellaneous*.

2. Как и в случае с *Созданием виртуального растра*, используйте кнопку … для того, чтобы выбрать, какие слои вы хотите объединить.

   Вы также можете указать виртуальный растр в качестве входных данных, и тогда все растры, из которых он состоит, будут обработаны.

3. Если вы знаете библиотеку GDAL, вы также можете добавить свои собственные опции, открыв меню *Advanced parameters*.



7.1.4 В заключении

QGIS облегчает включение растровых данных в ваши существующие проекты.

7.1.5 Что дальше?

Далее мы собираемся использовать растровые данные, которые не являются аэрофотоснимками, чтобы увидеть, как символы могут быть полезными и в случае с растрами.

7.2 Идем дальше: Измененяем символику растрового слоя

Не все растровые данные являются аэрофотоснимками. Есть много других форм растровых данных, и во многих таких случаях необходимо обозначить их символами, чтобы их можно было потом соответствующим образом видеть и использовать.

Цель этого урока: Изменение символов для растрового слоя

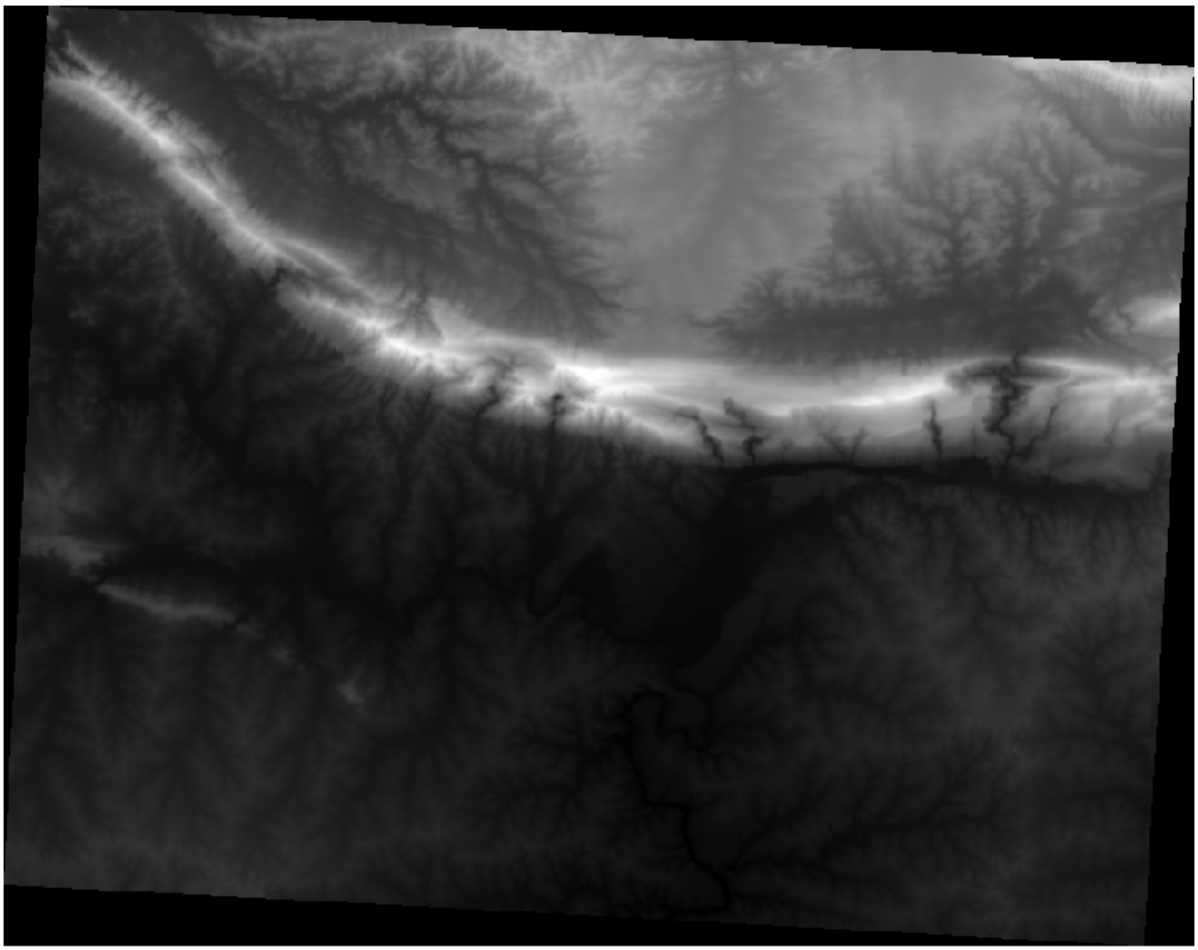
7.2.1  Попробуйте сами

1. Используйте панель *Browser* для загрузки `srtm_41_19.tif`, который находится внутри `exercise_data/raster/SRTM/`.
2. Увеличьте масштаб этого слоя, кликнув на него правой кнопкой мыши в *Layers panel* и выбрав *Zoom to Layer*.

Этот набор данных представляет собой *Цифровую Модель Рельефа (ЦМР) / Digital Elevation Model (DEM)*. Это карта возвышенности (высота) местности, которая позволяет нам, к примеру, увидеть, где находятся горы и долины.

В то время как каждый пиксель набора данных предыдущего раздела содержал сведения о цвете, в *ЦРМ*, каждый пиксель содержит значения высоты.

Как загрузки ЦМР вы заметите, что картина представляется в оттенках серого:



QGIS автоматически растянул значения пикселей изображения для целей визуализации. Мы более подробно изучим как это работает, по мере продвижения дальше.

7.2.2 Идем дальше: Меняем символику растрового слоя

Есть два способа изменения символов растра:

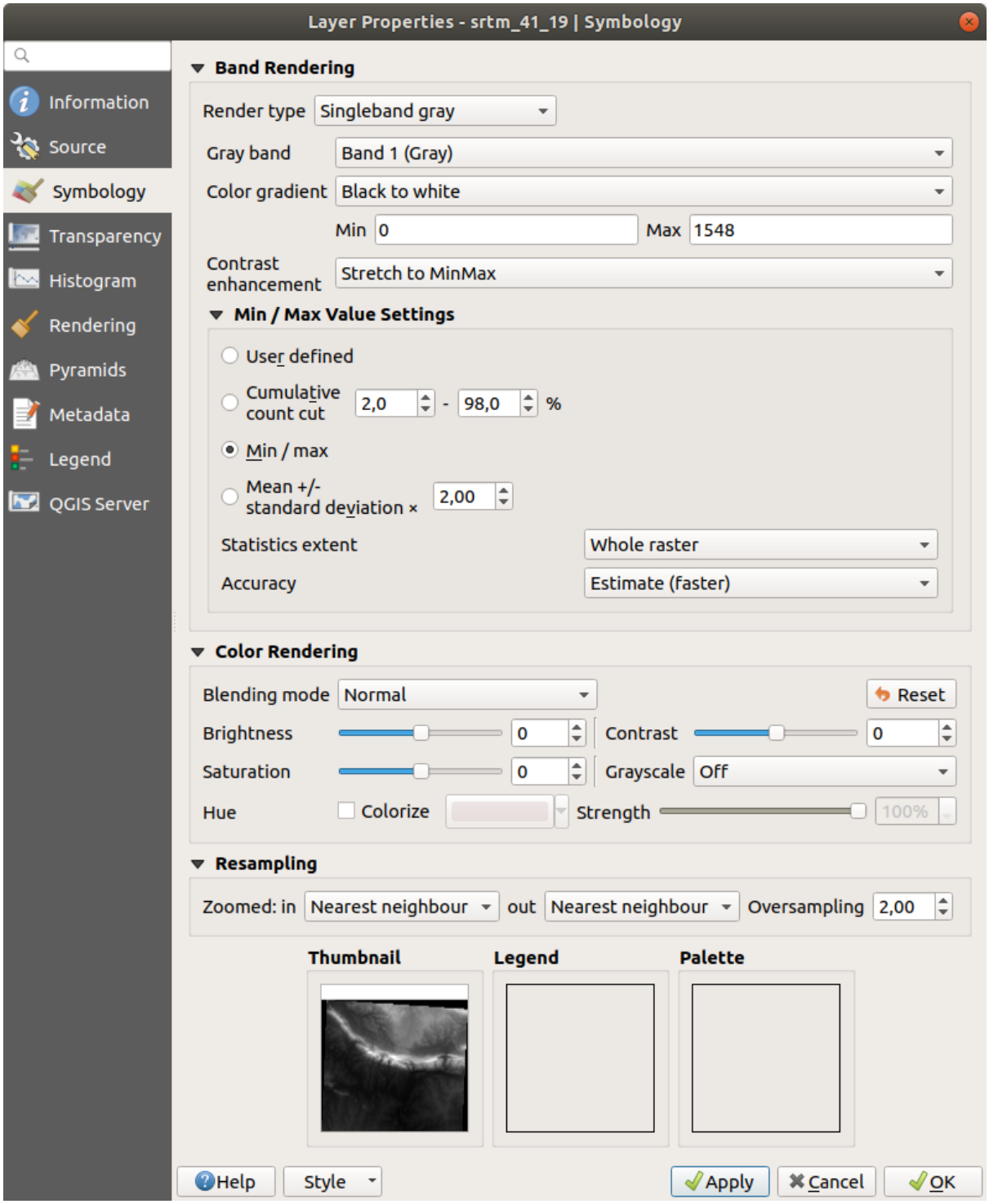
1. Внутри диалогового окна *Layer Properties* - кликнув правой кнопкой мыши на внутри древа слоев и выбрав опцию *Properties*. Затем надо переключиться на вкладку *Symbology*.
2. Надо кликнуть на кнопку Open the Layer Styling panel прямо над панелью *Layers* (ярлык `F7`). Таким образом откроется панель *Layer Styling*, где вы можете переключиться на вкладку Symbology.

Выберите метод, которым вы захотите использовать.

7.2.3 Идем дальше: Однополосная серая

Загруженный растровый файл не является фотоизображением подобно тем, которые были в предыдущем разделе. По умолчанию это градиент в оттенках серого.

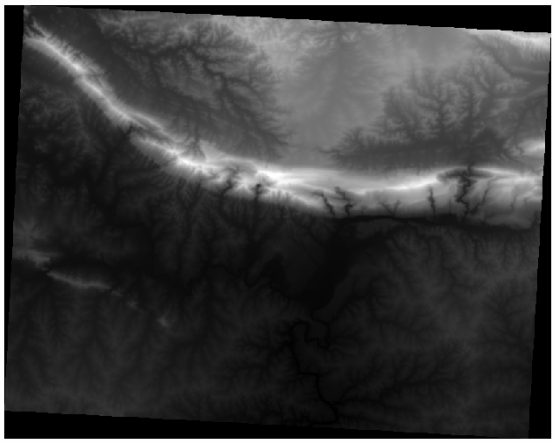
Давайте изучим некоторые особенности данного средства визуализации.



По умолчанию *Color gradient* настроен на `Black to white`, и это означает, что низкие значения пикселей являются черными, а высокие значения - белыми. Вам надо попробовать изменить этот параметр на `White to black` и посмотреть что получится.

Надо отметить важность параметра *Contrast enhancement* parameter: по умолчанию он настроен на `Stretch to MinMax`, и это означает, что значения пикселей растянуты до минимальных и максимальных значений.

Посмотрите на отличия с улучшением (слева) и без (справа):





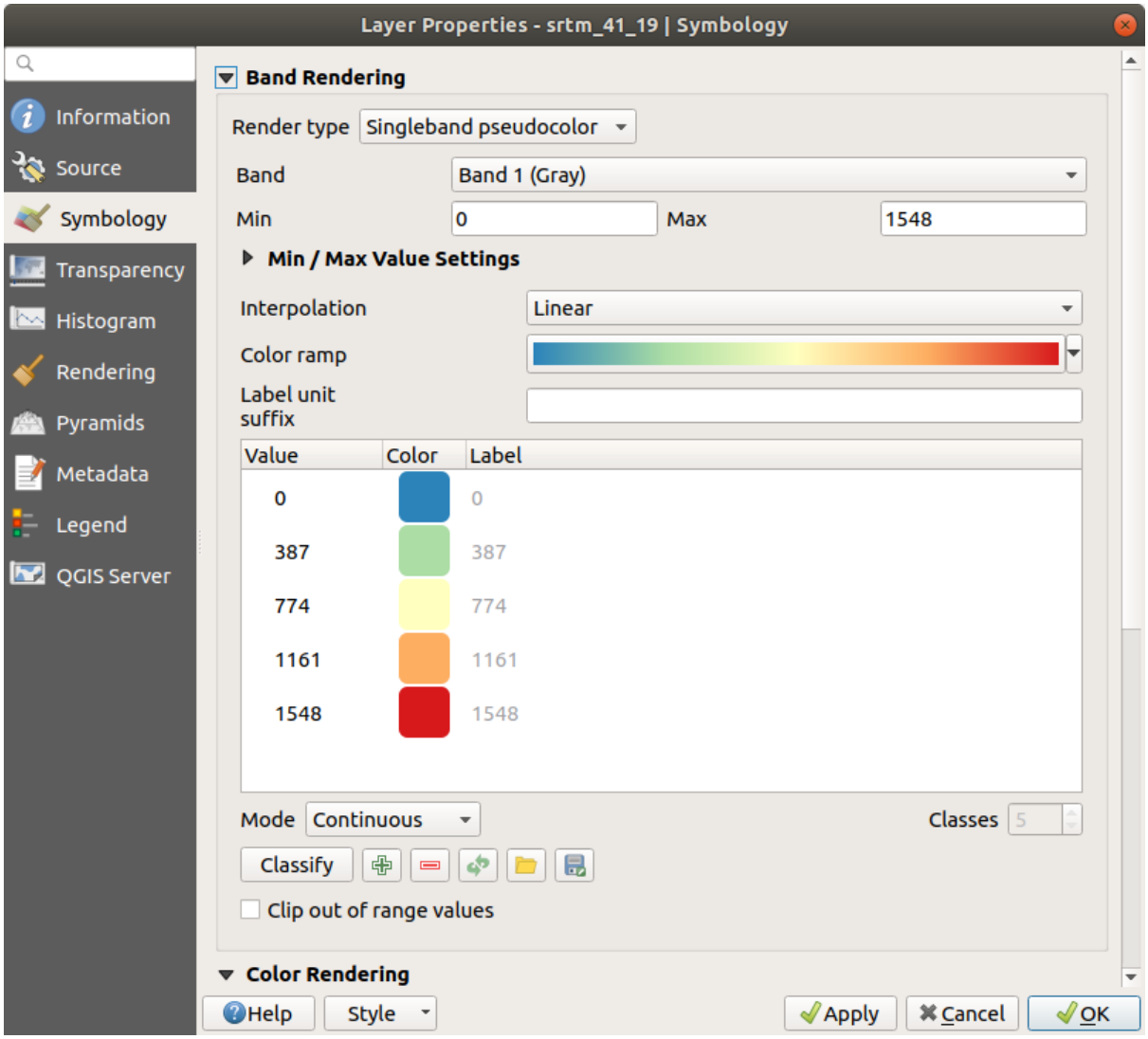
Но каковы минимальные и максимальные значения, которые следует использовать для растяжки? Те, которые в настоящее время охвачены *Min / Max Value Settings*. Есть много способов вычисления минимальных и максимальных значений и использования их для растяжки:

1. **Определяемые самим пользователем**: вам надо вручную внести значения *Min* и *Max*.
2. **Срез кумулятивного подсчета**: этот способ хорош, когда у вас имеются экстремально низкие или высокие значения. Этот способ *срезает* `2%` (или выбранное вами значение) этих значений.
3. **Мин / Макс**: *Реальные* или *Расчетные* минимальное и максимальное значения растра.
4. **Среднее +/- стандартное отклонение**: значения рассчитываются в соответствии со средним значением и стандартным отклонением.

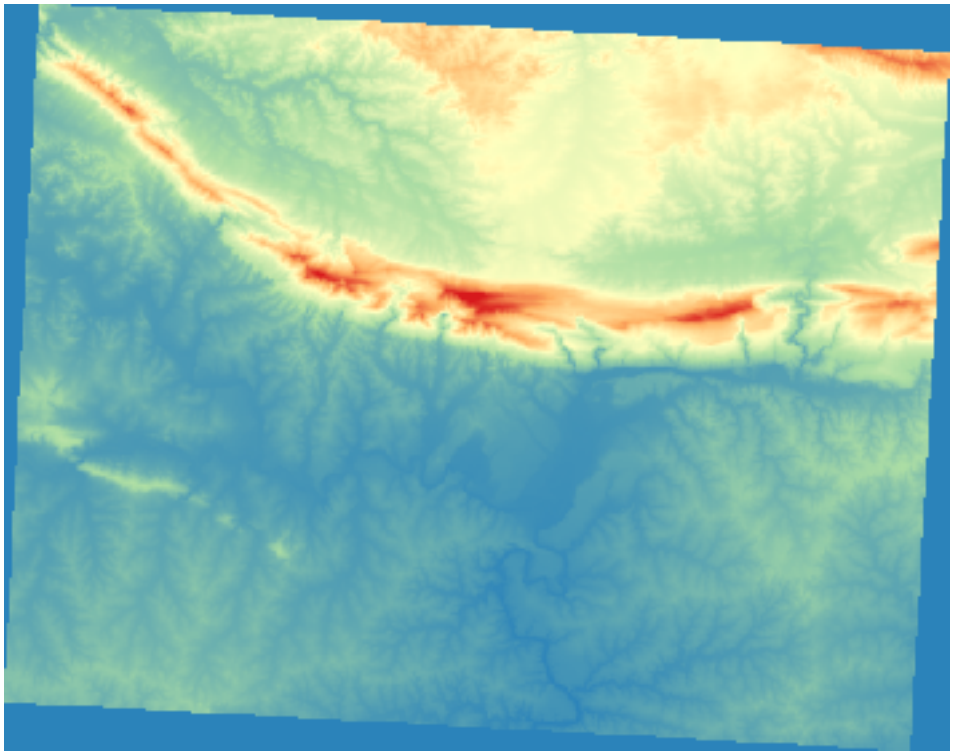
7.2.4  Идем дальше: Однополосная псевдо-цветная

Оттенки серого не всегда подходят для растровых слоев. Попробуем сделать ЦМР более красочной.

- Надо изменить *Render type* на *Singleband pseudocolor*. Если вам не нравятся загруженные цвета по умолчанию, вы можете выбрать другой *Color ramp*.
- Кликните на кнопку *Classify* для получения новой классификации цветов.
- Если цвета автоматически не сгенерированы, кликните на кнопку *OK* для того, чтобы применить эту классификацию по отношению к ЦМР.



Вы увидите, что ваш растр будет выглядеть так:

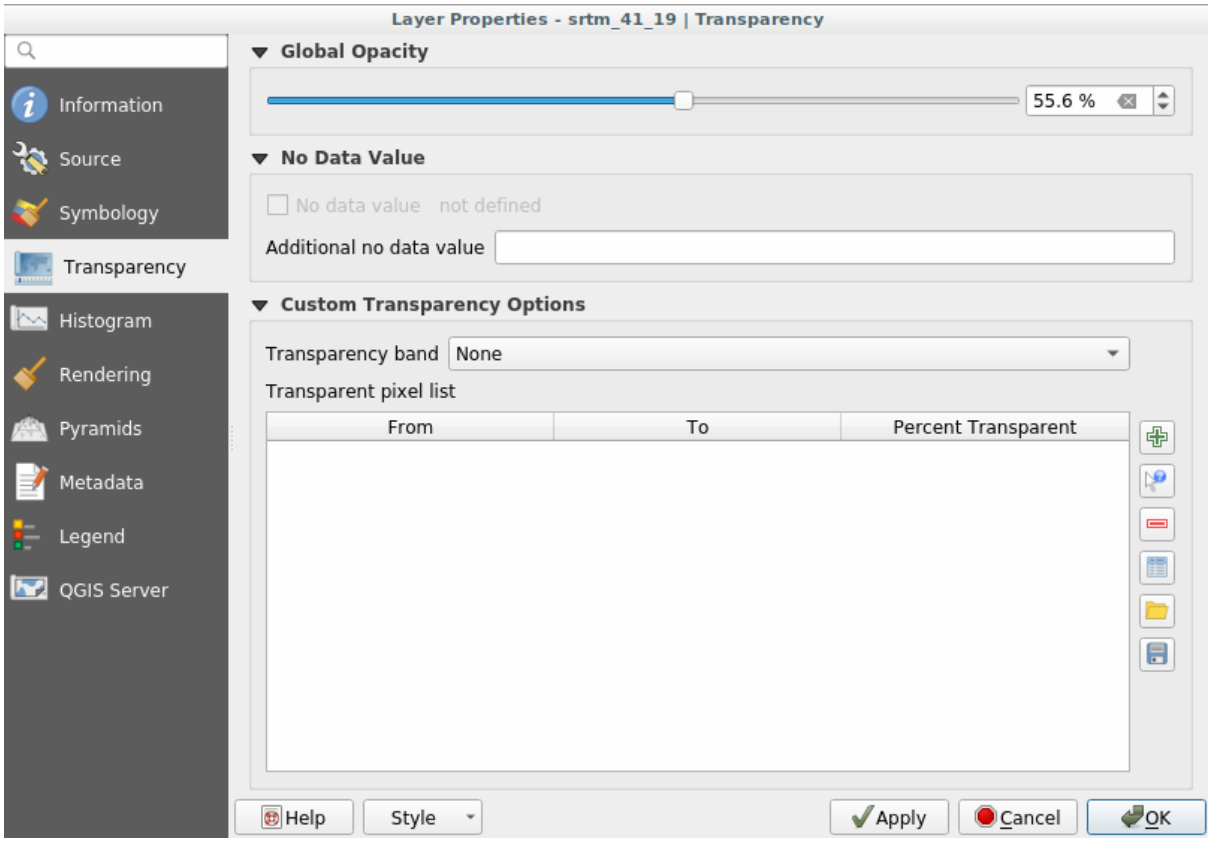


Это интересный способ представления ЦМР. Вы видите теперь, что значения растра снова отображены соответствующим образом: от синего цвета, который показывает низменные местности, до красного цвета – места, расположенные на более высоких точках.

7.2.5 Идем дальше: Меняем прозрачность

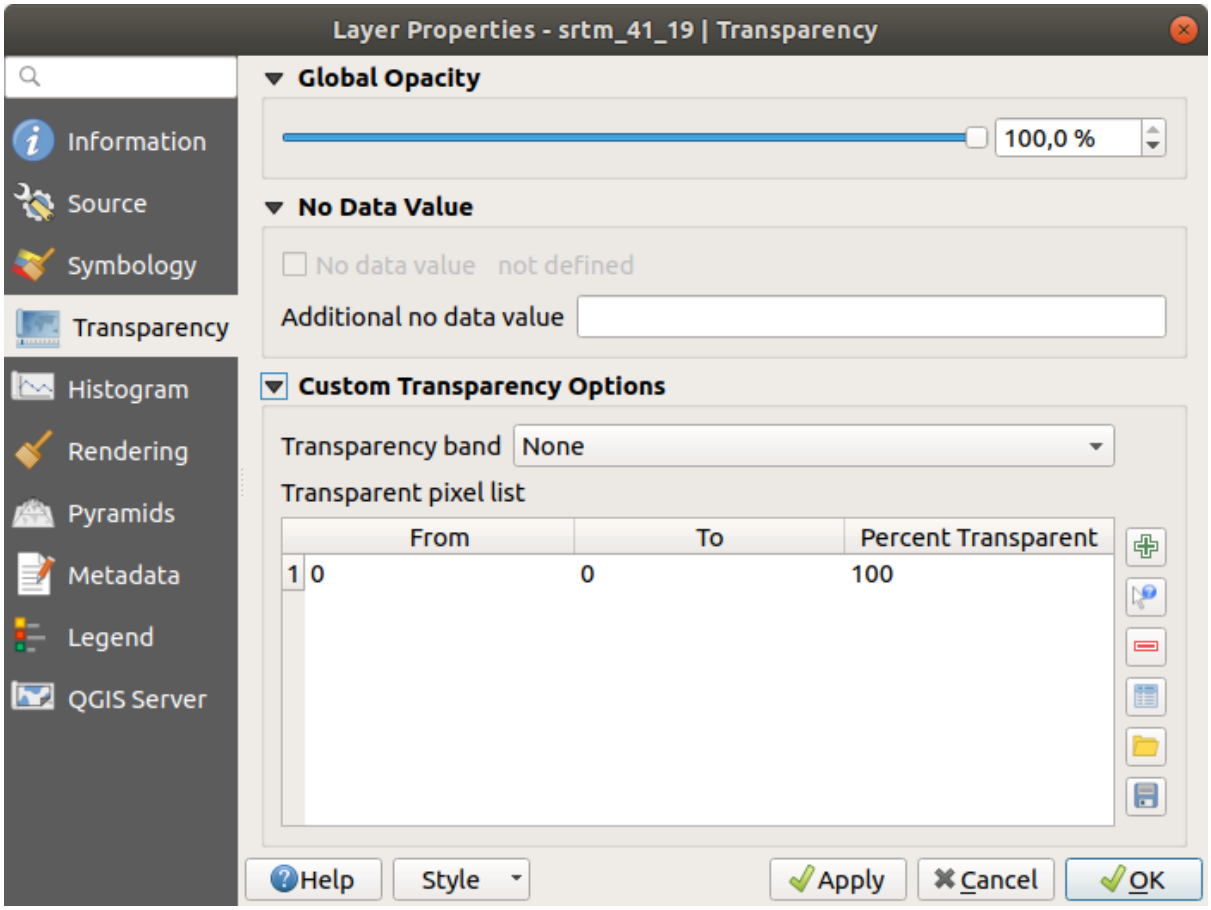
Изменение прозрачности всего растрового слоя иногда позволяет вам увидеть другие слои, покрытые самим растром, и лучше понять исследуемую местность.

Чтобы изменить прозрачность всего растра, надо перейти на вкладку *Transparency* и использовать ползунок *Global Opacity*, для того чтобы уменьшить непрозрачность:

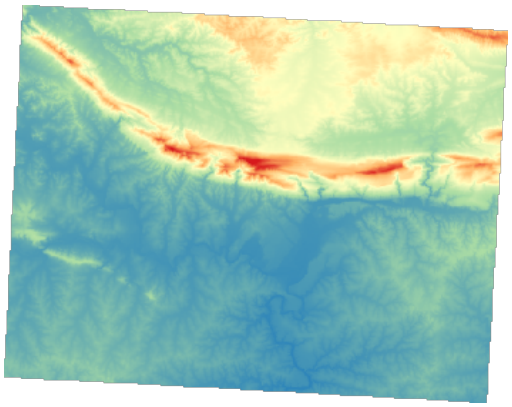


Изменение прозрачности для некоторых значений пикселей является также более интересным способом. Например, в используемом нами растре вы можете увидеть однородный цвет по углам. Чтобы настроить эти пиксели как прозрачные вам надо перейти на *Custom Transparency Options* во вкладке *Transparency*.

- Нажав на кнопку Add values manually, вы сможете добавить набор значений и настроить их процент прозрачности.
- Что касается единичных значений лучше использовать кнопку Add values from display.
- Надо кликнуть на кнопку Add values from display. Диалоговое окно исчезнет и вы сможете взаимодействовать с картой.
- Надо кликните на однородный цвет в углу ЦМР.
- Вы увидите, что таблица прозрачности будет заполнена выбранными значениями:



- Надо кликнуть на кнопку *OK* для того, чтобы закрыть диалоговое окно и увидеть изменения.



Видите? Углы теперь прозрачны на 100%.

7.2.6 В заключении

Мы рассмотрели некоторые основные функции, которые помогут вам начать работу с растровыми символами. QGIS также может предложить вам множество других опций, таких как обозначение слоя с помощью палитры/уникальных значений, представление различных полос разными цветами на мультиспектральном изображении или создание автоматического эффекта оттенка холмов (только с растровыми файлами ЦМР).

7.2.7 Ссылки

Набор данных SRTM был получен из `http://srtm.csi.cgiar.org/ `_

7.2.8 Что дальше?

Теперь мы можем видеть, что наши данные отображаются соответствующим образом. Давайте изучим, как мы сможем проанализировать их дальше.

7.3 Занятие: Анализ рельефа местности

Отдельные виды растров позволяют вам лучше изучить ландшафт, который они представляют. Цифровые Модели Рельефа (ЦМР)/ DEM особенно полезны в этом отношении. В этом уроке вы будете использовать инструменты анализа местности, чтобы узнать больше об изучаемой местности для предлагаемой жилой застройки, о которой говорилось ранее.

Цель этого урока: Использовать инструменты анализа местности для того, чтобы получить дополнительные сведения о местности.

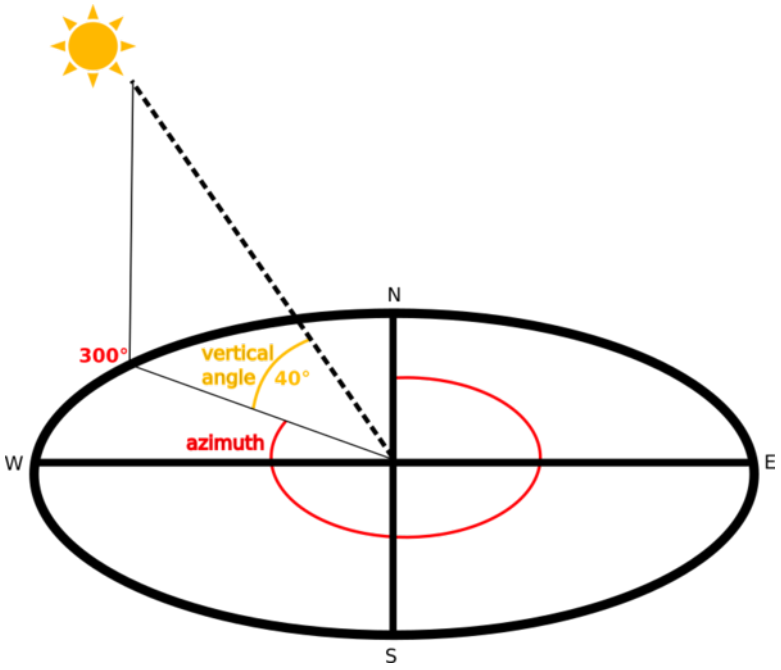
7.3.1  Идем дальше: Вычисляем отмывки

Мы будем использовать все тот же слой ЦМР, который был использован в предыдущем уроке. Если вы начинаете этот раздел с нуля, вам надо использовать панель *Browser* и загрузить `raster/SRTM/srtm_41_19.tif`.

Слой ЦМР показывает вам высоту местности, но иногда это может показаться немного абстрактным. Он содержит всю необходимую вам трехмерную информацию о местности, но она не выглядит похожей на трехмерный объект. Для того, чтобы получить лучшее представление о местности, можно вычислить *отмывку* / Hillshade, которая представляет собой растр, отображающий местность с использованием света и тени для создания трехмерного изображения.

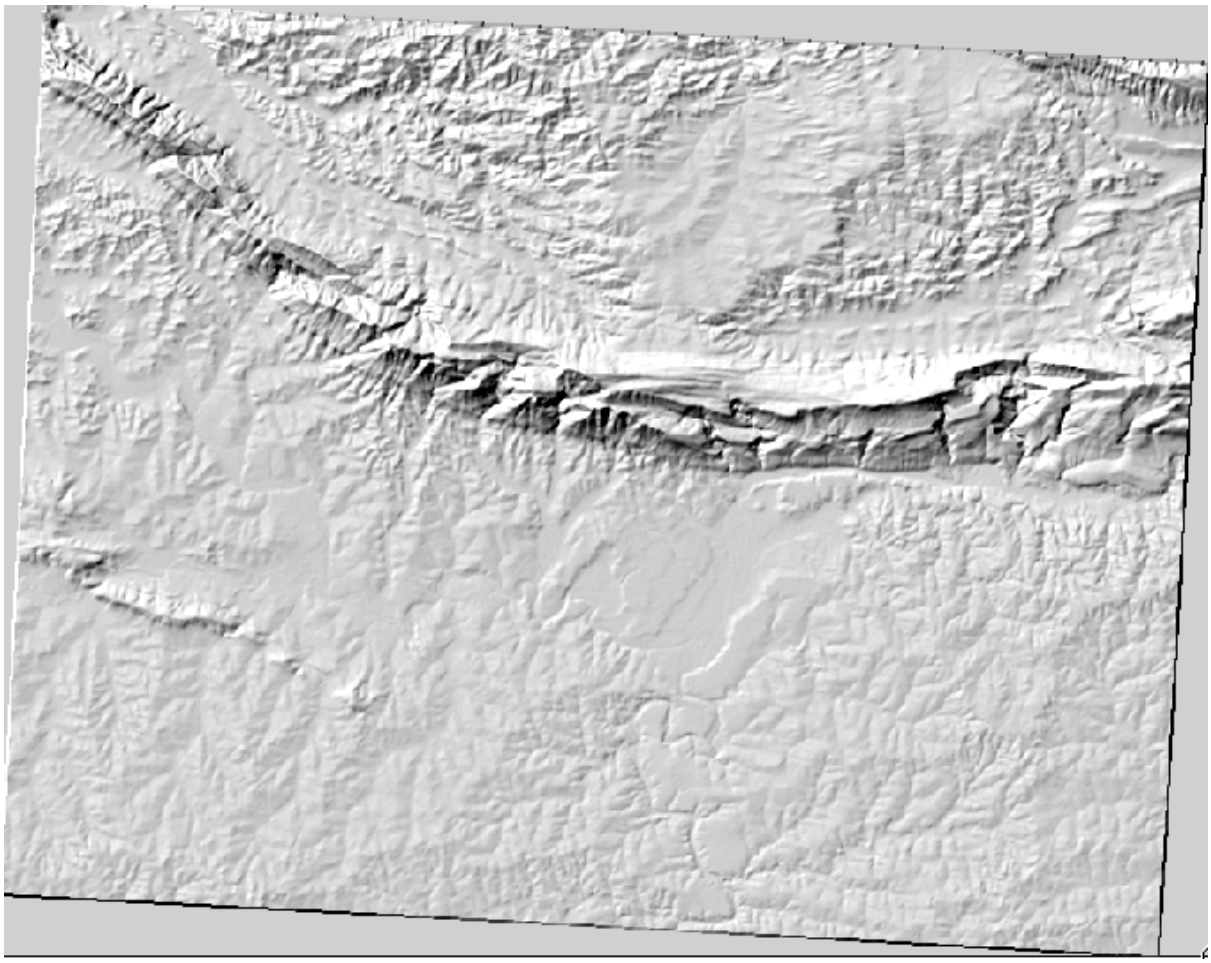
Нам надо использовать алгоритмы в меню *Raster* ► *Raster terrain analysis*.

1. Надо кликнуть на меню *Hillshade*.
2. Алгоритм позволит вам указать положение источника света: *Azimuth* имеет значения от 0 (север) до 90 (восток), 180 (юг) и 270 (запад), а *Vertical angle* настраивает высоту источника света (от 0 до 90 градусов).
3. Нам надо использовать следующие значения:
   - *Z factor* на `1.0`
   - *Azimuth (horizontal angle)* на `300.0°`
   - *Vertical angle* на `40.0°`



4. Сохраните файл в новой папке `exercise_data/raster_analysis/` с названием `hillshade`.
5. И кликните на кнопку *Run*

У вас будет новый слой под названием *hillshade*, который выглядит так:



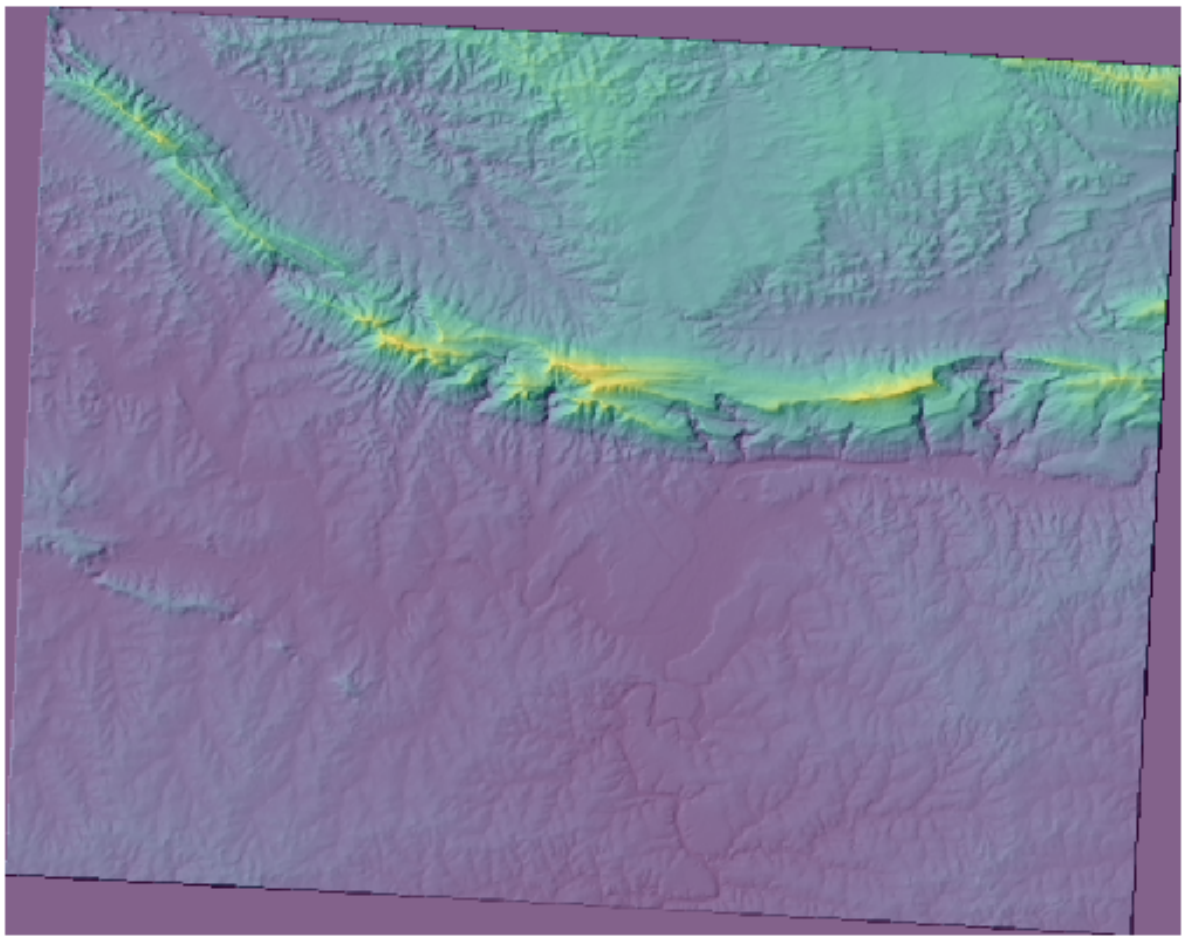
Красиво выглядит и трехмерно, но можем ли мы улучшить картинку? Сама по себе отмывка выглядит как гипсовая повязка. Разве мы не можем как-то использовать его вместе с другими, более красочными растрами? Конечно, можем, используя отмывку в качестве наложения.

7.3.2  Идем дальше: Используем отмывки в качестве наложения (оверлей)

Отмывка может предоставить очень полезную информацию о солнечном свете в определенное время суток. Но отмывку также можно использовать в эстетических целях для того, чтобы карта выглядела лучше. Это можно сделать, настроив отмывку так, чтобы она была в основном прозрачной.

1. Надо изменить символы исходного слоя *srtm_41_19*, чтобы использовать схему *Pseudocolor* как в предыдущем упражнении.
2. Надо скрыть все слои, за исключением слоев *srtm_41_19* и слоев *hillshade*.
3. Кликните на *srtm_41_19* и перетащите его и поместите под слой *hillshade* в панели *Layers*.
4. Настроите слой *hillshade* на прозрачность, кликнув по вкладке *Transparency* в свойствах слоя.
5. Настройте *Global opacity* на `50%`.

   Вы получите такой результат:



6. Отключите слой *hillshade* и включите снова в панели *Layers*, чтобы увидеть разницу.

Используя отмывку таким образом, вы сможете улучшить топографию ландшафта. Если эффект кажется вам недостаточно сильным, вы можете изменить прозрачность слоя *hillshade*; но, конечно, чем ярче становится отмывка, тем тусклее будут цвета за ней. Вам надо найти баланс, который вас устроит.

Не забудьте сохранить проект, когда закончите.

7.3.3 Идем дальше: Поиск наилучших мест

Вспомните о проблеме агента по недвижимости, которую мы в последний раз рассматривали на уроке по *векторному анализу*. Предположим, что покупатели теперь хотят приобрести здание и построить на его территории небольшой коттедж. Мы знаем, что в Южном полушарии идеальным местом для этого должен быть участок, который:

- смотрит на север,
- с уклоном менее 5 градусов,
- но если уклон меньше 2 градусов, то аспект не имеет значения.

Давайте найдем лучшее место для них.

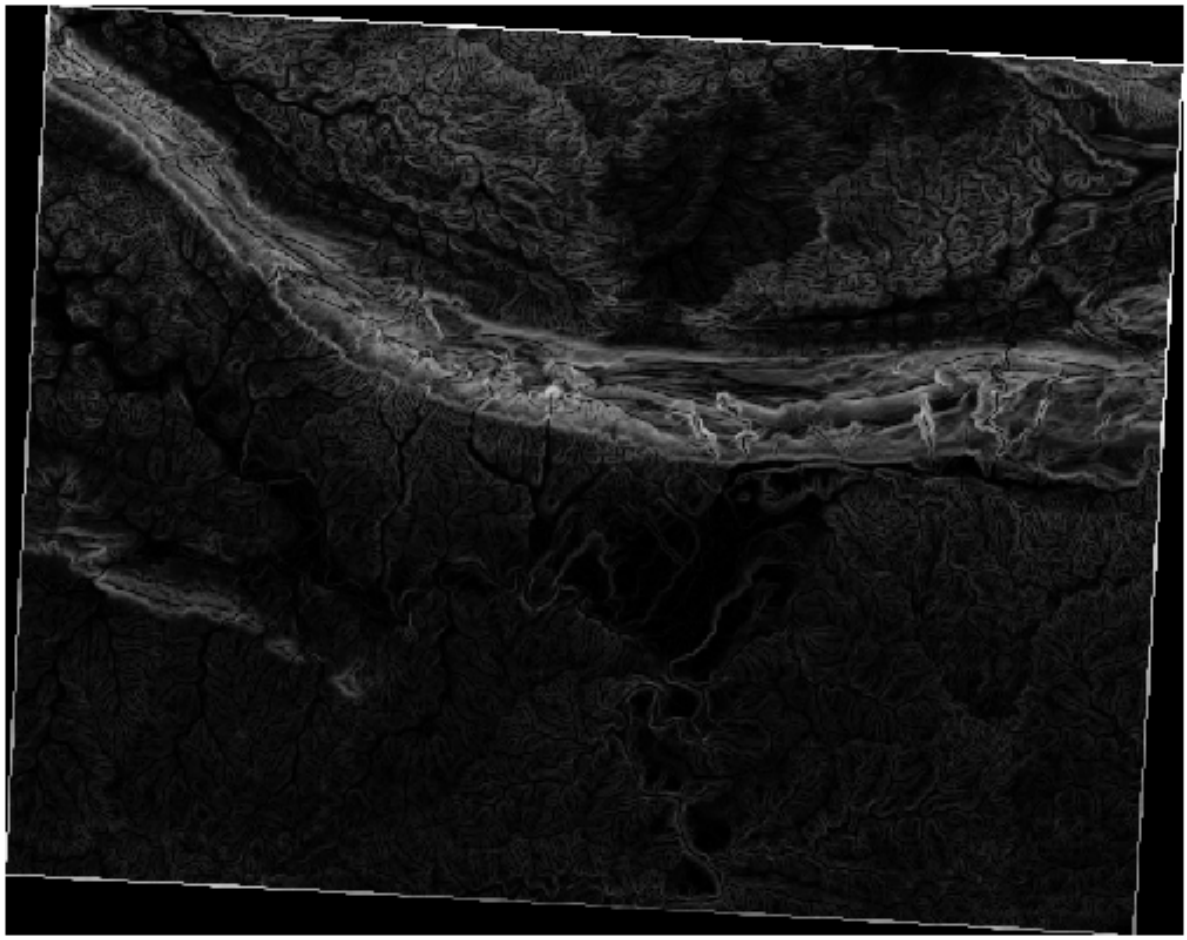
7.3.4  Идем дальше: Вычисляем уклон

Уклон показывает, уровень отвесности рельефа. Если, например, вы хотите построить на этой земле дома, то вам нужна относительно ровная земля.

Чтобы вычислить уклон, вам надо использовать алгоритм *Slope* в *Processing* ► *Raster terrain analysis*.

1. Вам надо открыть алгоритм.
2. Выберите *srtm_41_19* в качестве *Elevation layer*.
3. Установите *Z factor* на `1.0`.
4. Сохраните выход как файл с названием `slope` в той же папке, что и `hillshade`.
5. Кликните на кнопку *Run*.

Теперь вы увидите уклон рельефа. Каждый пиксель показывает соответствующее значение уклона. Черные пиксели показывают ровный рельеф, а белые – крутой рельеф:



7.3.5  Попробуйте сами вычислить аспект

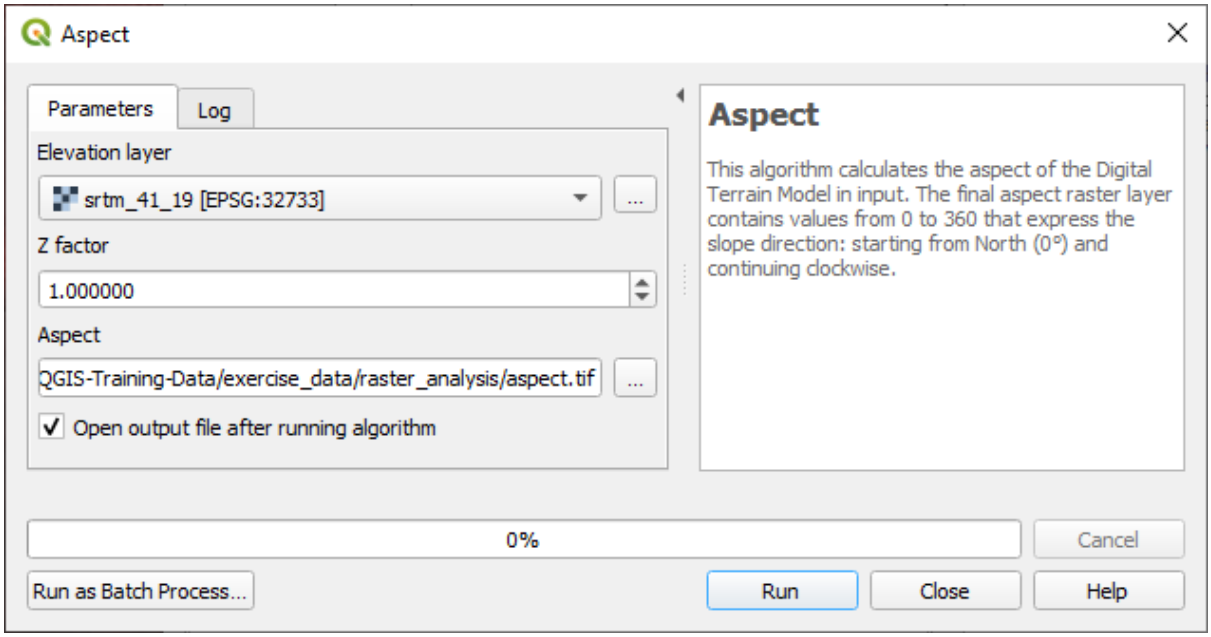
Аспект - направление по компасу, к которому обращен уклон местности. Значение аспекта 0 означает, что уклон обращен на север, 90 на восток, 180 на юг и 270 на запад.

Поскольку это исследование проводится в Южном полушарии, дома должны быть построены на северном уклоне, чтобы они могли оставаться под лучами солнца.

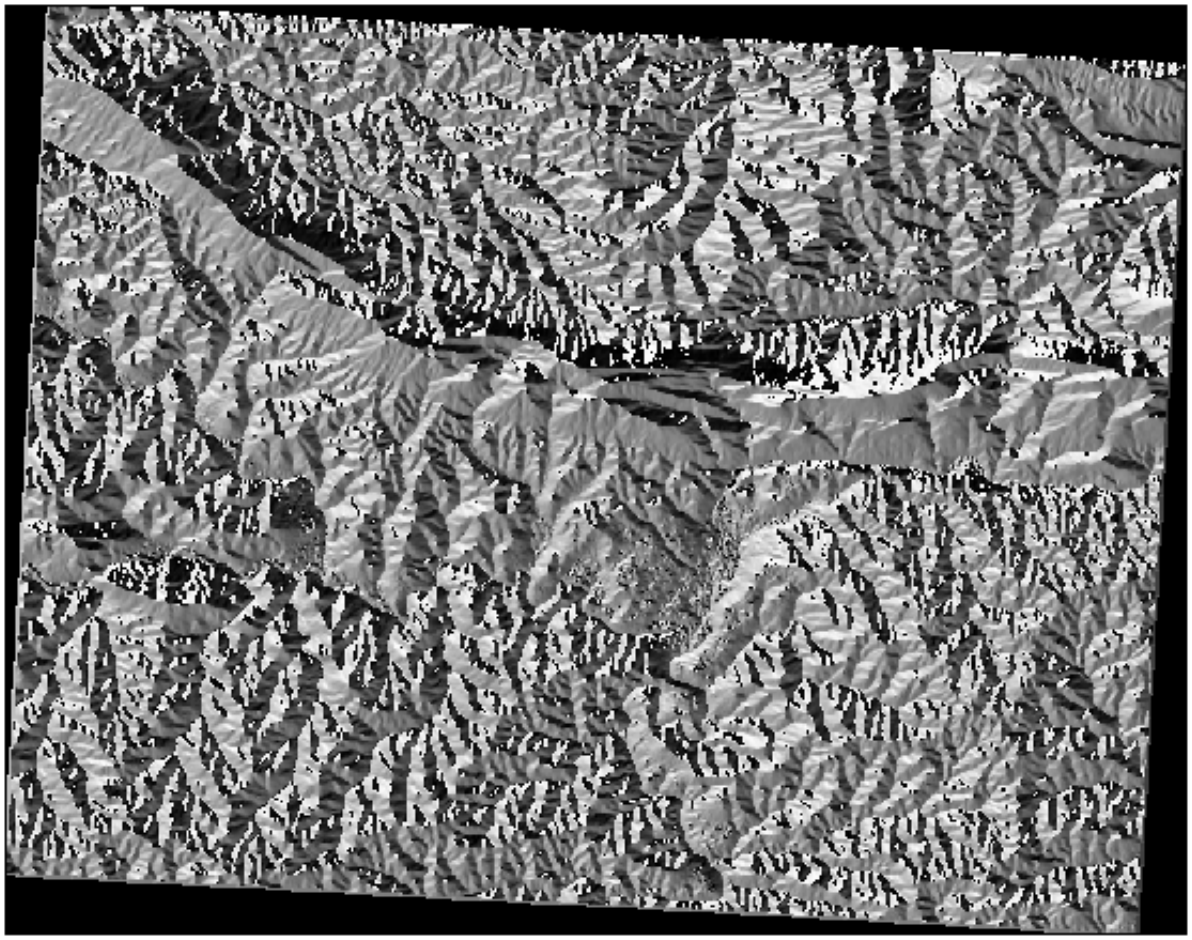
Вам надо использовать алгоритм *Aspect* в *Processing* ► *Raster terrain analysis* для того, чтобы сохранить слой `aspect` вместе с `slope`.

Answer

Set your *Aspect* dialog up like this:


Aspect
Parameters
Log
Elevation layer
srtm_41_19 [EPSG:32733]
Z factor
1.000000
Aspect
QGIS-Training-Data/exercise_data/raster_analysis/aspect.tif
Open output file after running algorithm
Aspect
This algorithm calculates the aspect of the Digital Terrain Model in input. The final aspect raster layer contains values from 0 to 360 that express the slope direction: starting from North (0°) and continuing clockwise.
0%
Cancel
Run as Batch Process...
Run
Close
Help

Your result:



7.3.6  Идем дальше: Находим северную сторону

У вас теперь есть растры, которые показывают уклон, а также аспект, но вы пока не знаете, каковы идеальные условия. Как можно было бы провести этот анализ?

Ответ сидит внутри *Raster calculator*.

У QGIS есть различные калькуляторы растра:

- *Raster* ► *Raster Calculator*
- В процессе:
  - *Raster Analysis* ► *Raster calculator*
  - *GDAL* ► *Raster miscellaneous* ► *Raster calculator*
  - *SAGA* ► *Raster calculus* ► *Raster calculator*

Каждый инструмент дает один и тот же результат, но синтаксис может немного отличаться, а наличие операторов может отличаться.

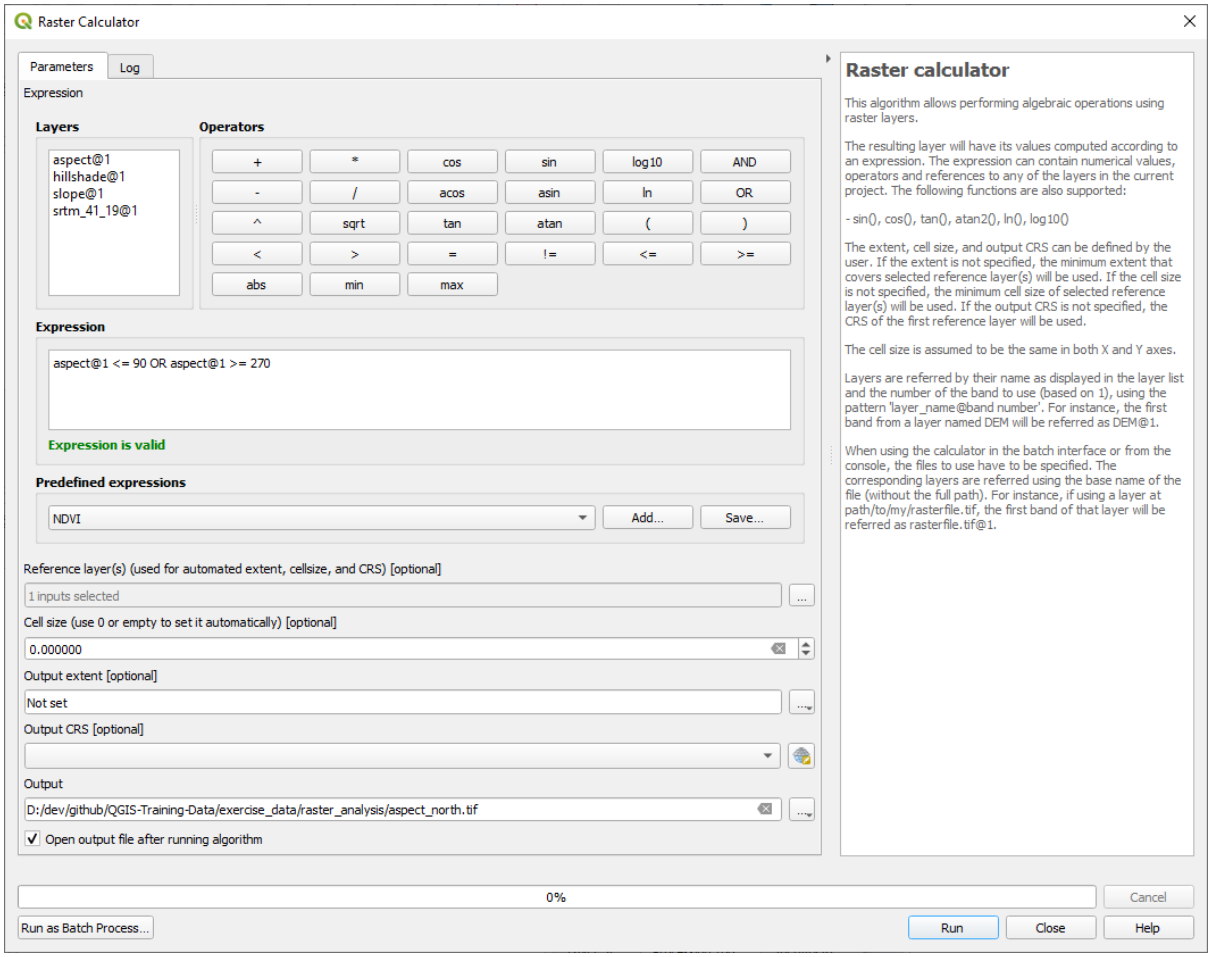
Нам надо использовать *Raster Analysis* ► *Raster calculator* в *панели инструментов обработки*.

1. Надо открыть инструмент, дважды кликнув по нему:
   - В верхней левой части диалогового окна перечислены все загруженные растровые слои как `name@N`, где `name` это название слоя и `N` это группа.
   - В правой верхней части вы увидите много разных операторов. На мгновение подумайте и представьте, что растр - это изображение. Вы должны представить его как 2D-матрицу, заполненную числами.
2. Север находится под углом 0 (ноль) градусов, поэтому, чтобы местность смотрела на север, его аспект должен быть больше 270 градусов или меньше 90 градусов. Поэтому, формула такова:

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

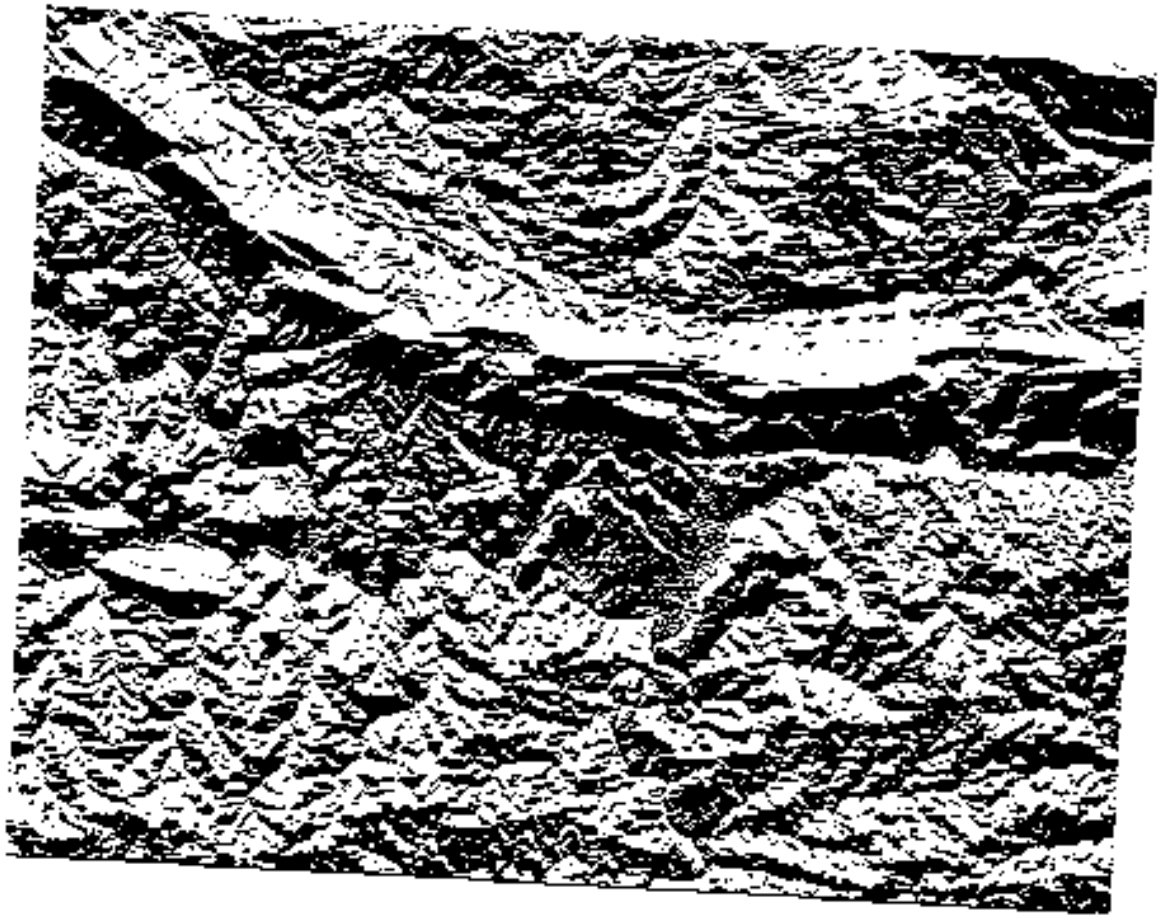
3. Теперь вам нужно настроить детали растра, например, размер ячейки, масштаб и ССК. Можно вручную или автоматически, выбрав `Reference layer`. Выберите эту последнюю опцию, нажав на кнопку … рядом с параметром *Reference layer(s)*.
4. В диалоговом окне надо выбрать слой *aspect* так как мы хотим получить слой с таким же разрешением.
5. Сохраните слой как `aspect_north`.

   Диалоговой окно должно выглядеть так:



6. И наконец надо кликнуть на кнопку *Run*.

Вы получите такой результат:



Выходные значения `0` или же `1`. Что это означает? Для каждого пикселя в растре написанная нами формула, показывает, соответствует ли она условиям или нет. Поэтому окончательный результат будет *Неверно* (0) и *Верно* (1).

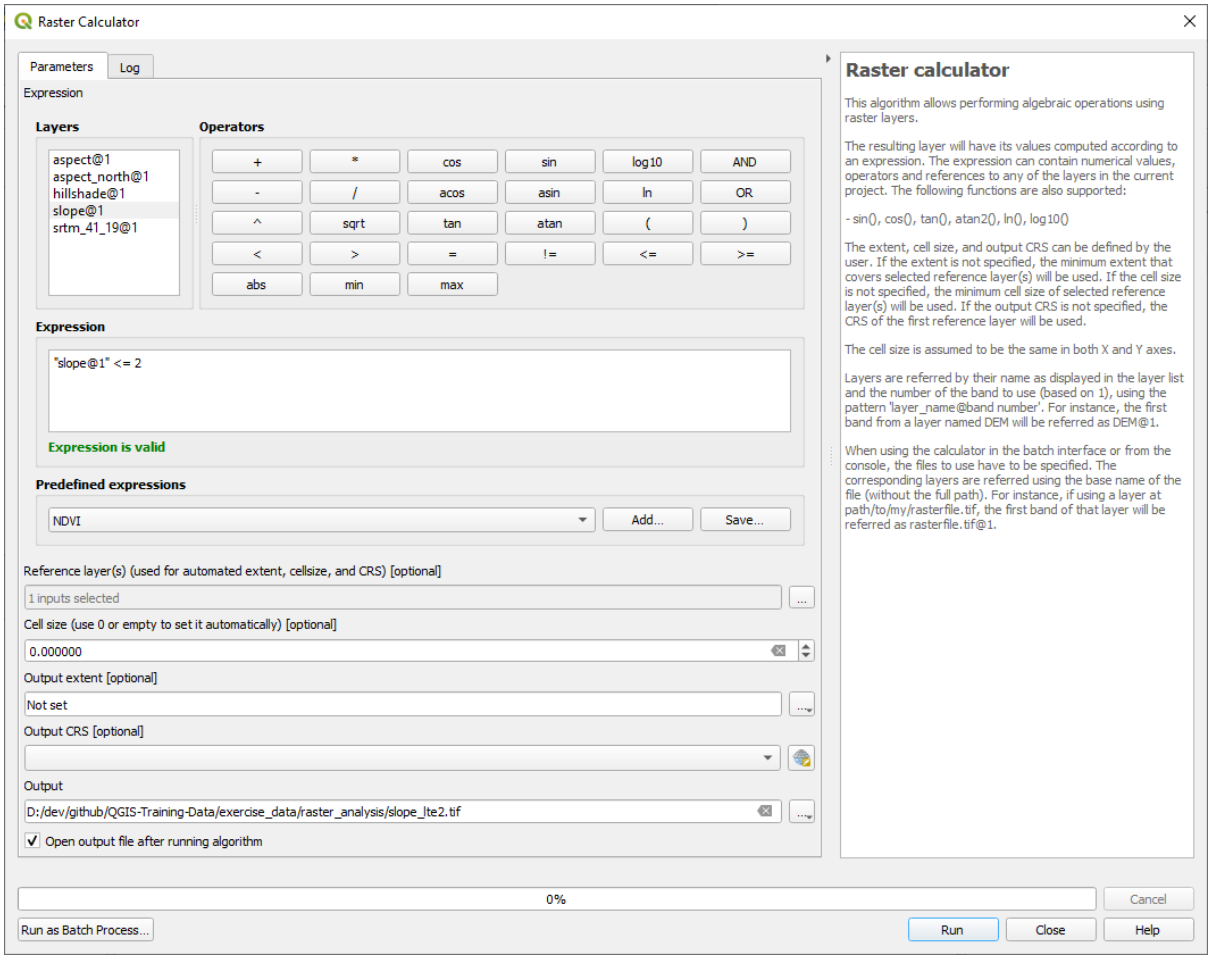
7.3.7 Попробуйте сами: Больше критериев

Теперь после того как вы закончили аспект, создайте два новых слоя из ЦМР.

- Первый слой должен определять участки, где уклон меньше или равен 2 градусам.
- Второй аналогичен, но уклон должен быть меньше или равен `5` градусам.
- Сохраните их внутри `exercise_data/raster_analysis` как `slope_lte2.tif` и `slope_lte5.tif`.

Answer

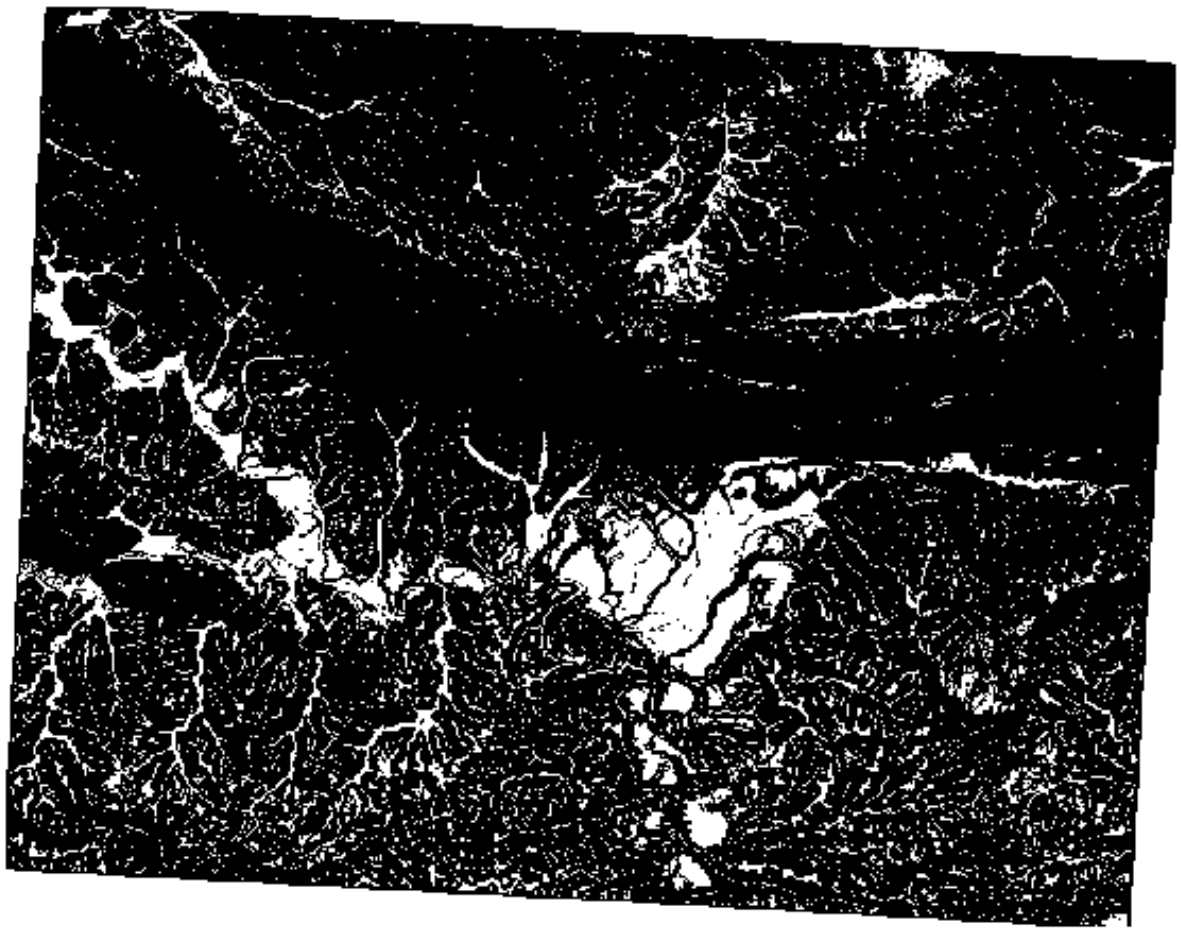
- Set your *Raster calculator* dialog up with:
- the following expression: `slope@1 <= 2`
- the `slope` layer as the *Reference layer(s)*



- For the 5 degree version, replace the `2` in the expression and file name with `5`.

Your results:

- 2 degrees:



- 5 degrees:



7.3.8  Идем дальше: Объединяем результаты анализа растра

Вы теперь получили три растровых слоя из ЦМР:

- *aspect_north*: местность, обращенная на север
- *slope_lte2*: уклон, равный 2 градусу или ниже
- *slope_lte5*: уклон не более 5 градусов

Если условие выполняется, значение пикселя равно `1`. В остальных случаях он равен `0`. Поэтому, если вы умножите эти растры, пиксели со значением `1` для всех они получат значение `1` (остальные получат `0`).

Условия, которые должны быть выполнены:

- при уклоне 5 градусов или ниже местность должна быть обращена на север,
- при уклоне 2 градуса или ниже направление местности не имеет значения.

Поэтому, вам нужно найти местность, где уклон не превышает пяти градусов `AND` местность смотрит на север, `OR` уклон не более 2 градусов. Такая местность подойдет для постройки.

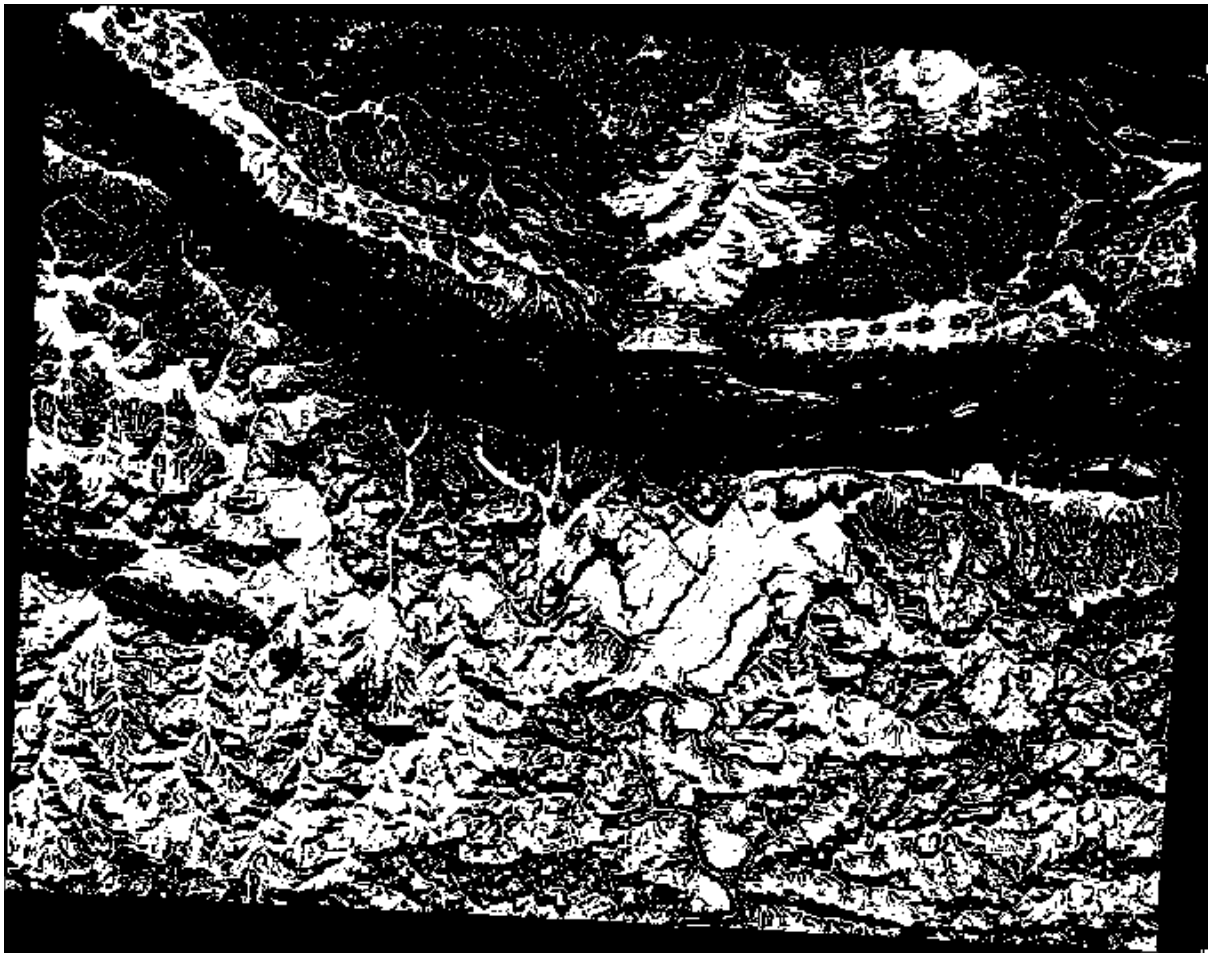
Чтобы вычислить ту местность, которая соответствует этим критериям, необходимо сделать следующее:

1. Надо открыть снова *Raster calculator*.
2. Используйте это выражение в *Expression*:

```
( aspect_north@1 = 1 AND slope_lte5@1 = 1 ) OR slope_lte2@1 = 1
```

3. Надо настроить параметр *Reference layer(s)* на `aspect_north` (неважно, выберете ли вы другой – они все были вычислены из `srtm_41_19`).
4. Сохраните выход/результат внутри `exercise_data/raster_analysis/` как `all_conditions.tif`.
5. Кликните на кнопку *Run*

Результат:



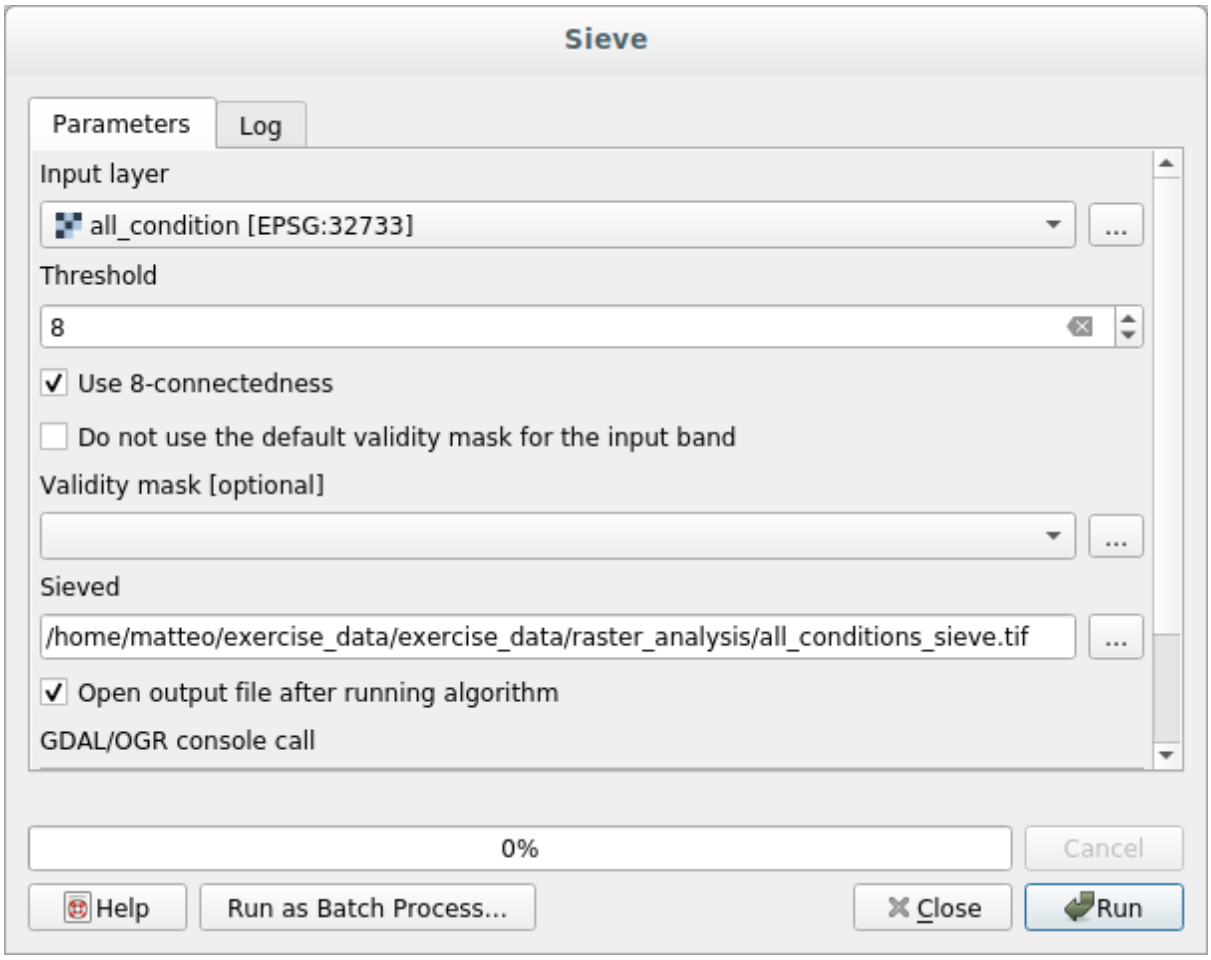
Подсказка: Предыдущие шаги можно было упростить с помощью следующей команды:

```
((aspect@1 <= 90 OR  aspect@1 >= 270) AND slope@1 <= 5) OR slope@1 <= 2
```

7.3.9  Идем дальше: Упрощаем растр

Как показывает изображение выше, комбинированный анализ выдал нам множество очень маленьких участков, которые отвечают условиям (выделены белым цветом). Но они не совсем полезны для нашего анализа, так как они слишком малы, чтобы на них что-либо строить. Давайте избавимся от всех этих крошечных непригодных для использования участков.

1. Надо открыть инструмент *Sieve* (*GDAL* ► *Raster Analysis* внутри *Processing Toolbox*).
2. Настройте *Input file* на `all_conditions`, а *Sieved* на `all_conditions_sieve.tif` (внутри `exercise_data/raster_analysis/`).
3. Установите *Threshold* до 8 (минимум восемь смежных пикселей) и отметьте *Use 8-connectedness*.

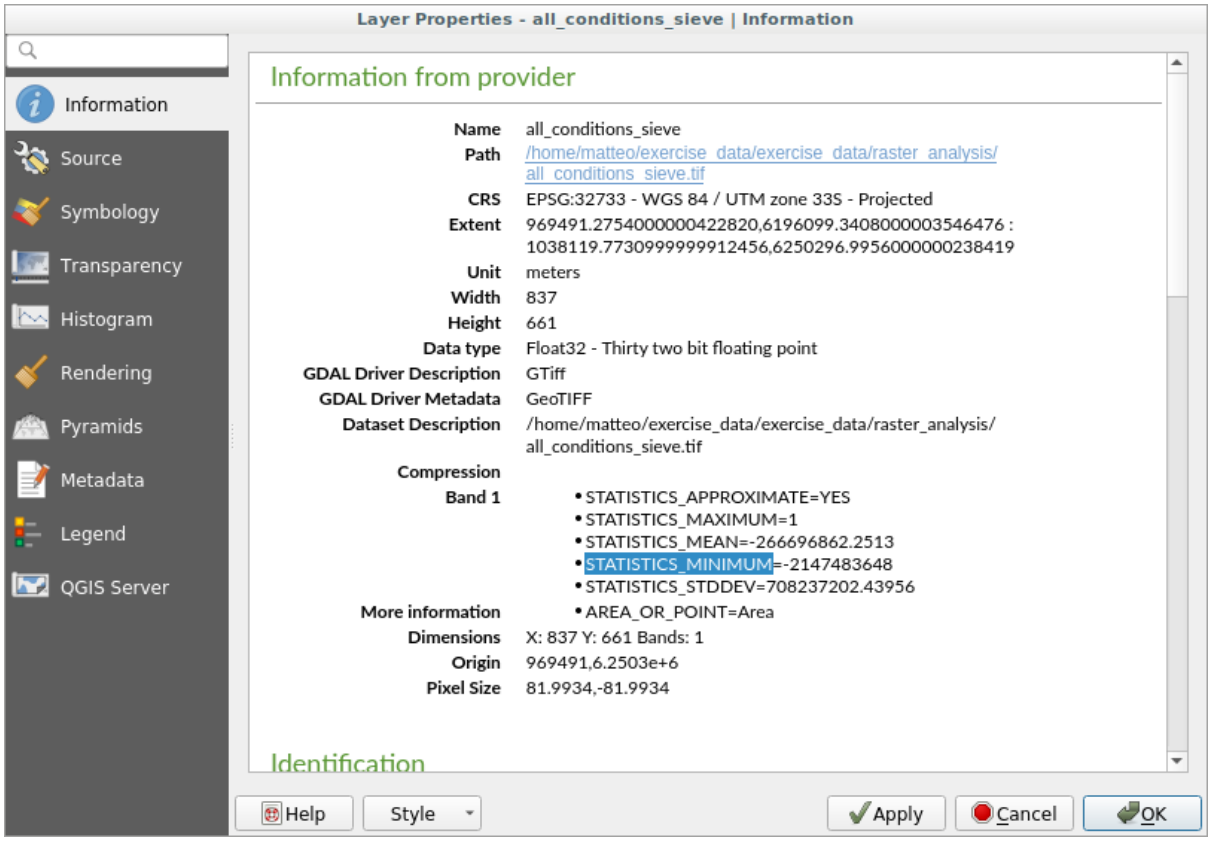


После завершения обработки будет загружен новый слой.



Что происходит? Ответ кроется в метаданных нового растрового файла.

4. Просмотрите метаданные внутри вкладки *Information* диалогового окна *Layer Properties*. Посмотрите на значение `STATISTICS_MINIMUM`:



Этот растр, как и тот, из которого он получен, должен содержать только значения `1` и `0`, но он имеет очень большое отрицательное число. Изучение данных показывает, что это число представлено как нулевое значение. Поскольку нам нужны только те участки, которые не были отфильтрованы, давайте настроим эти нулевые значения на ноль.

5. Надо открыть *Raster Calculator* и построить это выражение:

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

Это сохранит все положительные значения и переведет отрицательные числа на ноль, оставив все участки со значением `1` нетронутыми.

6. Сохраните выход/результат внутри `exercise_data/raster_analysis/` как `all_conditions_simple.tif`.

Ваш выход/результат выглядит так:



Что и следовало ожидать: упрощенная версия предыдущих результатов. Помните, что если результаты, которые вы получаете, не соответствует вашим ожиданиям, надо посмотреть на метаданные (и атрибуты вектора, если есть), которые помочь решить эти проблемы.

7.3.10  Идем дальше: Пере-классификация растра

Мы использовали калькулятор растра для вычисления растровых слоев. Есть еще один надежный инструмент, который мы можем использовать для того, чтобы извлечь информацию из существующих слоев.

Вернемся к слою `aspect`. Нам известно, что он имеет числовые значения в диапазоне от 0 до 360. Нам надо *пере-классифицировать* этот слой на другие дискретные значения (от 1 до 4) в зависимости от аспекта:

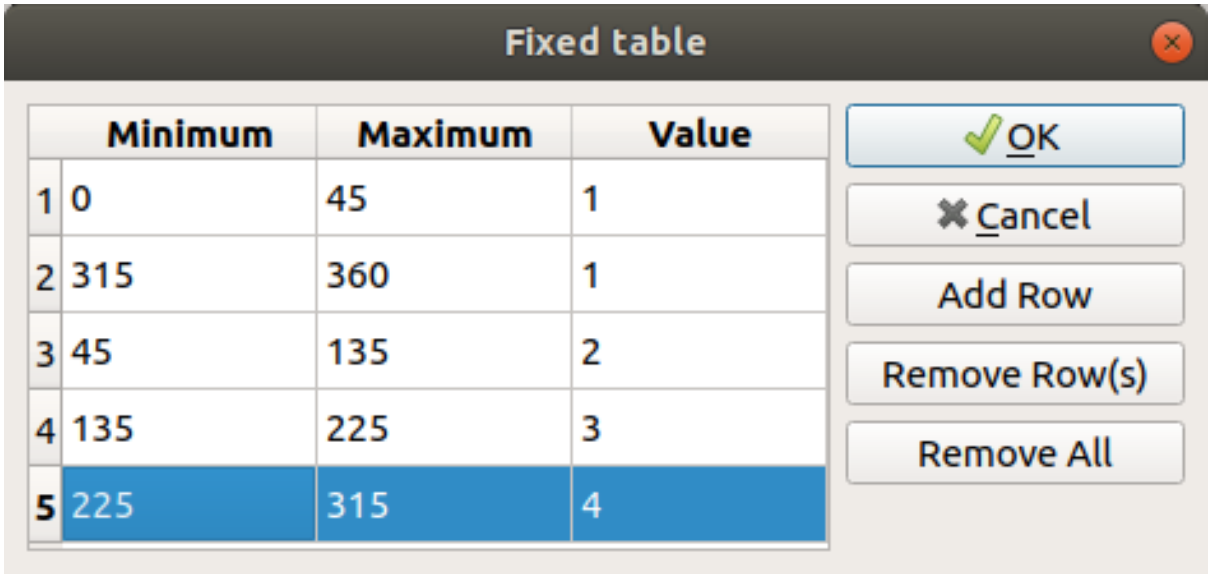
- 1 = север (от 0 до 45 и от 315 до 360);
- 2 = Восток (от 45 до 135);

- 3 = Юг (от 135 до 225);
- 4 = Запад (от 225 до 315).

Эту операцию можно выполнить с помощью калькулятора растра, но формула станет очень длинной.

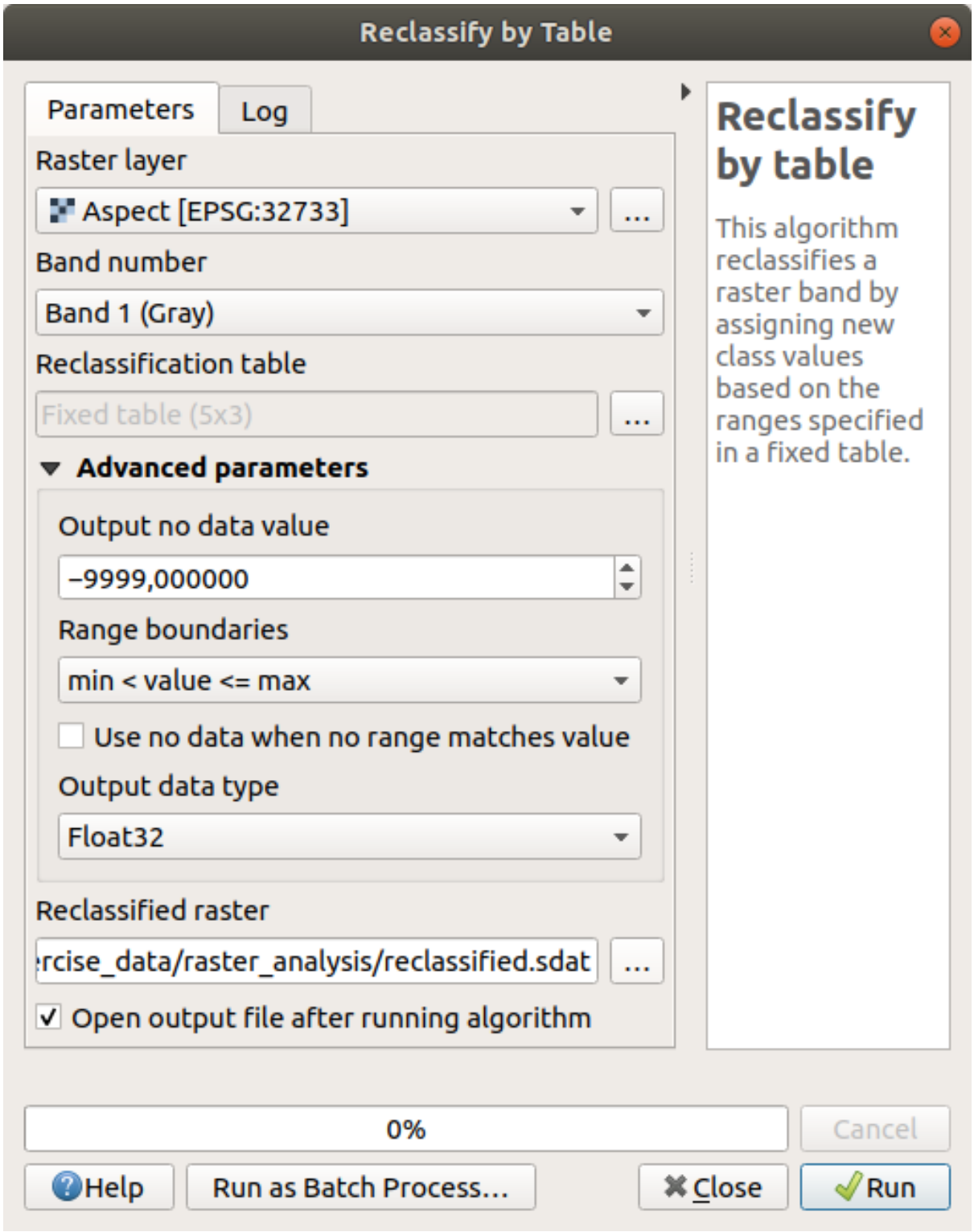
Альтернативный инструмент это инструмент *Reclassify by table* в *Raster analysis* внутри *Processing Toolbox*.

1. Надо открыть этот инструмент.
2. Выберите *aspect* как `Input raster layer`.
3. Кликните на кнопку … из *Reclassification table*. Появится диалоговое окно в виде таблицы, в котором вы можете выбрать минимальные, максимальные и новые значения для каждого класса.
4. Кликните на кнопку *Add row* и добавьте 5 строк. Заполните каждую строку, как показано на следующем рисунке, и кликните на кнопку *OK*:



   Метод, используемый алгоритмом для обработки пороговых значений каждого класса, определяется *Range boundaries*.

5. Сохраните слой как `reclassified.tif` внутри папки `exercise_data/raster_analysis/`.



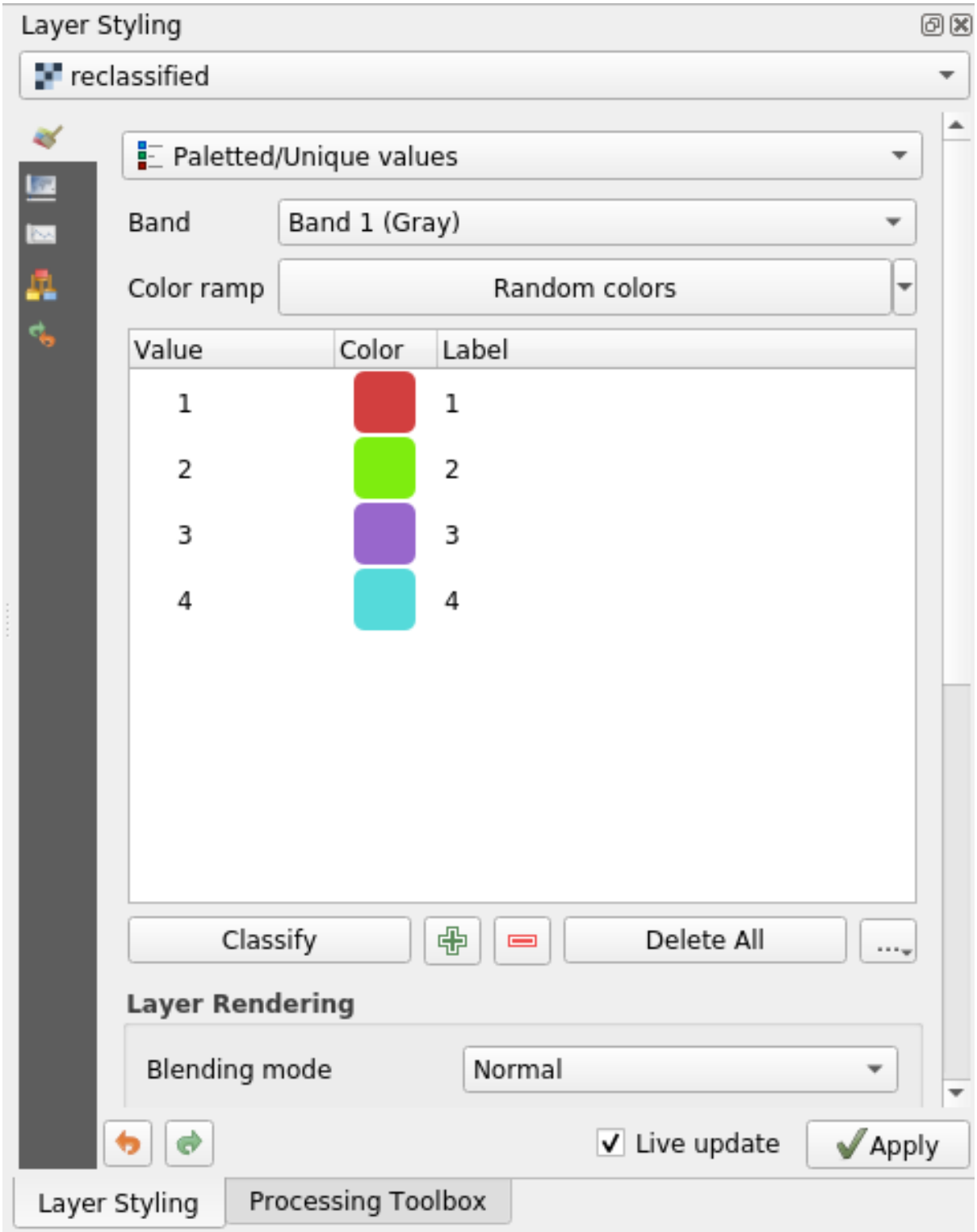
6. Кликните на кнопку *Run*.

Если вы сравните исходный слой *aspect* со слоем *reclassified*, вы не увидите больших различий. Но посмотрев на символы, вы увидите, что значения начинаются от `1` до `4`.

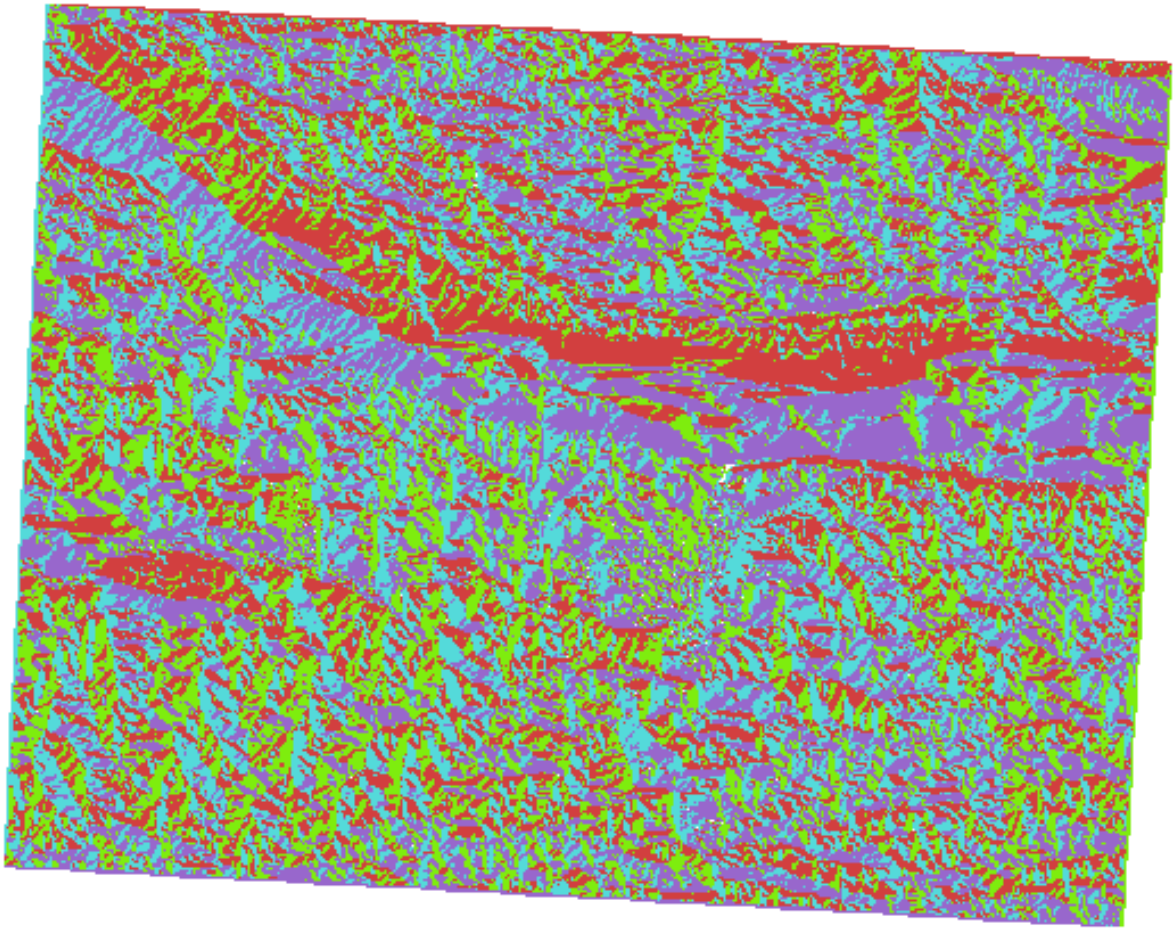
Давайте улучшим стиль этого слоя.

1. Надо открыть панель *Layer Styling*.
2. Выберите *Paletted/Unique values*, вместо *Singleband gray*.

3. Кликните на кнопку *Classify* для того, чтобы автоматически получить значения и присвоить им случайные цветов:



Выход/результат должен выглядеть следующим образом (цвета у вас могут быть разными, если они были сгенерированы случайным образом):



Применив к слою данную пере-классификацию и стиль палитры, вы можете сразу различать области аспекта.

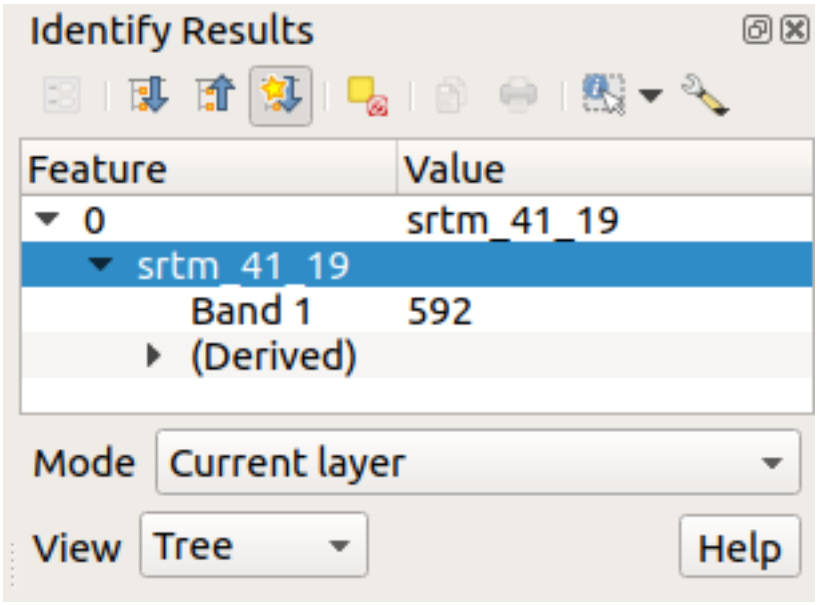
7.3.11  Идем дальше: Запрос к растру

В отличие от векторных слоев, растровые слои не имеют таблицу атрибутов. Каждый пиксель содержит одно или несколько числовых значений (однополосные или многополосные растры).

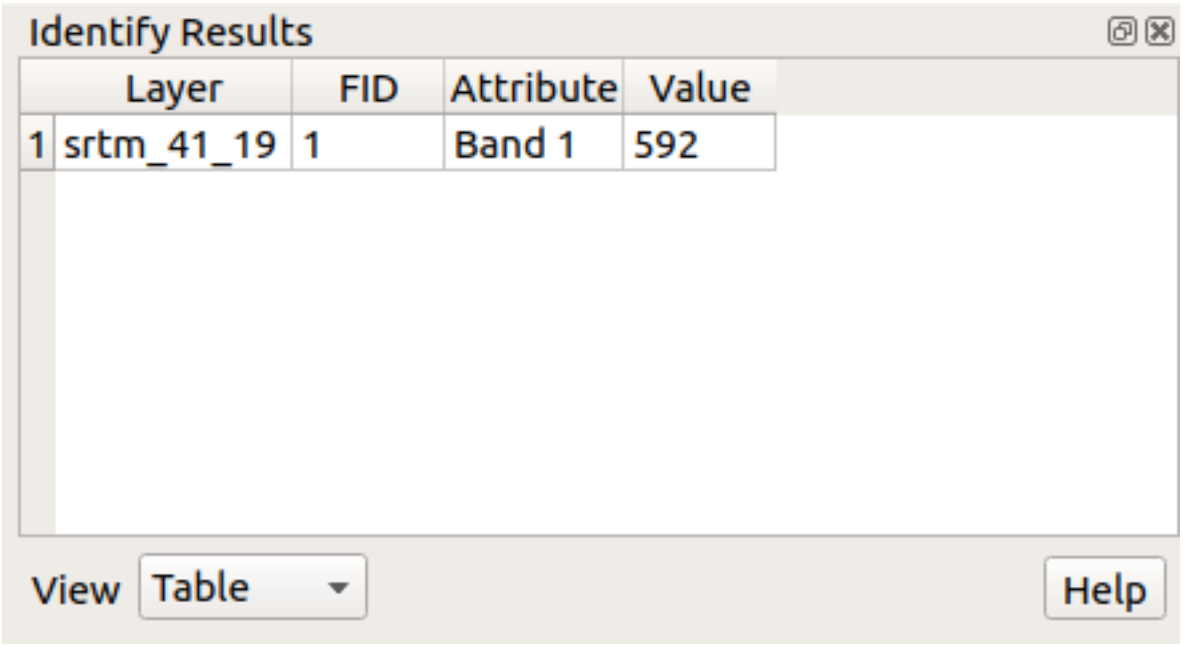
Все растровые слои, которые мы использовали в этом упражнении, состоят только из одной полосы. В зависимости от слоя значения пикселей могут показывать значения высоты, аспекта или уклона.

Как мы можем запросить растровый слой, чтобы получить значение пикселя? Мы можем использовать кнопку  Identify Features !

1. Надо выбрать инструмент в панели инструментов «Атрибуты».
2. Кликните на случайное место в слое *srtm_41_19*. Появится *Identify Results* со значением полосы в месте, которое вы выбрали:

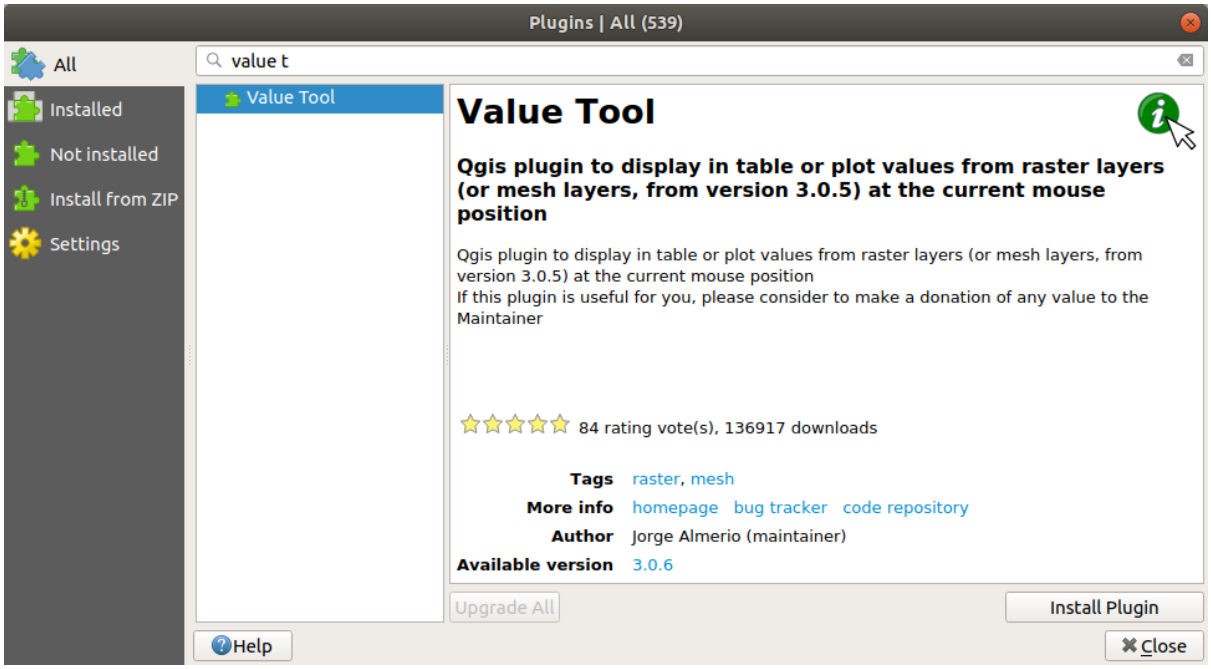


3. Вы сможете изменить выход панели *Identify Results* из текущего режима `tree` на режим `table`, выбрав *Table* в меню *View* внизу панели:



Через некоторое время вам может надоесть нажимать на каждый пиксель для получения значения растра. Мы можем использовать плагин *Value Tool* для решения этой проблемы.

1. Надо перейти на *Plugins* ► *Manage/Install Plugins...*
2. Во вкладе *All* наберите `value t` в поле поиска.
3. Выберите плагин *Value Tool*, кликните на *Install Plugin* , а потом *Close* диалоговое окно.

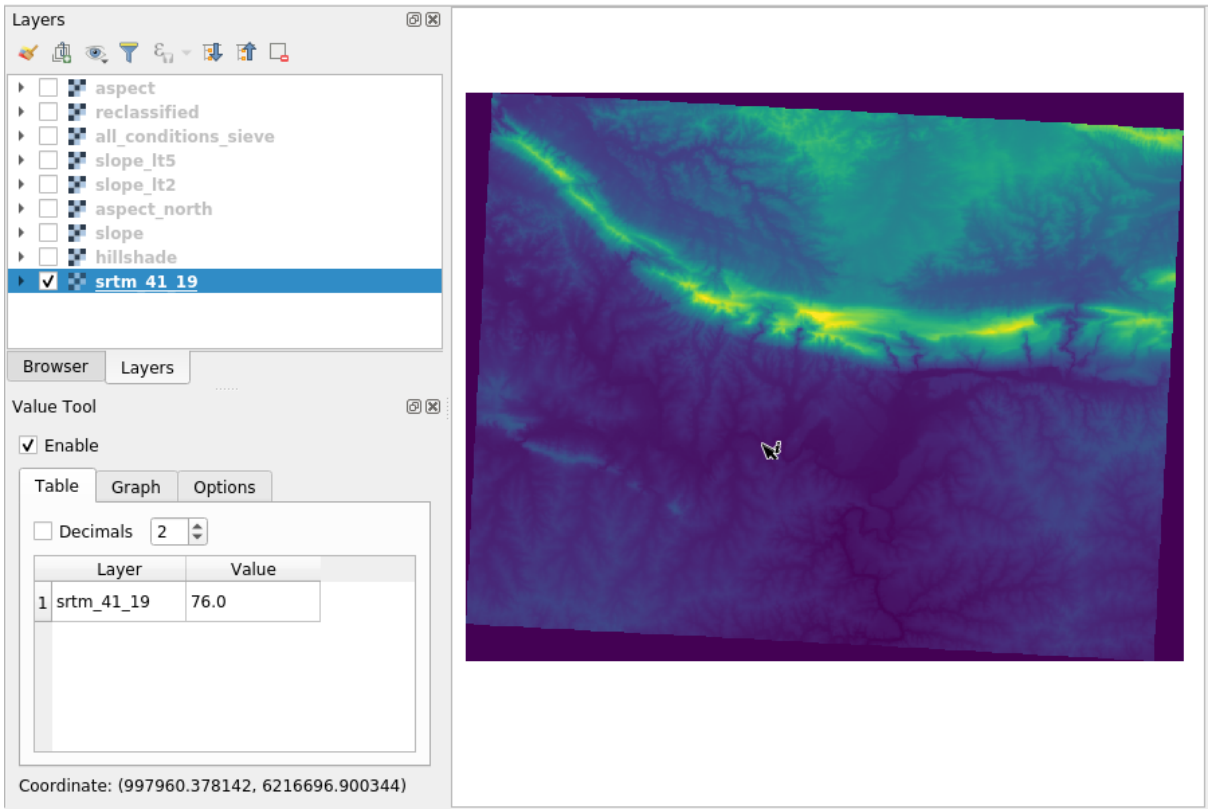


Появится новая панель *Value Tool*.

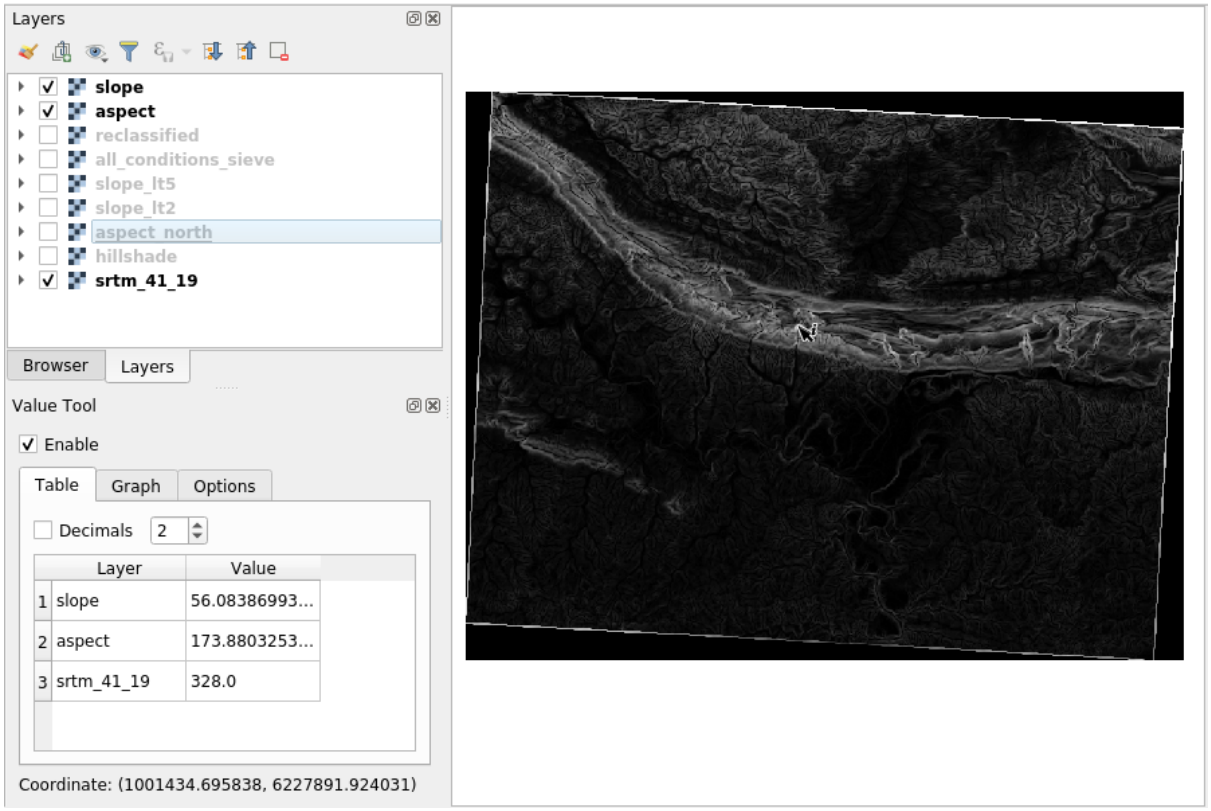
Совет: Если вы закроете панель, вы сможете снова открыть ее, активировав ее в *View* ► *Panels* ► *Value Tool* или кликнув на значок на панели инструментов.

---

4. Чтобы использовать плагин, просто кликните на флажок *Enable* и убедитесь, что слой `srtm_41_19` является активным (отмечен) в панели *Layers*.
5. Наведите курсор на карту, чтобы увидеть значение пикселей.



6. Но есть еще кое-что. Плагин Value Tool позволяет запрашивать **все** активные растровые слои в панели *Layers*. Сделайте слои *aspect* и *slope* снова активными и наведите указатель мыши на карту:



7.3.12 В заключении

Вы посмотрели, как можно получить все виды аналитических продуктов из ЦМР. К ним относятся вычисление отмывки, уклона и аспекта. Вы также узнали, как использовать калькулятор растра для дальнейшего анализа и объединения этих результатов. И наконец, вы узнали, как пере-классифицировать слой и как запрашивать результаты.

7.3.13 Что дальше?

Теперь у вас есть два анализа: векторный анализ, показывающий вам потенциально подходящие участки, и растровый анализ, который показывает потенциально подходящую местность. Как их можно объединить, чтобы получить окончательный результат для решения этой проблемы? Это будет темой следующего урока, который начинается со следующего модуля.

ГЛАВА 8

Модуль: Завершение анализа

Вы теперь имеете две половины анализа: векторный и растровый. В этом модуле вы узнаете, как их объединить. Вы сможете завершить анализ и представить окончательные результаты.

8.1 Занятие: Преобразование растрового в векторное

Преобразовав между растровыми и векторными форматами, вы сможете использовать как растровые, так и векторные данные при решении задач GIS, а также применить различные методы анализа, которые уникальны для этих двух форм географических данных. Таким образом вы сможете повысить гибкость при рассмотрении источников данных и методов обработки для решения задач GIS.

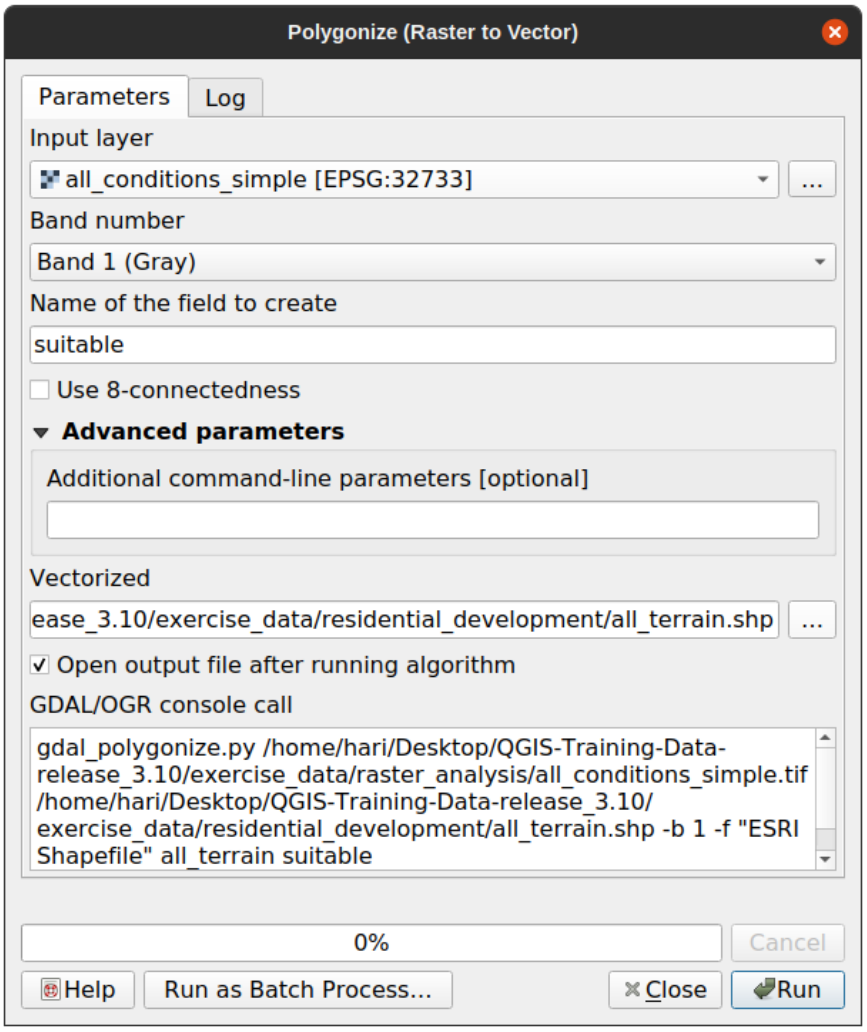
Для объединения растрового и векторного анализ вам надо преобразовать один тип данных в другой. Давайте попробуем преобразовать результаты работы с растровыми данными предыдущего урока в векторную плоскость.

Цель этого урока: Перевести результаты работы с растровыми данными в векторный, которые можно использовать для завершения анализа.

8.1.1  Идем дальше: Инструмент *Raster to Vector*

Начните работу с карты из предыдущего модуля, `raster_analysis.qgs`. У вас там должен быть файл `all_conditions_simple.tif`, вычисленный во время выполнения предыдущих упражнений.

- Кликните на *Raster* ► *Conversion* ► *Polygonize (Raster to Vector)*. У вас появится диалоговое окно инструмента.
- Настройте его таким образом:



- Поменяйте название поля (описывающее значения растра) на *suitable*.
- Сохраните слой внутри `exercise_data/residential_development` как `all_terrain.shp`.

У вас теперь есть векторный файл, который имеет все значения растра, но вам нужны те местности, которые являются подходящими; т.е. те полигоны, где значение *suitable* является `1`. Вы можете изменить стиль этого слоя, если хотите более четко его визуализировать.

8.1.2 Попробуйте сами

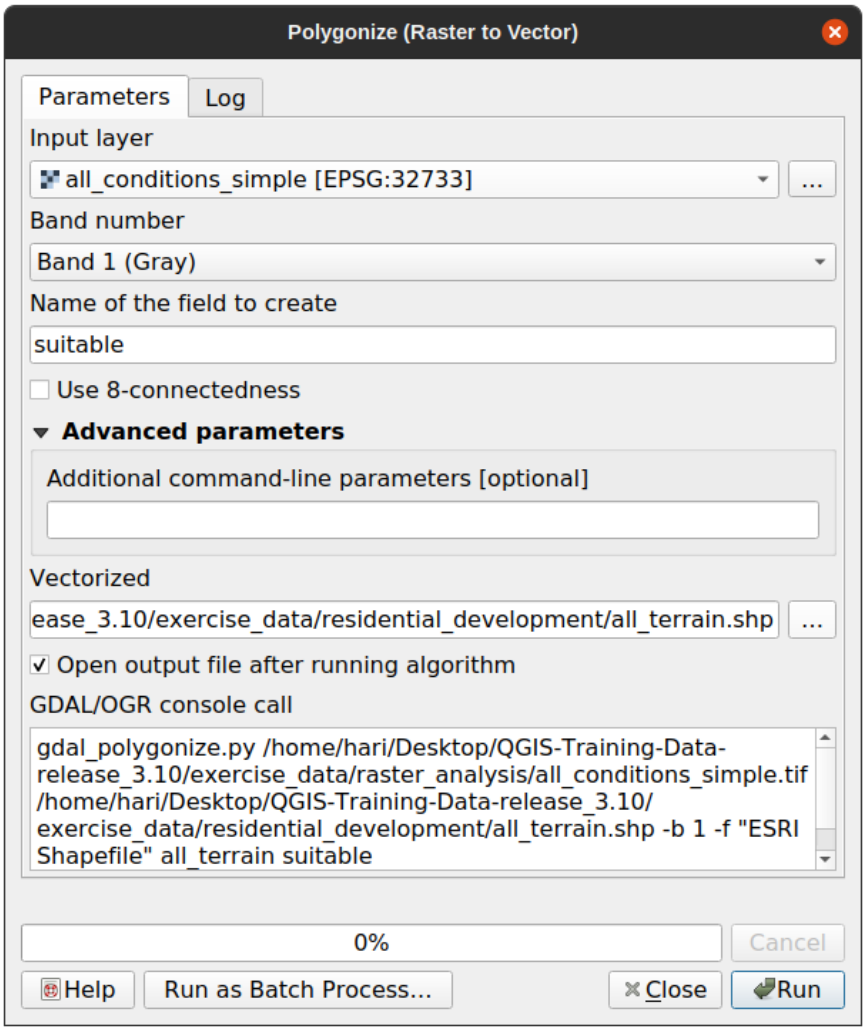
Вернитесь к модулю векторного анализа.

- Создайте новый векторный файл, который содержит только полигоны, в которых значение *suitable* равно `1`.
- Сохраните новый файл в `exercise_data/residential_development/` как `suitable_terrain.shp`.

Answer

1. Open the *Query Builder* by right-clicking on the *all_terrain* layer in the *Layers* panel, and selecting the *Properties* ► *Source* tab.

2. Then build the query `"suitable" = 1`.
3. Click *OK* to filter out all the polygons where this condition isn't met. When viewed over the original raster, the areas should overlap perfectly:

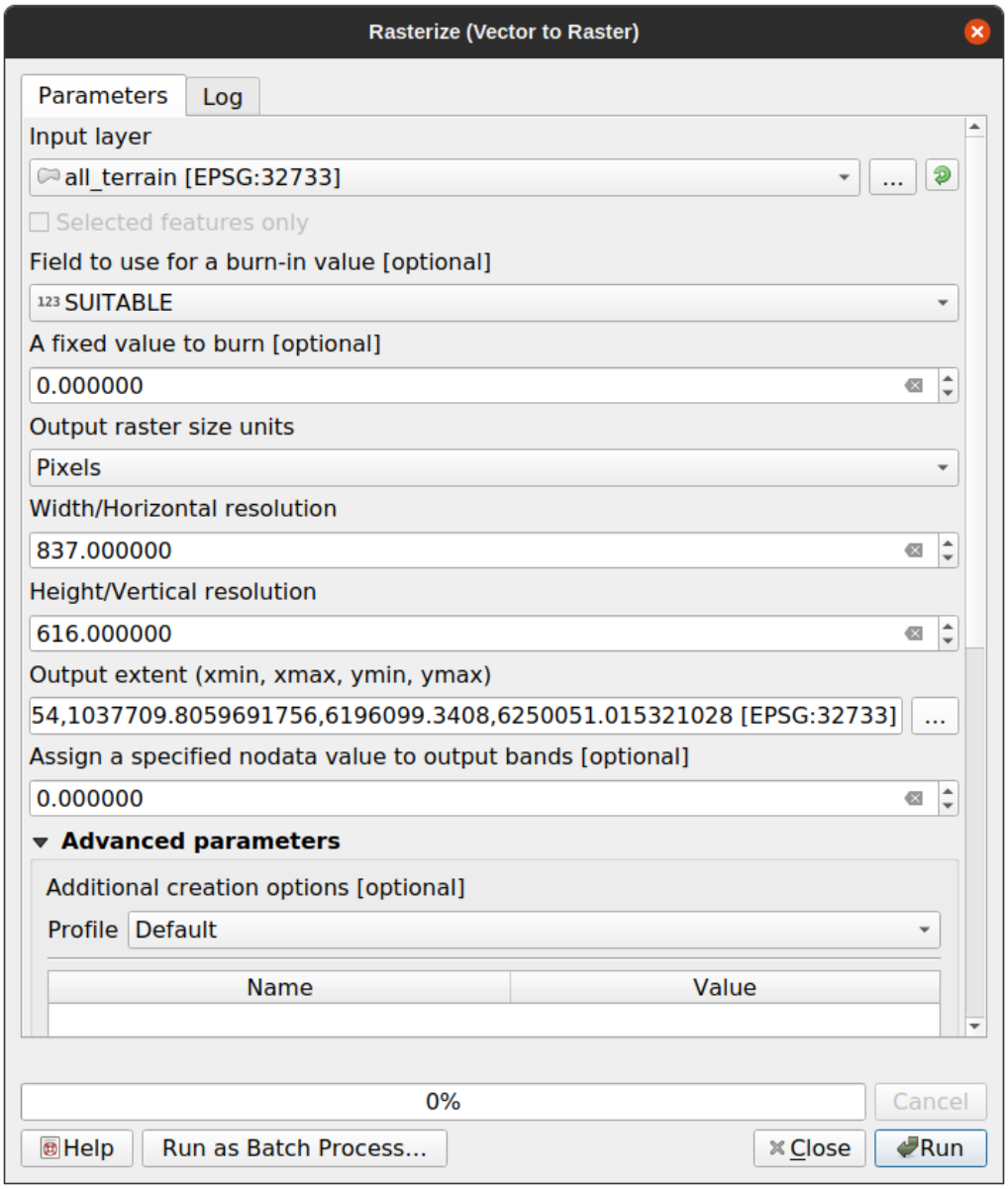


4. You can save this layer by right-clicking on the *all_terrain* layer in the *Layers* panel and choosing *Save As...*, then continue as per the instructions.

8.1.3 Идем дальше: Инструмент *Vector to Raster*

Хотя это и не является обязательным для нашей текущей задачи, было бы хорошо знать обратное преобразование. Преобразуйте в растровый предыдущий векторный файл `suitable_terrain.shp`, который вы создали в предыдущем задании.

- Надо кликнуть на *Raster* ► *Conversion* ► *Rasterize (Vector to Raster)* чтобы запустить этот инструмент, затем надо настроить его, как показано ниже:



- *Input layer* это *all_terrain*.
- Имя поля *suitable*.
- *Output raster size units* это *Pixels*.
- *Width* и *Height* составляют `837` и `661`, соответственно.
- Возьмите *Output extent* из слоя *all_terrain*.
- Установите выходной файл *Rasterized* на `exercise_data/residential_development/raster_conversion.tif`.

Примечание: TРазмер выходного изображения, указанный здесь, является таким же, как исходный растр, который был преобразован в векторный. Для того, чтобы просмотреть параметры изображения, откройте его метаданные (Вкладка *Metadata* в *Layer Properties*).

- Надо кликнуть на кнопку *OK* в диалоговом окне для запуска процесса преобразования.
- Когда процесс будет завершен, оцените его успех, сравнив новый растр с исходным. Они должны быть полностью одинаковыми, пиксель к пикселю.

8.1.4 В заключении

Преобразование между растровыми и векторными форматами даст вам возможность более шире применять данные и это не должно приводить к ухудшению качества данных.

8.1.5 Что дальше?

У нас теперь есть результаты анализа местности в векторном формате и их можно использовать для решения задачи, какие здания следует рассматривать для жилой застройки.

8.2 Занятие: Объединение анализов

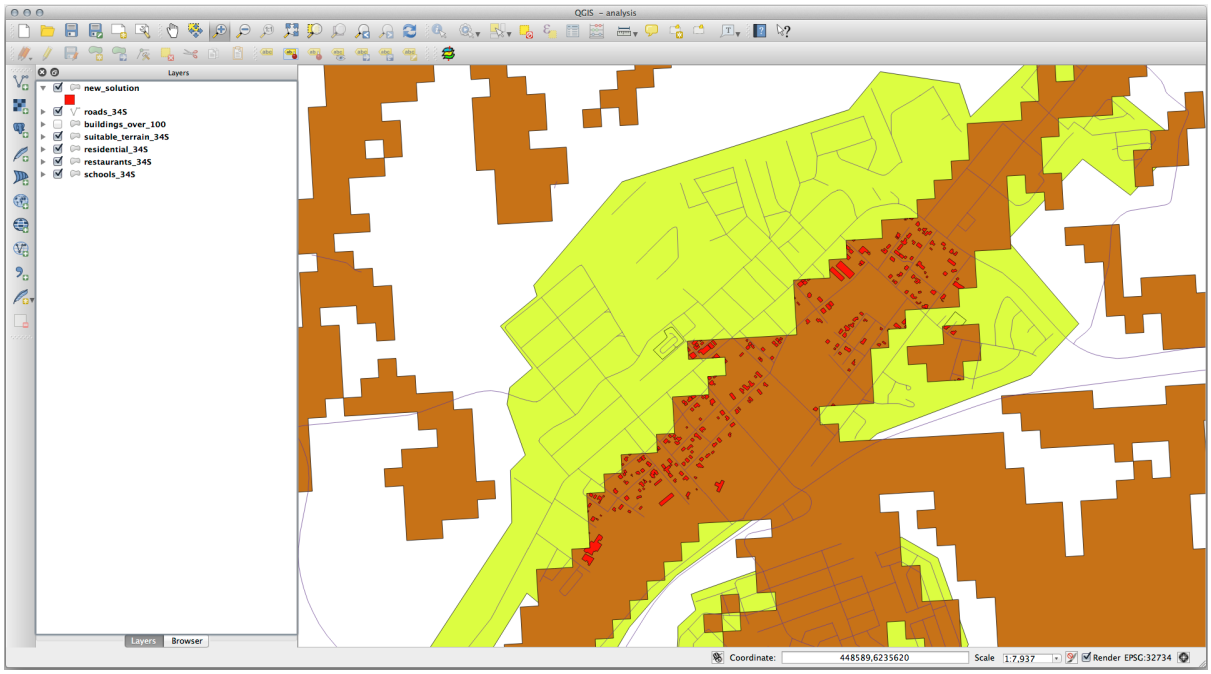
Использование векторизованных результатов растрового анализа позволит вам выбрать только те здания на подходящей местности.

Цель этого урока: Использовать результаты анализа векторизованного ландшафта для выбора подходящих участков.

8.2.1  Попробуйте сами

1. Сохраните вашу текущую карту (`raster_analysis.qgs`).
2. Откройте созданную вами карту во время векторного анализа, проведенного ранее (вы должны были сохранить файл как `analysis.qgs`).
3. В панели *Layers* вам надо активировать эти слои:
   - *hillshade*
   - *solution* (или *buildings_over_100*)
4. Кроме этих слоев, которые уже должны быть загружены на карту, когда вы работали над ней раньше, надо также набор данных `suitable_terrain.shp`.
5. Если вам не хватает некоторых слоев, вы должны найти их внутри `exercise_data/residential_development/`
6. Вам надо использовать инструмент *Intersection* tool (*Vector* ► *Geoprocessing Tools*) для того, чтобы создать новый векторный слой, который называется `new_solution.shp` и содержит только те здания, которые пересекают слой *suitable_terrain*.

У вас теперь должен быть слой, показывающий ваш вариант определенных зданий, например:



8.2.2 Попробуйте сами: Проверить результаты

Посмотрите на каждое из зданий в вашем слое *new_solution*. Сравните их со слоем *suitable_terrain*, изменив символы для слоя *new_solution* так, чтобы на нем были только контуры. Что вы заметили в некоторых зданиях? Все ли они подходят только потому, что пересекаются со слоем *suitable_terrain*? Почему или почему нет? Какие из них вы считаете не подходят?

Answer

You may notice that some of the buildings in your `new_solution` layer have been «sliced» by the *Intersection* tool. This shows that only part of the building - and therefore only part of the property - lies on suitable terrain. We can therefore sensibly eliminate those buildings from our dataset.

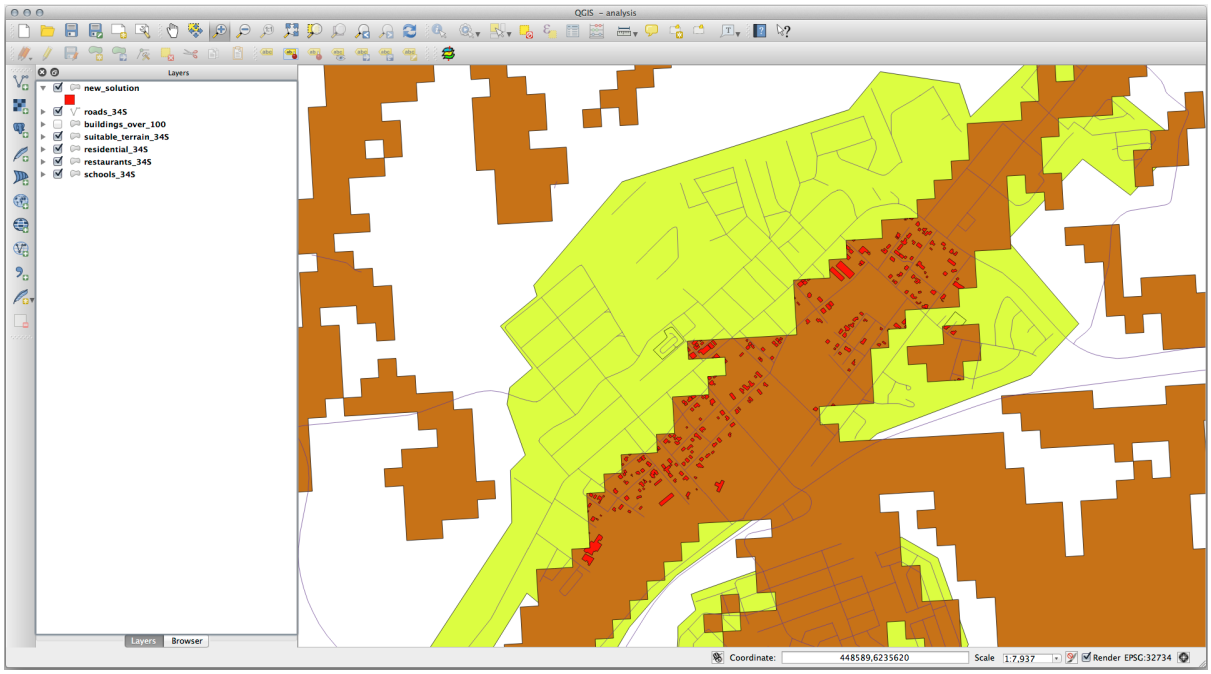
8.2.3 Попробуйте сами: Уточнить анализ

Вы можете увидеть из результатов, что некоторые из включенных зданий на самом деле не походят и поэтому нам надо уточнить анализ.

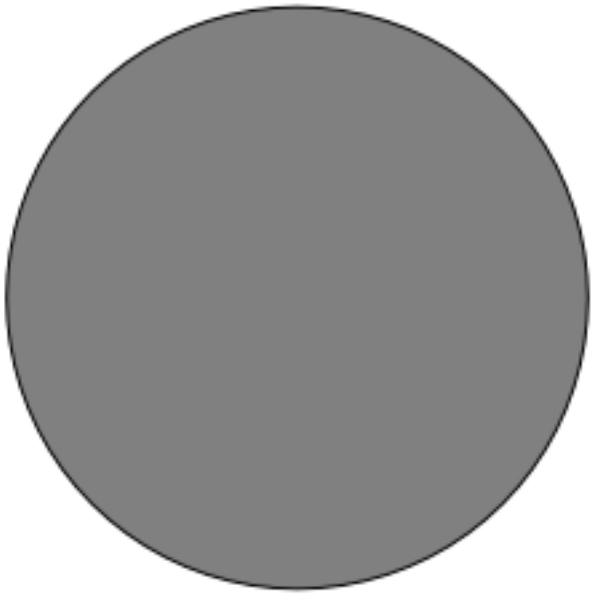
Мы хотим убедиться, что наш анализ выдаст только те здания, которые полностью сидят внутри слоя *suitable_terrain*. Как вам этого добиться? Надо использовать один или несколько инструментов векторного анализа и помните, что все наши здания имеют площадь более 100 квадратных метров.

Answer

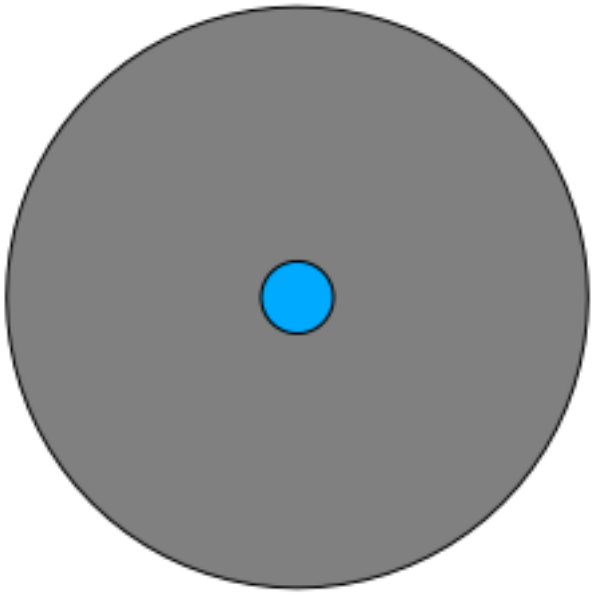
At the moment, your analysis should look something like this:



Consider a circular area, continuous for 100 meters in all directions.



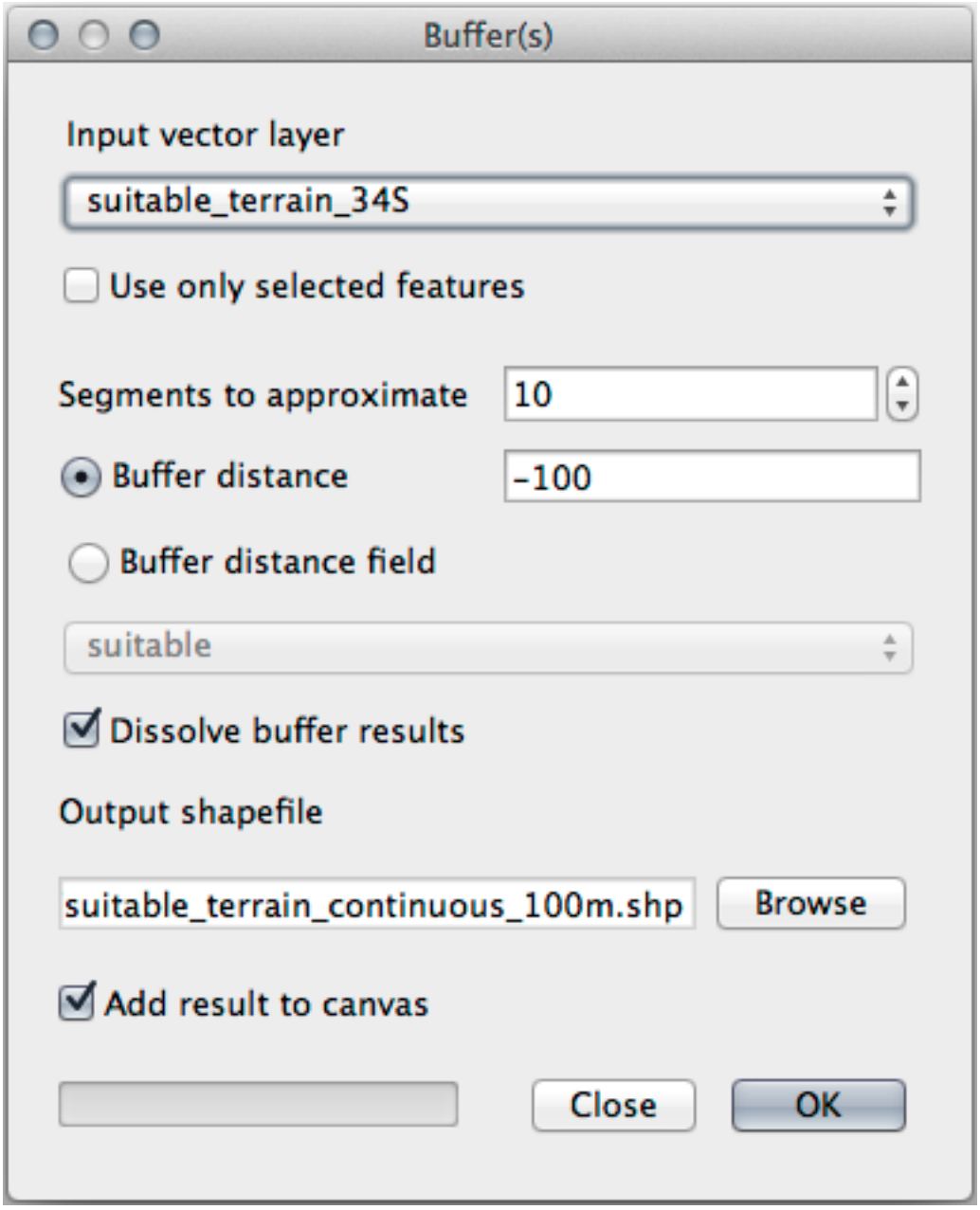
If it is greater than 100 meters in radius, then subtracting 100 meters from its size (from all directions) will result in a part of it being left in the middle.



Therefore, you can run an *interior buffer* of 100 meters on your existing *suitable_terrain* vector layer. In the output of the buffer function, whatever remains of the original layer will represent areas where there is suitable terrain for 100 meters beyond.

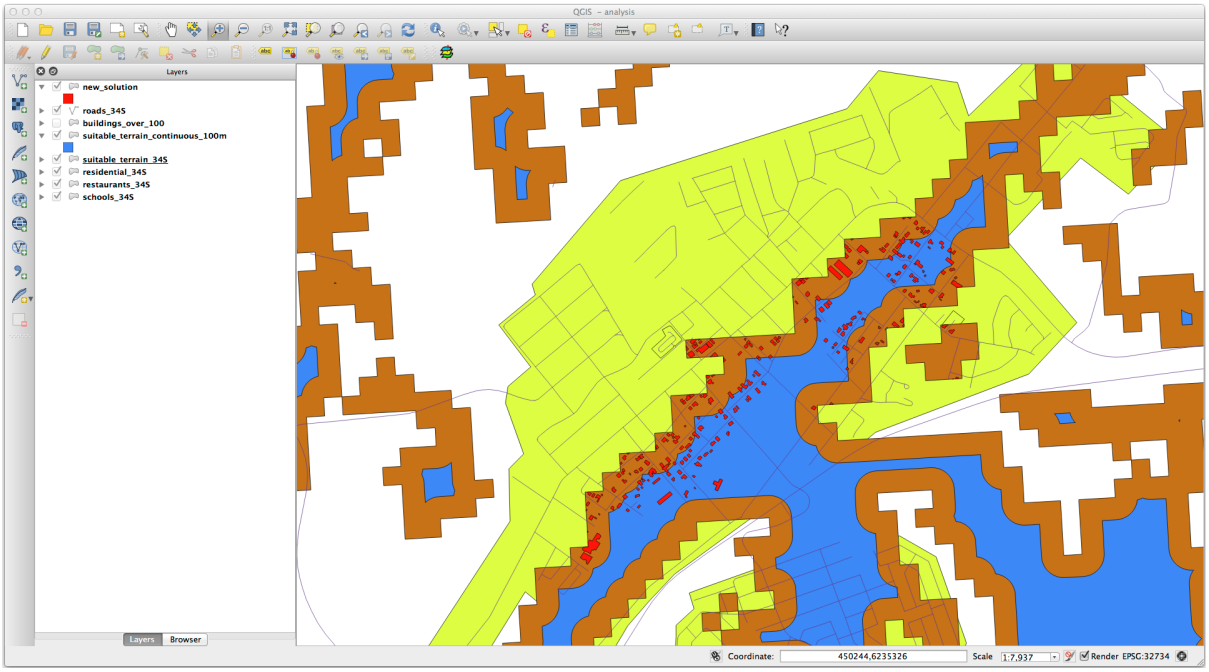
To demonstrate:

1. Go to *Vector* ► *Geoprocessing Tools* ► *Buffer(s)* to open the Buffer(s) dialog.
2. Set it up like this:

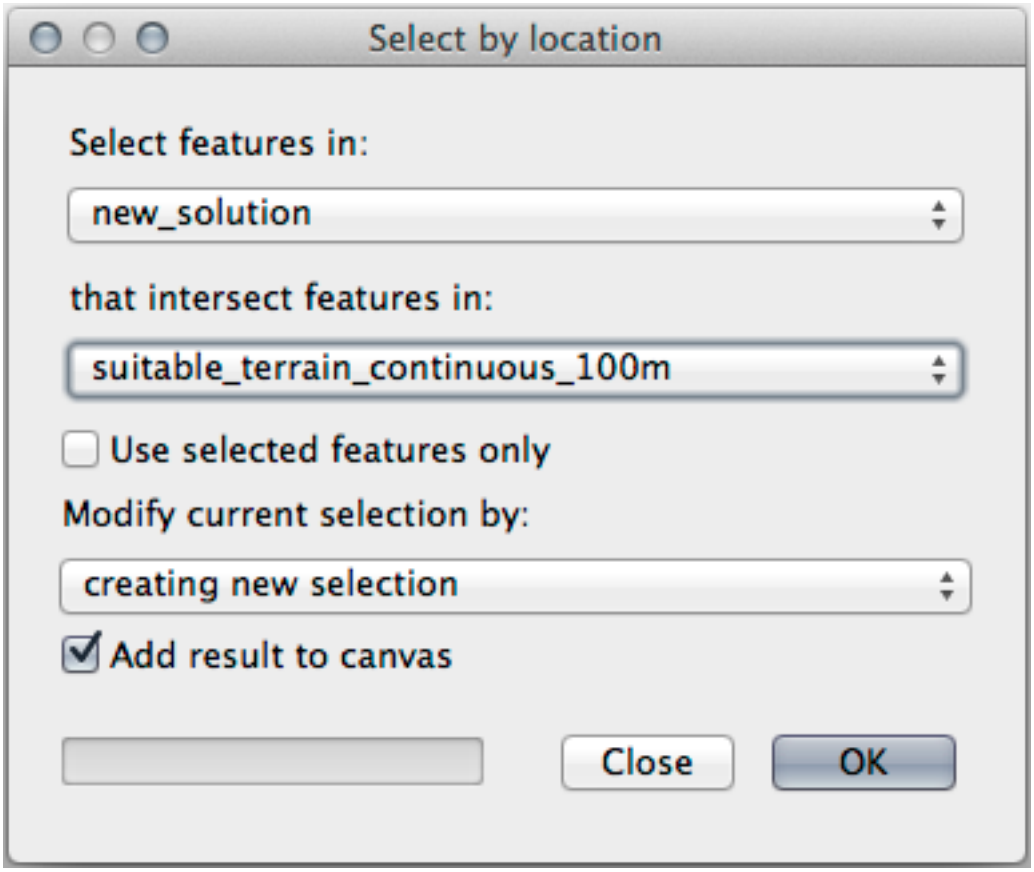


3. Use the *suitable_terrain* layer with `10` segments and a buffer distance of `-100`. (The distance is automatically in meters because your map is using a projected CRS.)
4. Save the output in `exercise_data/residential_development/` as `suitable_terrain_continuous100m.shp`.
5. If necessary, move the new layer above your original *suitable_terrain* layer.

   Your results will look like something like this:

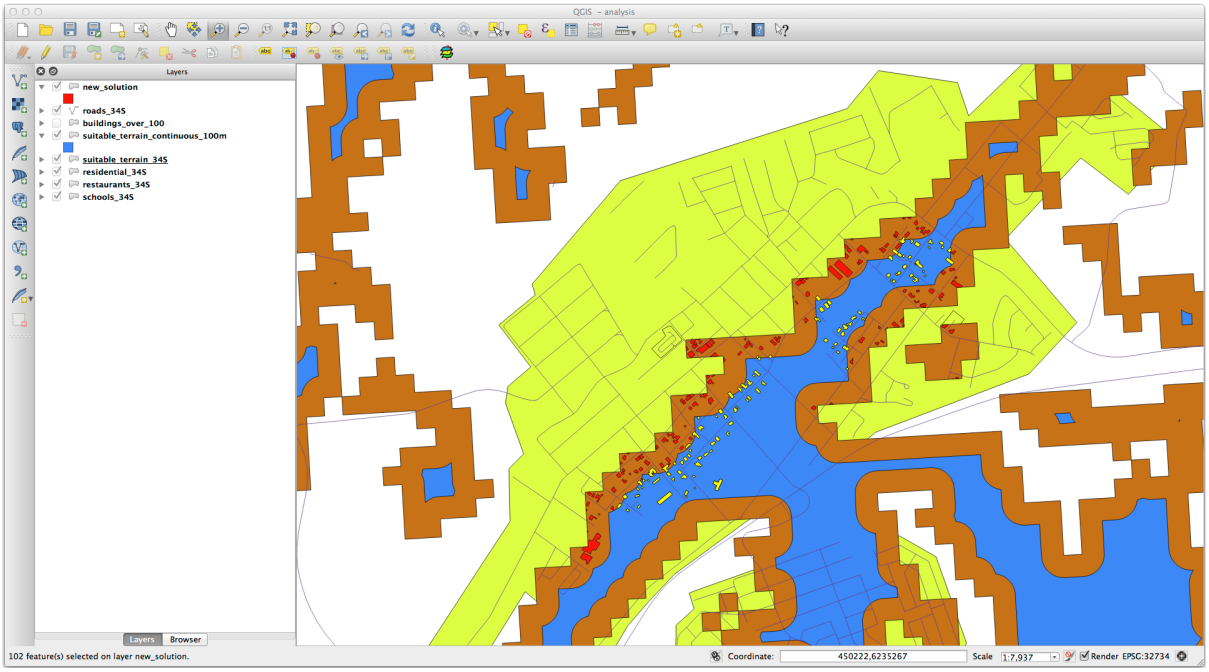


6. Now use the *Select by Location* tool (*Vector* ► *Research Tools* ► *Select by location*).
7. Set up like this:



8. Select features in *new_solution* that intersect features in *suitable_terrain_continuous100m.shp*.

   This is the result:



The yellow buildings are selected. Although some of the buildings fall partly outside the new *suitable_terrain_continuous100m* layer, they lie well within the original *suitable_terrain* layer and therefore meet all of our requirements.

9. Save the selection under `exercise_data/residential_development/` as `final_answer.shp`.

8.2.4 В заключении

Вы теперь ответили на исходный вопрос исследования и можете высказать свое мнение (с указанием причин, подкрепленных анализом) относительно того, какой объект недвижимости следует улучшить.

8.2.5 Что дальше?

Далее вы представите эти результаты как часть вашего второго задания.

8.3 Задание

Используя макет для печати, создайте новую карту с результатами вашего анализа. Включите эти слои:

- *places* (с надписями),
- *hillshade*,
- *solution* (или *new_solution*),
- *roads*,
- или *aerial_photos* или *DEM*.

Напишите краткий пояснительный текст к нему. В тексте пропишите критерий при оценке жилья для покупки и последующего улучшения, а также объяснение своих рекомендаций относительно того, какие здания подходят.

8.4 Занятие: Дополнительное упражнение

В этом уроке вы пройдете полный анализ GIS в QGIS.

Примечание: Содержание урока разработано Linfiniti Consulting (ЮАР) и Siddique Motala (Технологический университет Кейп-Пенинсула).

8.4.1 Постановка задачи

Перед вами поставлена задача найти районы на полуострове Кейп и вокруг него, которые являются подходящей средой обитания для редких видов растений финбос. Протяженность области вашего исследования охватывает Кейптаун и полуостров Кейп между Мелкбосстрандом на севере и Стрэндом на юге. Ботаники предоставили вам следующие предпочтения, присущие рассматриваемым видам растений:

- Растение растет на склонах восточного направления.
- Растение растет на склонах с уклоном от 15% до 60%.
- Растение растет в местности с общим годовым количеством осадков > 1000 мм.
- Растение можно найти только на расстоянии не менее 250 м от любого человеческого поселения.
- Площадь растительности, на которой растение встречается, должна быть не менее 6000 m^2.

Как студент университета, вы согласились искать растение на четырех разных подходящих участках земли. Вы хотите, чтобы эти четыре подходящих участка были ближайшими к Кейптаунскому университету, в котором вы живете. Воспользуйтесь своими навыками работы с GIS, чтобы определиться, где вам следует искать.

8.4.2 Описание решения

Данные для этого упражнения можно найти в папке `exercise_data/more_analysis`.

Вам надо найти четыре подходящих участка, которые находятся ближе всего к Кейптаунскому университету.

Для того, чтобы решить эту задачу, необходимо сделать следующее:

1. Анализ растрового слоя ЦМР для поиска склонов восточного направления и с соответствующим уклоном.
2. Анализ растрового слоя осадков, чтобы найти участки с соответствующим количеством осадков.
3. Анализ векторного слоя зонирования для того, чтобы найти участки, удаленные от населенных пунктов, и имеющие соответствующий размер.

8.4.3 Идем дальше: Настройка карты

1. Надо кликнуть на кнопку  Current CRS в правом нижнем углу экрана. Под вкладкой *CRS* появившегося диалогового окна используйте инструмент «Фильтр» для поиска «33S». Выберите запись *WGS 84 / UTM zone 33S* (с кодом EPSG `32733`).
2. Кликните *OK*.
3. Сохраните файл проекта, нажав н кнопку панели  Save Project или воспользуйтесь пунктом меню *File* ► *Save As...*

   Сохраните его в новом каталоге под названием `Rasterprac`, который вы должны создать где-нибудь на своем компьютере. В этом каталоге вы должны также сохранять все создаваемые вами слои. Сохраните проект как `your_name_fynbos.qgs`.

8.4.4 Загрузка данных на карту

Для того, чтобы обработать данные, вам надо загрузить необходимые слои (названия улиц, зоны, осадки, ЦМР, районы) на полотно карты.

Для векторов...

1. Кликните на кнопку  Open Data Source Manager на панели инструментов *диспетчера источников данных* и активируйте вкладку  *Vector* в открывшемся диалоговом окне или воспользуйтесь кнопкой меню *Layer* ► *Add Layer* ►  *Add Vector Layer*....
2. Убедитесь, что выбран  *File*.
3. Кликните на кнопку ... кнопка для просмотра векторных наборов данных.
4. В появившемся диалоговом окне вам надо открыть каталог `exercise_data/more_analysis/Streets`.
5. Выберите файл `Street_Names_UTM33S.shp`.
6. Кликните на кнопку *Open*.

   Диалоговое окно закроется и появится исходное диалоговое окно с указанием пути к файлу в текстовом поле рядом с *Vector dataset(s)*. Таким образом вы сможете убедиться, что выбран правильный файл. Можно также ввести путь к файлу в этом поле вручную, если захотите.
7. Кликните на кнопку *Add*. Векторный слой будет загружен на вашу карту. Цвет карты будет автоматически присвоен. Вы измените его позже.
8. Переименуйте слой на `Streets`.
   1. Кликните на его правой кнопкой мыши в панели *Layers* по умолчанию панель в левой части экрана).
   2. Кликните на кнопку *Rename* в появившемся диалоговом окне и переименуйте его, нажав на кнопку `Enter` когда закончите.
9. Повторите процесс добавления вектора, но на этот раз выберите файл `Generalised_Zoning_Dissolve_UTM33S.shp` в каталоге `Zoning`.
10. Переименуйте его на `Zoning`.
11. Также загрузите векторный слой `admin_boundaries/Western_Cape_UTM33S.shp` в вашу карту.
12. Переименуйте его на `Districts`.

Для растров...

1. Кликни на кнопку  Open Data Source Manager и активируйте панель  *Raster* в открывшемся диалоговом окне или воспользуйтесь кнопкой меню *Layer* ► *Add Layer* ►  *Add Raster Layer*....
2. Убедитесь, что выбран  *File*.
3. Перейдите к соответствующему файлу, выберите его и кликните на кнопку *Open*.
4. Проделайте то же самое для каждого из следующих двух растровых файлов, `DEM/SRTM.tif` и `rainfall/reprojected/rainfall.tif`.
5. Переименуйте растр SRTM на `DEM`, а растр осадков/rainfall на `Rainfall` (с заглавной буквы).

8.4.5 Изменение порядка слоев

Кликните и перетащите слои вверх и вниз в панели *Layers* для того, чтобы изменить порядок их отображения на карте, чтобы вы могли видеть как можно больше слоев.

После того как все данные загружены и соответствующим образом видны, можно начинать анализ. Сначала лучше сделать операцию обрезки. Это делается для того, чтобы никакая вычислительная мощность не тратилась впустую на вычисление значений в участках, которые в любом случае не будут использоваться.

8.4.6 Найти правильные районы

С учетом вышеупомянутой местности исследования нам необходимо ограничиться следующими районами:

- `Bellville`
- `Cape`
- `Goodwood`
- `Kuils River`
- `Mitchells Plain`
- `Simon Town`
- `Wynberg`

1. Кликните правой кнопкой мыши на слой `Districts` в панели *Layers*.
2. В появившемся меню выберите пункт меню *Filter…*. Появится диалоговое окно *Query Builder*.
3. Теперь вам надо создать запрос, чтобы выбрать только те районы, которые являются кандидатами:
    1. В списке *Fields* дважды кликните на поле `NAME_2`, чтобы оно появилось в *SQL where clause* текстовом поле ниже.
    2. Кликните на кнопку *IN* чтобы добавить его к SQL-запросу.
    3. Откройте скобки.
    4. Кликните на кнопку *All* под (в настоящее время пустая) списком *Values*.

       После небольшой паузы он заполнит список *Values* со значениями выбранного поля (`NAME_2`).
    5. Дважды кликните на значение `Bellville` в списке *Values* чтобы добавить его в запрос SQL.
    6. Добавьте запятую и дважды кликните, чтобы добавить район `Cape`.
    7. Повторите предыдущий шаг для остальных районов.
    8. Закройте скобки.

       Окончательный запрос должен быть таким (порядок расположения районов в скобках не имеет значения):

```
"NAME_2" in ('Bellville', 'Cape', 'Goodwood', 'Kuils River',
             'Mitchells Plain', 'Simon Town', 'Wynberg')
```

Примечание: Вы также можете использовать оператор `OR`; запрос будет выглядеть так:

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

---

9. Кликните на *OK* дважды.

   Районы, показанные на вашей карте, теперь ограничены теми, которые указаны в списке выше.

8.4.7 Обрезка растров

Теперь, когда вы создали интересующий вас район, вы можете обрезать растры.

1. Надо открыть диалоговое окно обрезки, выбрав пункт меню *Raster* ► *Extraction* ► *Clip Raster by Mask Layer…*
2. В раскрывающемся списке *Input layer* выберите слой выберите слой `DEM`.
3. В раскрывающемся списке *Mask layer* двыберите слой `Districts`.
4. Прокрутите вниз и укажите выходное местоположение в тексте поля *Clipped (mask)*, кликнув на кнопку … и выбрав *Save to File…*
   1. Перейдите в каталог `Rasterprac`,
   2. Введите имя файла - `DEM_clipped.tif`,
   3. Сохраните.
5. Убедись в том, что галочка  *Open output file after running algorithm* отмечена.
6. Кликните на *Run*.

   После завершения операции обрезки оставьте диалоговое окно *Clip Raster by Mask Layer* открытым для того, чтобы можно было повторно использовать местность обрезки.
7. Выберите растровый слой `Rainfall` в раскрывающемся списке *Input layer* и сохраните выход/результат как `Rainfall_clipped.tif`.
8. Не меняйте другие параметры. Оставьте все как есть и кликните на *Run*.
9. После завершения второй операции обрезки вы можете закрыть диалоговое окно *Clip Raster by Mask Layer*.
10. Сохраните карту.

Выровнять растры

Для нашего анализа нам нужно, чтобы растры имели одинаковые CRS и были выровнены.

Сначала мы меняем разрешение наших данных по осадкам на 30 метров (размер пикселя):

1. In the *Layers* panel, ensure that `Rainfall_clipped` is the active layer (i.e., it is highlighted by having been clicked on)
2. Click on the *Raster* ► *Projections* ► *Warp (Reproject)…* menu item to open the *Warp (Reproject)* dialog
3. Under *Resampling method to use*, select *Bilinear* from the drop down menu
4. Set *Output file resolution in target georeferenced units* to `30`
5. Scroll down to *Reprojected* and save the output in your `rainfall/reprojected` directory as `Rainfall30.tif`.
6. Убедись в том, что галочка  *Open output file after running algorithm* отмечена.

Затем выравниваем ЦМР:

1. In the *Layers* panel, ensure that `DEM_clipped` is the active layer (i.e., it is highlighted by having been clicked on)
2. Click on the *Raster* ► *Projections* ► *Warp (Reproject)…* menu item to open the *Warp (Reproject)* dialog
3. Under *Target CRS*, select *Project CRS: EPSG:32733 - WGS 84 / UTM zone 33S* from the drop down menu

4. Under *Resampling method to use*, select *Bilinear* from the drop down menu
5. Set *Output file resolution in target georeferenced units* to `30`
6. Scroll down to *Georeferenced extents of output file to be created*. Use the button to the right of the text box to select *Calculate from Layer* ► *Rainfall30*.
7. Scroll down to *Reprojected* and save the output in your `DEM/reprojected` directory as `DEM30.tif`.
8. Убедись в том, что галочка  *Open output file after running algorithm* отмечена.

Чтобы соответствующим образом увидеть, что происходит, необходимо изменить условные обозначения слоев.

8.4.8 Изменение условных обозначений векторных слоев

1. В панели *Layers* кликните правой кнопкой мыши на слой *Streets*.
2. Выберите *Properties* из появившегося меню.
3. Перейдите на вкладку *Symbology* в появившемся диалоговом окне.
4. Кликни на *Fill* в верхней части.
5. Выберите условное обозначение в списке ниже или настройте новый (цвет, прозрачность, …).
6. Кликните на кнопку *OK* для того, чтобы закрыть диалоговой окно *Layer Properties*. Это изменит визуализацию слоя *Streets*.
7. Выполните аналогичный процесс в отношении слоя *Zoning* и выберите для него подходящий цвет.

8.4.9 Изменение условных обозначений растровых слоев

Условные обозначения растрового слоя несколько отличаются.

1. Надо открыть диалоговое окно *Properties* для растрового слоя *Rainfall30*.
2. Перейдите на вкладку *Symbology*. Вы заметите, что это диалоговое окно сильно отличается от той, используемой для векторных слоев.
3. Расширьте *Min/Max Value Settings*.
4. Убедитесь, что выбрана кнопка *Mean +/- standard deviation*.
5. Убедитесь, что значение в соответствующем поле равно `2.00`.
6. В отношении *Contrast enhancement* убедитесь, что там указано *Stretch to MinMax*.
7. *Color gradient* поменяйте на *White to Black*.
8. Кликните *OK*.

   Растр `Rainfall30`, если он виден, должен поменять цвета, и вы должны увидеть разные значения яркости для каждого пикселя.
9. Повторите этот процесс для слоя `DEM30`, но настройте стандартные отклонения, используемые для растяжения, на `4.00`.

8.4.10 Вычистка карты

1. Удалите исходные слои `Rainfall` и `DEM`, а также `Rainfall_clipped` и `DEM_clipped` в панели *Layers*:
   - Кликните эти слои правой кнопкой мыши и выберите *Remove*.

   Примечание: Это не приведет к удалению данных с вашего устройства хранения, а просто уберет их с вашей карты.

2. Сохраните карту.
3. Теперь вы можете скрыть векторные слои, сняв флажок с ячейки рядом с ними в панели *Layers*. Это ускорит выдачу карты и сэкономит вам время.

8.4.11 Создание отмывки

Чтобы создать отмывку вам надо будет использовать алгоритм, который был написан для этой цели.

1. В панели *Layers* убедитесь, что `DEM30` является активным слоем (т. е. выделено щелчком).
2. Надо кликнуть на меню *Raster* ► *Analysis* ► *Hillshade...* чтобы открыть диалоговое окно *Hillshade*.
3. Прокрутите вниз до *Hillshade* и сохраните результат/выход в вашем каталоге `Rasterprac` как `hillshade.tif`.
4. Убедись в том, что галочка *Open output file after running algorithm* отмечена.
5. Кликните на *Run*.
6. Подождите, пока он закончит обработку.

Новый слой `hillshade` появляется в панели *Layers*.

1. Кликните правой кнопкой мыши на слой `hillshade` в панели *Layers* и вызовите диалоговое окно *Properties*.
2. Кликни на вкладку *Transparency* и настройте ползунок *Global Opacity* на `20%`.
3. Кликните *OK*.
4. Обратите внимание на эффект, когда прозрачная отмывка накладывается на обрезанный ЦМР. Вам придется, возможно, изменить порядок ваших слоев или выключить слой `Rainfall30`, чтобы увидеть эффект.

8.4.12 Уклон

1. Кликни на пункт меню *Raster* ► *Analysis* ► *Slope...* для того, чтобы открыть алгоритм *Slope*.
2. Выберите `DEM30` в качестве *Input layer*.
3. Отметьте галочку *Slope expressed as percent instead of degrees*. Уклон может быть выражен в различных единицах (процентах или градусах). Наши критерии предполагают, что наше растение растет на склонах с уклоном от 15% до 60%. Поэтому нам нужно убедиться, что наши данные об уклоне выражены в процентах.
4. Укажите соответствующее имя файла и расположение для вашего выхода/результата.
5. Убедись в том, что галочка *Open output file after running algorithm* отмечена.
6. Кликните на *Run*.

Изображение уклона было вычислено и добавлено на карту. Как обычно, он отображается в оттенках серого. Измените условное обозначение на более красочный свет:

1. Откройте диалоговое окно слоя *Properties* (как обычно через меню правого щелчка слоя).
2. Кликни на вкладку *Symbology*.
3. Там где сказано *Singleband gray* (в раскрывающемся меню *Render type*) измените его на *Singleband pseudocolor*.
4. Выберите *Mean +/- standard deviation x* для *Min / Max Value Settings* со значением `2.0`.
5. Выберите подходящий *Color ramp*.
6. Кликните на *Run*.

8.4.13 Попробуйте сами: Поработать с аспектом

Примените тот же подход, что и вычисления уклона, выбрав *Aspect…* в меню *Raster* ► *Analysis*.

Помните, что надо периодически сохранять проект.

8.4.14 Пере-классификация растров

1. Выберите *Raster* ► *Raster calculator…*
2. Укажите свой каталог `Rasterprac` как место для *Output layer* (кликни на кнопку …) и сохраните как `slope15_60.tif`.
3. Убедитесь, что ячейка *Open output file after running algorithm* отмечена.

   В списке *Raster bands* слева вы увидите все растровые слои в ваше панели *Layers*. Если ваш слой Уклона называется *slope*, он будет указан как `slope@1`. Обозначение полосы 1 растра уклонов.
4. Уклон должен быть между `15` и `60` градусами.

   Используя элементы списка и кнопки в интерфейсе, надо создать следующее выражение:

   ```
   (slope@1 > 15) AND (slope@1 < 60)
   ```
5. Настройте поле *Output layer* в соответствующее место и имя файла.
6. Кликните на кнопку *Run*.

Теперь найдите правильный аспект (направленный на восток: между `45` и `135` градусами), используя тот же подход.

1. Постройте следующее выражение:

   ```
   (aspect@1 > 45) AND (aspect@1 < 135)
   ```

Вы поймете, что у вас все получилось, когда все уклоны, направленные на восток, станут белыми в выходном растре (как если бы они были освещены утренним солнечным светом).

Вам надо найти определить правильное количество осадков (больше, чем `1000` мм) таким же образом. Используйте следующее выражение:

```
Rainfall30@1 > 1000
```

Теперь, когда у вас есть все три критерия в отдельных растрах, вам нужно объединить их, чтобы определить, какие участки соответствуют всем критериям. Для этого растры надо умножить друг на друга. Когда это происходит, все перекрывающиеся пиксели со значением `1` сохранят значение `1` (т.е. местоположение соответствует критериям), но если пиксель в любом из трех растров имеет значение `0` (т.е. местоположение не соответствует критериям), тогда оно будет `0` в результате. Таким образом, результат будет содержать только перекрывающиеся участки, которые соответствуют всем необходимым критериям.

8.4.15 Объединение растров

1. Надо открыть *Калькулятор растра* (*Raster* ► *Raster Calculator…*).
2. Создайте следующее выражение (с соответствующими названиями ваших слоев):

```
[aspect45_135] * [slope15_60] * [rainfall_1000]
```

3. Настройте выходное местоположение н каталог `Rasterprac`.
4. Назовите выходной растр `aspect_slope_rainfall.tif`.
5. Убедитесь, что отмечена галочка  *Open output file after running algorithm*.
6. Кликните на *Run*.

В новом растре теперь правильно отображаются участки, соответствующие всем трем критериям.

Сохраните проект..

Следующий критерий, в отношении которого должно быть обеспечено соответствие, это то, что участок должен находиться на расстоянии в `250` м от городской застройки. Мы обеспечим соответствие этому требованию тем, что вычисляемые нами участки будут находиться в сельской местности и в `250` м и более от края участка. Поэтому, нам надо сначала найти список всех сельских населённых пунктов.

8.4.16 Поиск сельских населенных пунктов

1. Надо скрыть все слои в панели *Layers*.
2. Покажите векторный слой `Zoning`.
3. Кликните на него правой кнопкой мыши и откройте диалоговое окно *Attribute Table*. Обратите внимание, что там есть множество различных способов зонирования земли. Нам надо изолировать сельские населенные пункты. Закройте таблицу атрибутов/атрибутов.
4. Кликните правой кнопкой мыши на слой `Zoning` и выберите *Filter…* для того, чтобы вызвать диалоговое окно *Query Builder*.
5. Создайте следующий запрос:

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

   См. предыдущие инструкции, если вы где-то застрянете.

6. Надо кликнуть на кнопку *OK* для того, чтобы закрыть диалоговое окно *Query Builder*. Запрос должен вернуть одну функцию.

Вы увидите сельские полигоны в слое `Zoning`. Вам надо их сохранить.

1. В меню после нажатия правой кнопки мыши на `Zoning`, надо выбрать *Export* ► *Save Features As…*
2. Сохраните свой слой внутри каталога `Rasterprac`.
3. Назовите выходной файл `rural.shp`.
4. Кликните *OK*.
5. Сохраните проект.

Теперь надо исключить участки, которые находятся на расстоянии `250m` от окраины сельской местности. Это можно сделать, создав отрицательный буфер, как описано ниже.

8.4.17 Создание отрицательного буфера

1. Кликните на пункт меню *Vector* ► *Geoprocessing Tools* ► *Buffer…*
2. В диалоговом окне, которое появится, выберите слой `rural` в качестве входного векторного слоя (галочка в *Selected features only* не должна быть активной).
3. Настройте *Distance* на `-250`. Отрицательное значение означает, что буфер будет внутренним. Убедитесь, что в раскрывающемся меню метры указаны в качестве единиц измерения.
4. Отметьте галочку *Dissolve result*.
5. В *Buffered*, поместите выходной файл в каталоге в `Rasterprac` и назовите его `rural_buffer.shp`.
6. Кликните на кнопку *Save*.
7. Кликните на кнопку *Run* и дождитесь завершения обработки.
8. Закройте диалоговое окно *Buffer*.

   Убедитесь, что ваш буфер отработал правильно, посмотрев как отличается слой `rural_buffer` от слоя `rural`. Вам, скорее всего, надо будет изменить порядок изображения, чтобы увидеть разницу.
9. Уберите слой `rural`.
10. Сохраните проект.

Вам теперь необходимо объединить ваш векторный свой `rural_buffer` с растром `aspect_slope_rainfall`. Для того, чтобы объединить их, нам надо изменить формат данных одного из слоев. В этом случае вам надо векторизовать растр, так как векторные слои более удобны, когда мы хотим вычислить площади.

8.4.18 Векторизация растра

1. Надо кликнуть на пункт меню *Raster* ► *Conversion* ► *Polygonize (Raster to Vector)…*
2. Выберите растр `aspect_slope_rainfall` как *Input layer*.
3. Настройте *Name of the field to create* на `suitable` (название поля по умолчанию `DN` - Данные цифрового номера).
4. Сохраните выход. Внутри *Vectorized* выберите *Save file as*. Установите местоположение в `Rasterprac` и назовите файл `aspect_slope_rainfall_all.shp`.
5. Убедитесь, что отмечена галочка *Open output file after running algorithm*.
6. Кликните на *Run*.
7. Закройте диалоговое окно после завершения обработки.

Все участки растра были векторизованы и вам надо выбрать только те участки, которые имеют значение `1` в поле `suitable` (Цифровой номер).

1. Откройте диалоговое окно *Query Builder* (кликните правой кнопкой мыши - *Filter…*) для нового векторного слоя.
2. Создайте этот запрос:

```
"suitable" = 1
```

3. Кликните *OK*.
4. После того, как вы будете уверены, что запрос завершен (и показаны только те участки, которые соответствуют всем трем критериям, т.е. со значением `1`), создайте новый векторный файл на основе результатов, используя *Export –> Save Features As…* через меню при нажатии правой кнопки мыши после нажатия на слой.

5. Сохраните файл в каталоге `Rasterprac`.
6. Назовите файл `aspect_slope_rainfall_1.shp`.
7. Уберите слой `aspect_slope_rainfall_all` с вашей карты.
8. Сохраните свой проект.

При использовании алгоритма для векторизации растра, иногда алгоритм выдает так называемые «неверные геометрии», то есть пустые полигоны или полигоны с ошибками, которые трудно проанализировать в будущем. Нам надо использовать инструмент “Fix Geometry” (исправление геометрии).

8.4.19 Исправление геометрии

1. В *Processing Toolbox* надо искать «Fix geometries» и выполнить его *Execute…*.
2. Для *Input layer* надо выбрать `aspect_slope_rainfall_1`.
3. Под *Fixed geometries* надо выбрать *Save file as* и сохраните выход в `Rasterprac` и назвать файл `fixed_aspect_slope_rainfall.shp`.
4. Убедитесь, что отмечена галочка  *Open output file after running algorithm*.
5. Кликните на *Run*.
6. Закройте диалоговое окно после завершения обработки.

Вы векторизовали растр и исправили геометрию и вы теперь сможете объединить критерии аспекта, уклона и осадков с критериями расстояния от населенных пунктов, путем определения пересечения слоя `fixed_aspect_slope_rainfall` и слоя `rural_buffer`.

8.4.20 Определение пересечения векторов

1. Кликните в пункт меню *Vector* ► *Geoprocessing Tools* ► *Intersection…*
2. В диалоговом окне, которое появится, выберите слой `rural_buffer` как *Input layer*.
3. Для *Overlay layer* выберите слой `fixed_aspect_slope_rainfall`.
4. В *Intersection* поместите выходной файл в каталоге `Rasterprac`.
5. Назовите выходной файл `rural_aspect_slope_rainfall.shp`.
6. Кликните на кнопку *Save*.
7. Кликните на кнопку *Run* и дождитесь завершения обработки.
8. Закройте диалоговое окно *Intersection*.

   Убедитесь, что ваше пересечение сработало правильно, отметив, что остались только перекрывающиеся участки.
9. Сохраните проект.

Следующим критерием в списке является то, что размер участка должен быть больше, чем `6000` m^2. Вам надо вычислить площади полигона для того, чтобы определить участки, которые имеют подходящий размер для этого проекта.

8.4.21 Вычисление площади для каждого полигона (многоугольника)

1. Откройте меню нового векторного слоя через нажатие правой кнопки мыши.
2. Выберите *Open attribute table*.
3. Кликните на кнопку  Toggle editing в верхнем левом углу таблицы или кликните `Ctrl+e`.
4. Кликните на кнопку  Open field calculator в панели инструментов в верхней части таблицы или кликните `Ctrl+i`.
5. В появившемся диалоговом окне убедитесь, что отмечена галочка *Create new field* и настройте *Output field name* на `area`. Тип поля выхода должен быть десятичным числом (действительным). Настройте *Precision* на `1` (одно десятичное).
6. В *Expression* наберите:

   ```
   $area
   ```

   Это означает, что калькулятор поля вычислит площадь каждого полигона в векторном слое и затем заполнит новый целочисленный столбец (называемый `area`) с вычисленным значением.
7. Кликните *OK*.
8. Проделайте то же самое в отношении другого нового поля под названием `id`. В *Field calculator expression* наберите:

   ```
   $id
   ```

   Это обеспечит то, что каждый полигон будет иметь уникальный идентификационный номер для целей идентификации.
9. Кликните снова на  Toggle editing и сохраните ваши изменения, если будет предложено сделать так.

8.4.22 Выбор участка заданного размера

Теперь, когда вам известны участки:

1. Создайте запрос (как обычно), чтобы выбрать только те полигоны, которые больше чем `6000` m^2. Запрос такой:

   ```
   "area" > 6000
   ```
2. Сохраните выбор в каталоге `Rasterprac` как новый векторный слой с названием `suitable_areas.shp`.

У вас теперь подходящие участки, отвечающие всем критериям среды обитания для редкого растения финбос, из которых вы должны выбрать четыре участка, которые являются ближайшими к Кейптаунскому университету.

8.4.23 Оцифровка Кейптаунского университета

1. Создайте новый векторный слой в каталоге `Rasterprac` , как и раньше, но на этот раз используйте *Point* как *Geometry type* и назовите его `university.shp`.
2. Убедитесь в правильности ССК (`Project CRS:EPSG:32733 - WGS 84 / UTM zone 33S`).
3. Завершите создание нового слоя (кликните на кнопку *OK*).
4. Скройте все слои, кроме нового слоя `university` и слоя `Streets`.
5. Добавьте фоновую карту (OSM):

1. Заходите на панель *Browser* и перейдите к *XYZ Tiles* ► *OpenStreetMap*.
2. Перетащите и разместите `OpenStreetMap` в нижнюю часть панели *Layers*.

В своем интернет-браузере найдите расположение Кейптаунского университета. Учитывая уникальную топографию Кейптауна, университет находится в очень узнаваемом месте. Прежде чем вы вернетесь в QGIS, обратите внимание на то, где находится университет и что находится рядом с ним.

6. Убедитесь, что слой `Streets` отмечен, в слой `university` выделен в панели *Layers*.
7. Перейдите в пункт меню *View* ► *Toolbars* и убедитесь, что выбран *Digitizing*. После этого вы должны увидеть на панели инструментов значок с карандашом (Toggle editing). Это кнопка *переключения редактирования*.
8. Кликните на кнопку *Toggle editing* для того, чтобы войти в *режим редактирования*. Это позволит вам редактировать векторный слой.
9. Кликните на кнопку Add Point Feature, которая должна быть рядом с кнопкой Toggle editing.
10. Активировав инструмент *Add feature*, кликните левой кнопкой мыши по наиболее точной оценке местоположения Кейптаунского университета.
11. Укажите произвольное целое число `id`.
12. Кликните *OK*.
13. Кликните на кнопку Save Layer Edits.
14. Кликните на кнопку *Toggle editing* для того, чтобы остановить режим редактирования.
15. Сохраните проект.

8.4.24 Поиск ближайших мест к Университету Кейптауна

1. Перейдите в *Панель инструментов обработки*, найдите *Join Attributes by Nearest* алгоритм (*Vector general* ► *Join Attributes by Nearest*) и выполните его.
2. *Input layer* должен быть `university`, а *Input layer 2* `suitable_areas`.
3. Укажите соответствующее выходное местоположение и название (*Joined layer*).
4. Настройте *Maximum nearest neighbors* на `4`.
5. Убедитесь, что отмечена галочка *Open output file after running algorithm*.
6. Для остальных параметров оставьте значения по умолчанию.
7. Кликните на *Run*.

Выходной точечный слой будет содержать четыре объекта - все они будут иметь местоположение университета и его атрибуты, а также атрибуты близлежащих подходящих участков (включая `id`) и расстояние до этого участка.

1. Откройте таблицу атрибутов результата соединения.
2. Отметьте `id` четырех ближайших подходящих участков, а затем закройте таблицу атрибутов.
3. Откройте таблицу атрибутов слоя `suitable_areas`.
4. Создайте запрос, чтобы выбрать четыре подходящих участка, ближайших к университету (выбирая их с помощью поля `id`).

Это и есть окончательный ответ на вопрос исследования.

Создайте полностью маркированный макет, который включает полупрозрачный слой отмывки поверх растра по вашему выбору (например, ЦМР или растра уклона). Также включите слой университета и `suitable_areas` с выделенными четырьмя подходящими участками, ближайшими к университету. При создании выходной карты следуйте всем передовым методам картографии.

ГЛАВА 9

Модуль: Плагины

Плагины дают возможность расширить функциональность QGIS. В этом модуле мы вам покажем, как активировать и использовать плагины.

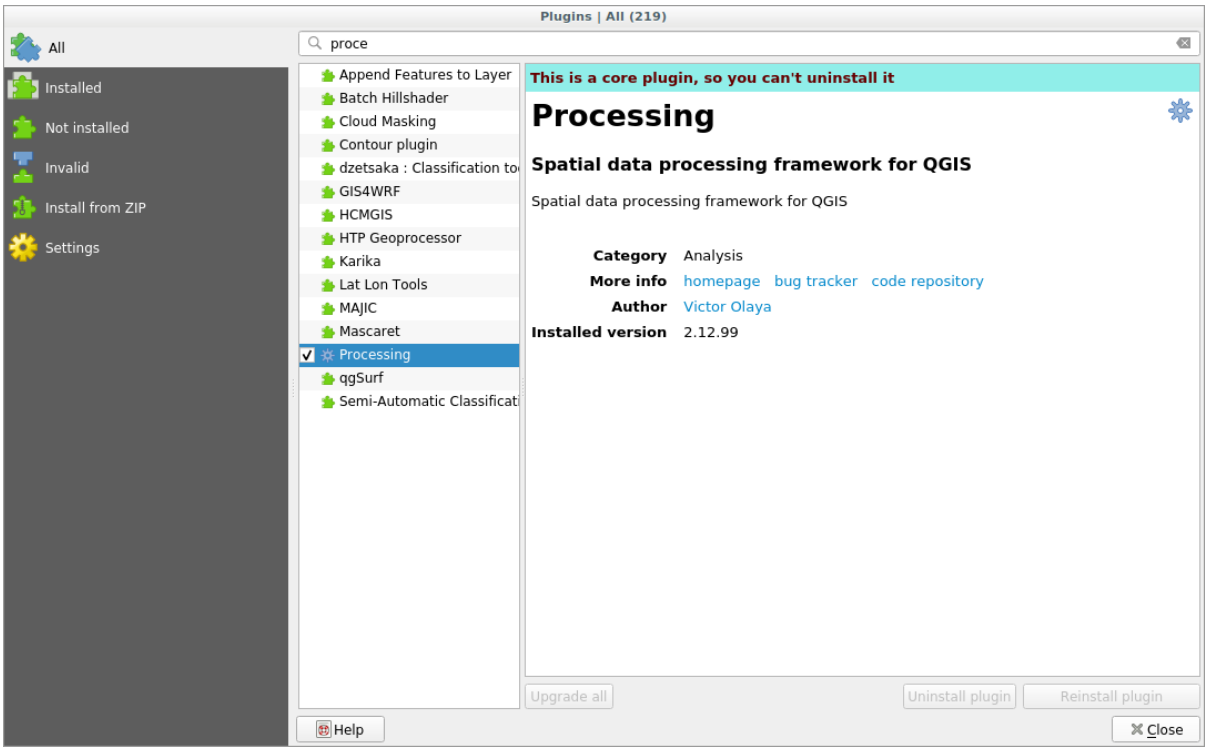
9.1 Занятие: Установка и управление плагинами

ТДля того, чтобы начать использовать плагины, вам нужно знать, как их загрузить, устанавливать и активировать. Вы научитесь использовать *Plugin Installer* и *Plugin Manager*.

Цель этого урока: Понять и использовать систему плагинов QGIS.

9.1.1  Идем дальше: Управление плагинами

1. Для того, чтобы открыть *Plugin Manager* надо кликнуть на пункту меню *Plugins* ► *Manage and Install Plugins*.
2. В открывшемся диалоговом окне вам надо найти плагин *Processing* plugin:

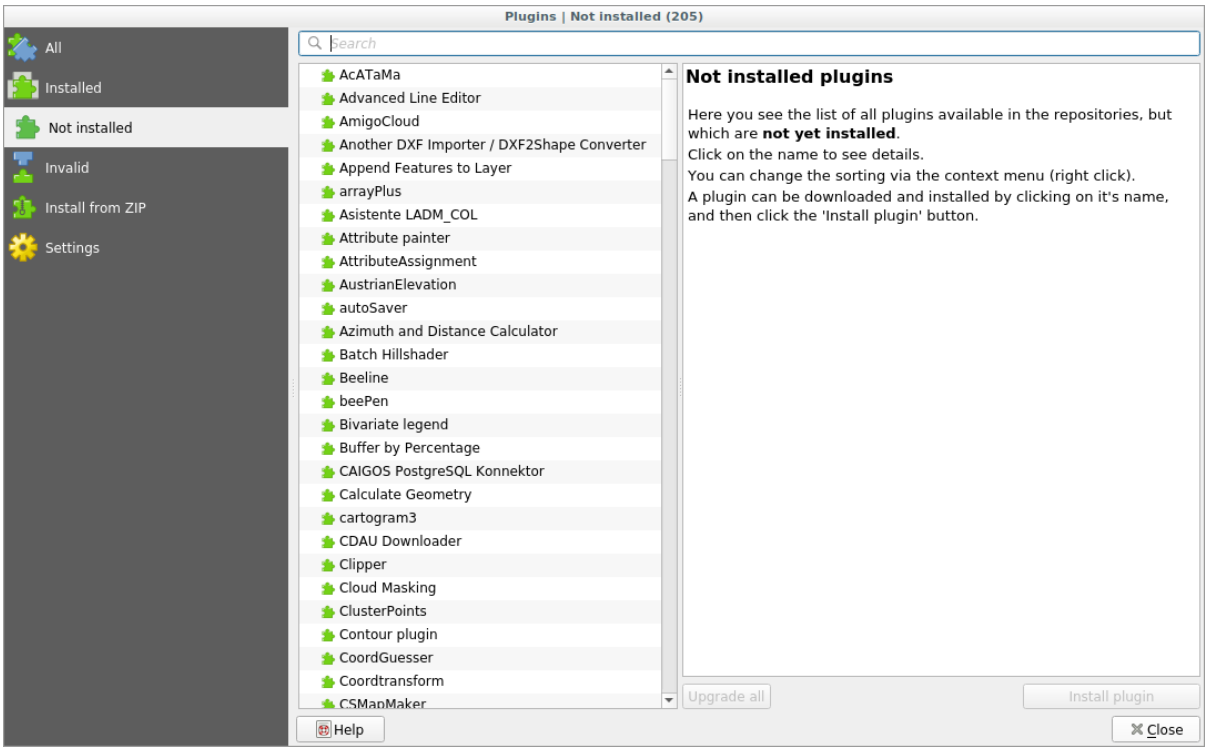


3. Кликните на ячейку рядом с этим плагином и снимите флажок, чтобы деактивировать его.
4. Кликните на кнопку *Close*.
5. Посмотрите на меню, вы заметите, что меню *Processing* исчезло. Это означает, что многие из функций обработки, которыми вы пользовались ранее, исчезли! Например, посмотрите на меню *Vector* ► и *Raster* ►. Это из-за того, что они являются частью плагина *Processing*, который необходимо активировать, чтобы использовать их.
6. Откройте снова *Plugin Manager* и активируйте плагин *Processing*, отметив галочку в ячейке рядом с ним.
7. *Close* диалоговое окно. Меню *Processing* функции снова должны стать доступными.

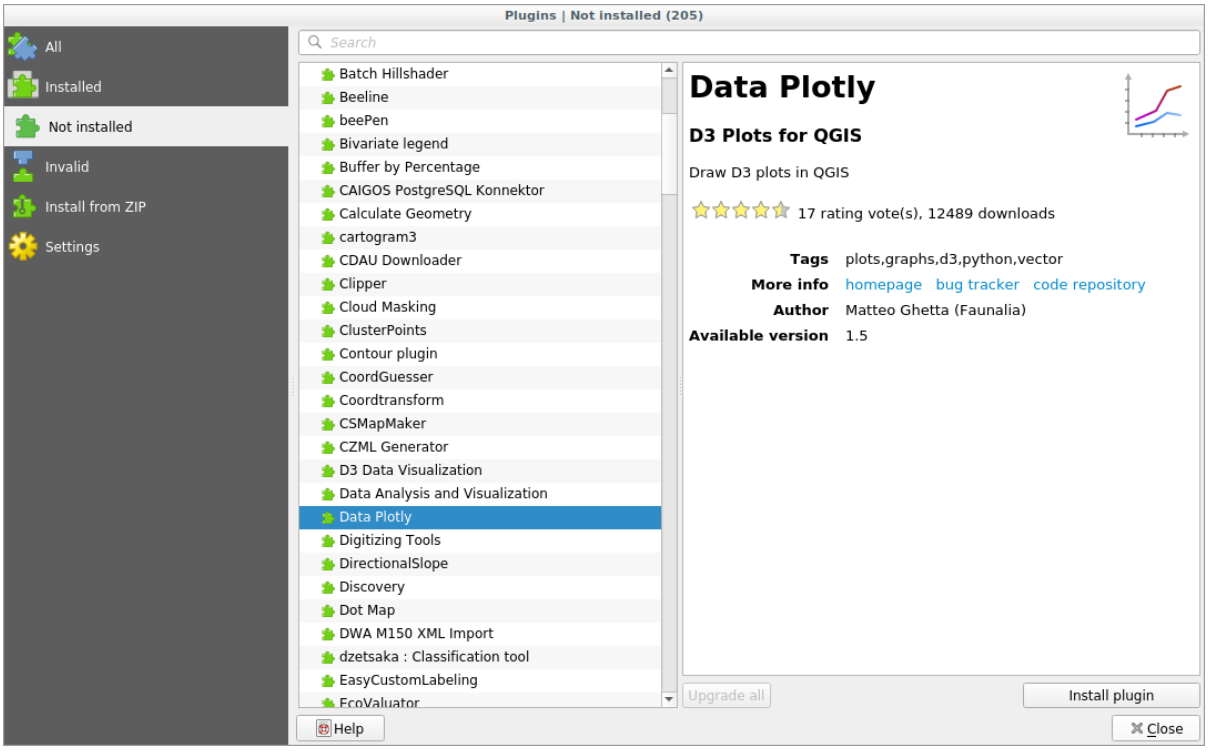
9.1.2 Идем дальше: Установка новых плагинов

Список плагинов, которые вы можете активировать и деактивировать, получен из плагинов, которые вы в настоящее время установили. Для того, чтобы установить новые плагины вам надо:

1. Выберите опцию *Not Installed* в диалоговом окне *Plugin Manager*. Плагины, доступные для установки, будут показаны в списке. Этот список будет отличаться в зависимости от вашей существующей настройки системы.



2. Найдите сведения о плагине, выбрав его в списке.



3. Установите те плагины, которые вас интересуют, нажав на кнопку *Install Plugin* под панелью сведений о плагине.

Примечание: Если в плагине есть ошибка, он будет в списке закладки *Invalid*. Вы сможете связаться с владельцем плагина, чтобы решить проблему.

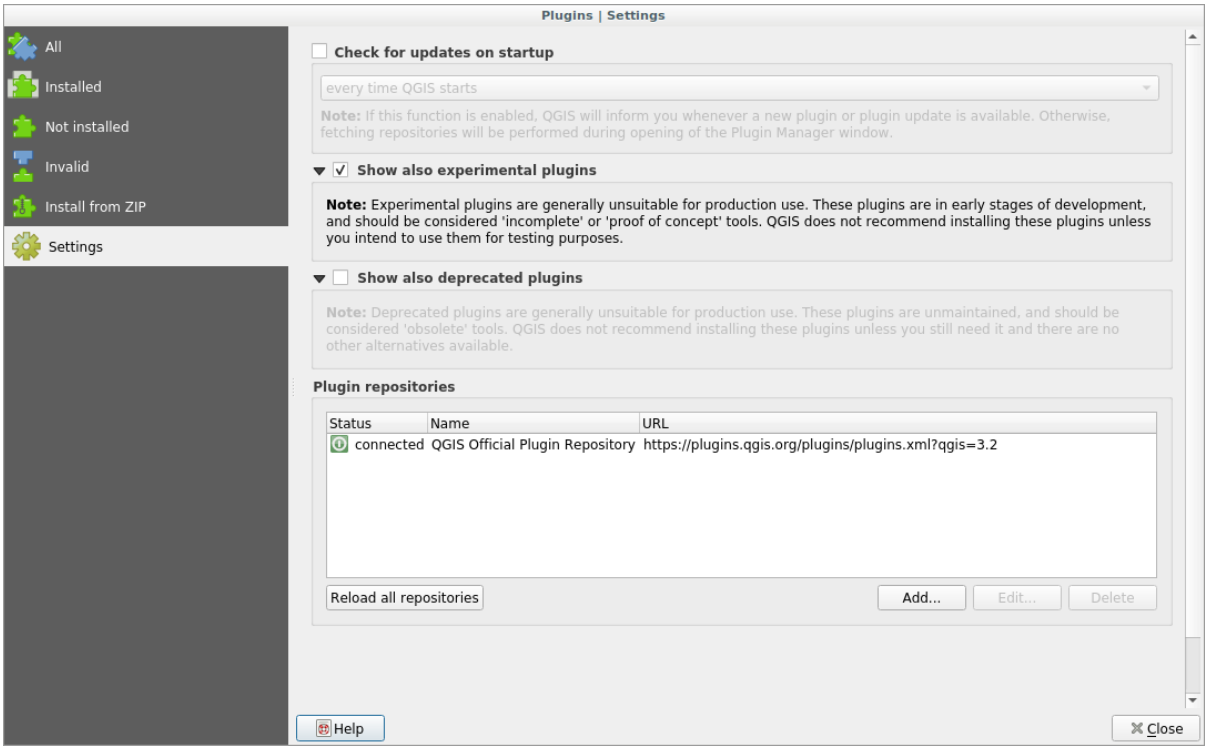
9.1.3 Идем дальше: Настройка дополнительных репозиториев плагинов

Набор плагинов, доступных для вас, зависит от того, на какие *репозитории* плагинов настроена ваша система.

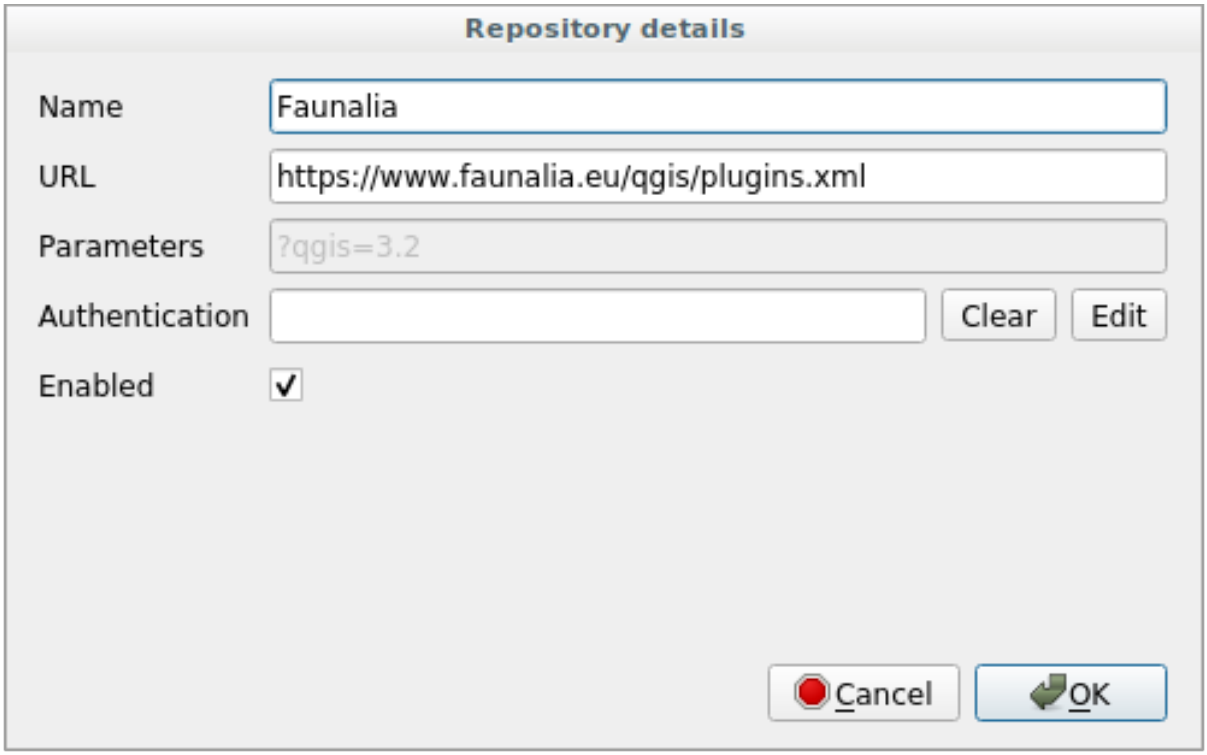
Плагины QGIS хранятся в онлайн репозиториях. По умолчанию только `официальный репозиторий `_ является активным и это означает, что вы можете получить доступ только к тем плагинам, которые там опубликованы. С учетом разнообразия доступных инструментов, этот репозиторий должен удовлетворить большинство ваших потребностей.

Тем не менее, вы сможете попробовать больше плагинов, чем те, которые доступны по умолчанию. Во-первых, вам надо настроить дополнительные репозитории. Для того чтобы сделать это, необходимо:

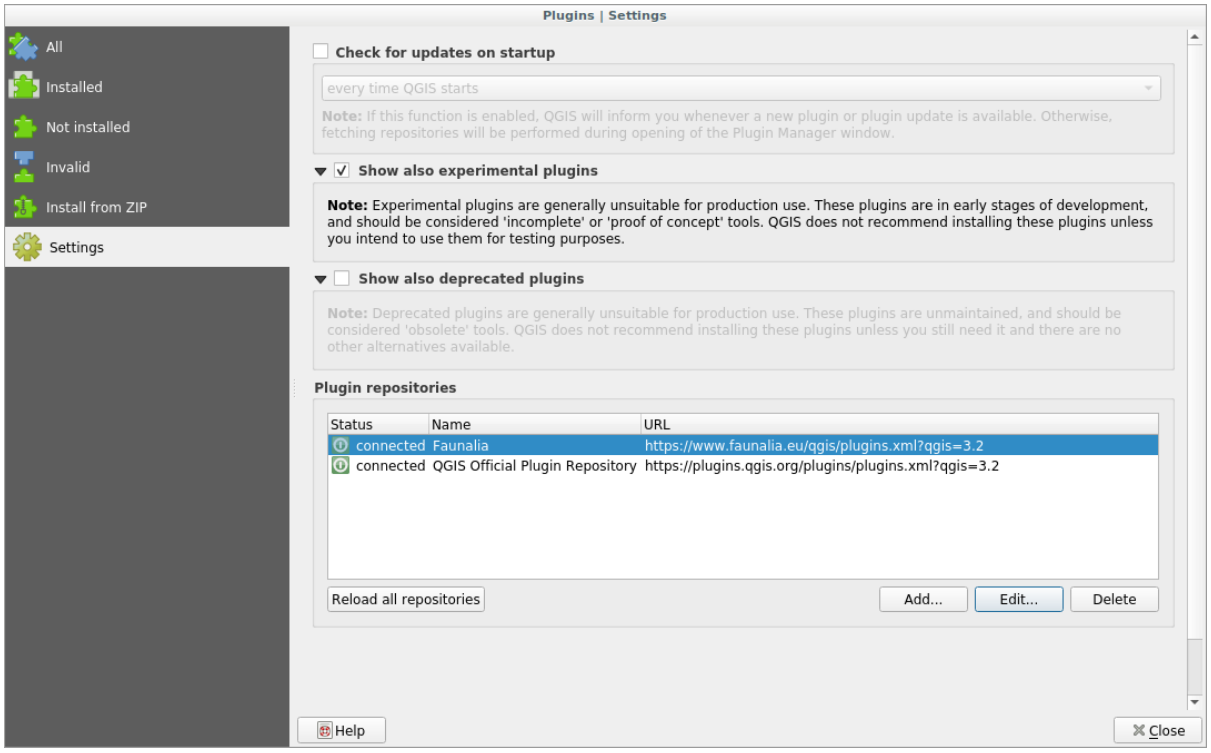
1. Открыть вкладку *Settings* в диалоговом окне *Plugin Manager*.



2. Надо кликнуть на *Add* чтобы найти и добавить новый репозиторий.
3. Укажите название или адрес сайта нового репозитория, который вы хотите настроить, и убедитесь, что отмечена галочка в ячейке *Enabled*.



4. Вы теперь увидите репозиторий нового плагина в списке настроенных репозиториев плагинов.



5. Вы также можете выбрать опцию отображения Экспериментальных плагинов, выбрав флажок *Show also experimental plugins*.
6. Если вы теперь вернетесь обратно на вкладку *Not Installed*, то вы увидите, что появились дополнительные плагины для установки.
7. Для того, чтобы установить плагин, кликните на него в списке, а затем на кнопку *Install plugin*.

9.1.4 В заключении

Установка плагинов в QGIS должна быть легкой и эффективной!

9.1.5 Что дальше?

Далее мы расскажем вам о некоторых полезных плагинах в качестве примеров.

9.2 Занятие: Полезные плагины QGIS

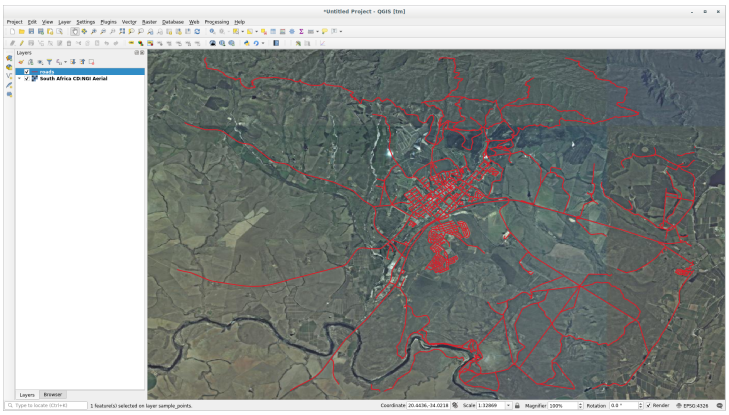
Теперь, когда вы можете устанавливать, включать и отключать плагины, давайте посмотрим, как это может вам помочь на практике. Рассмотрим несколько примеров о полезных плагинах.

Цель этого урока: Ознакомиться с интерфейсом плагина и узнать больше о некоторых полезных плагинах.

9.2.1  Идем дальше: Плагин QuickMapServices

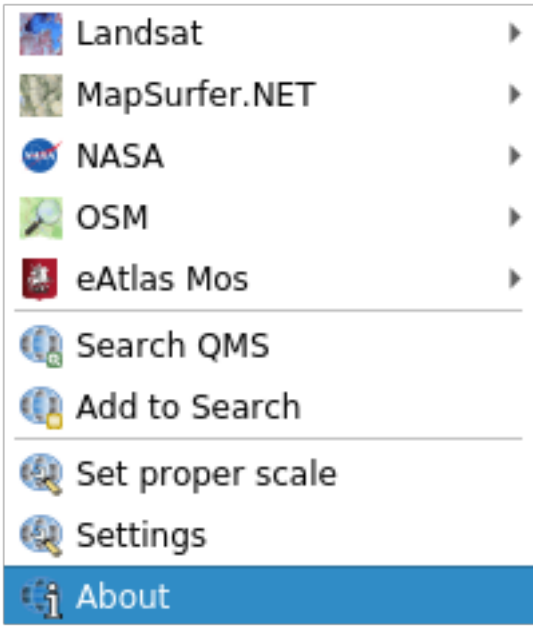
Плагин QuickMapServices - это простой и легкий в использовании плагин, который добавляет базовые карты в ваш проект QGIS. У него много разных опций и настроек. Давайте изучим некоторые из его функций.

1. Начните новую карту и добавьте слой *roads* из Геопакета `training_data`.
2. Установите плагин **QuickMapServices**.
3. Откройте вкладку поиска плагина, нажав на *Web* ► *QuickMapServices* ► *Search QMS*. Эта опция плагина даст вам возможность отфильтровать имеющиеся базовые карты по текущему экстенту холста карты.
4. Кликните на *Filter by extent* и вы должны увидеть одну доступную услугу.
5. Кликните на кнопку *Add* рядом с картой, чтобы загрузить ее.
6. Базовая карта будет загружена, и у вас будет спутниковый фон для карты.



Плагин QuickMapServices делает доступными множество базовых карт.

1. Закройте панель *Search QMS*, которую мы открыли раньше.
2. Кликните еще раз на *Web* ► *QuickMapServices*. В первом меню перечислены различные поставщики карт с доступными картами:



Но есть еще кое-что.

Если карты, доступные по умолчанию, вам недостаточны, вы сможете добавить других поставщиков карт.

1. Кликните на *Web* ► *QuickMapServices* ► *Settings* и перейдите во вкладку *More services*.
2. Внимательно прочтите сообщение на этой вкладке и, если вы согласны, кликните на кнопку *Get Contributed pack*.

Если вы сейчас откроете меню *Web* ► *QuickMapServices*, то вы увидите, что поставщиков карт стало больше. Выберите того, кто лучше всех соответствует вашим потребностям!

9.2.2  Идем дальше: Плагин QuickOSM

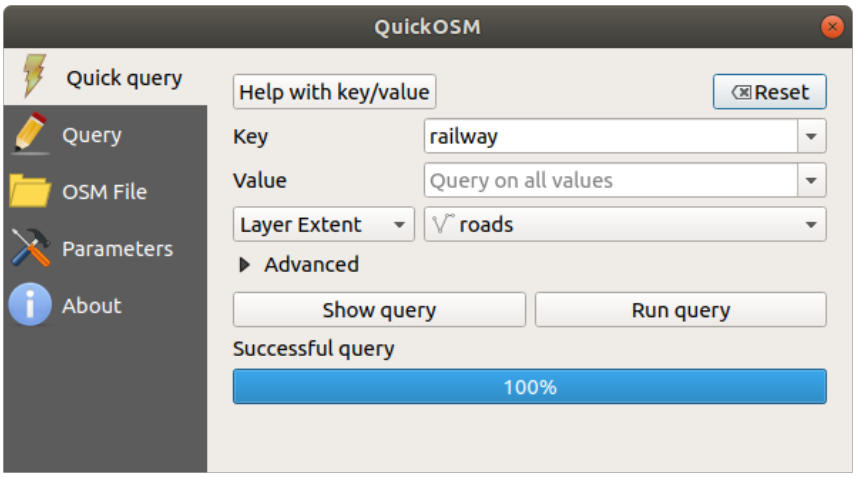
Благодаря невероятно простому интерфейсу плагин QuickOSM дает вам возможность загружать данные `OpenStreetMap `_ data.

1. Начните новый пустой проект и добавьте слой *roads* из Геопакета `training_data`.
2. Установите плагин **QuickOSM**. Плагин добавляет две новые кнопки в панель инструментов QGIS и он доступен в меню *Vector* ► *QuickOSM*.
3. Откройте диалоговое окно QuickOSM. В плагине есть много разных вкладок: мы будем использовать *Quick Query*.
4. Вы сможете загрузить определенные функции, выбрав общий *Key* или конкретный набор *Key* и *Value*.

   Совет: Если вы не знакомы с системой *Key* и *Value*, кликните на кнопку *Help with key/value*. Откроется веб-страница с полным описанием концепции OpenStreetMap.

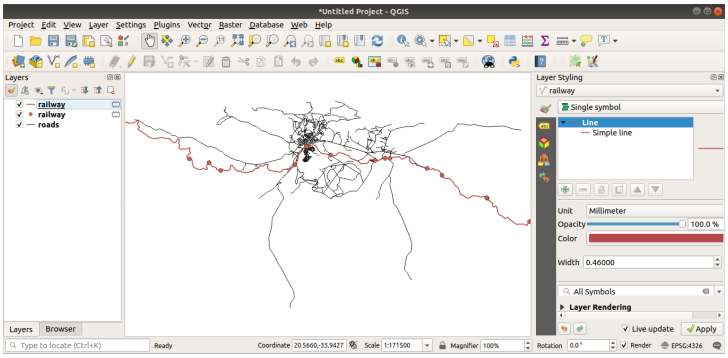
   ---

5. Найдите *railway* в меню *Key* и пусть *Value* будет пустым: и так, мы загружаем все объекты *railway* без указания каких-либо значений.
6. Выберите *Layer Extent* в следующем раскрывающемся меню и выберите *roads*.
7. Кликните на кнопку *Run query*.



Через несколько секунд плагин загрузит все объекты, отмеченные в OpenStreetMap как `railway` и загрузит их прямо на карту.

Ничего больше! Все слои загружены в условные обозначения и отображаются на холсте карты.



Предупреждение: QuickOSM создает временный слой при загрузке данных. Если вы хотите сохранить их навсегда, кликните на  рядом со слоем и выберите нужные вам опции. В качестве альтернативы вы можете открыть меню `Advanced` в QuickOSM и выбрать, где сохранить данные в меню *Directory*.

9.2.3  Идем дальше: Механизм запросов QuickOSM

Самый быстрый способ загрузить данные из плагина QuickOSM это использование вкладки *Quick query* и настройка небольших параметров. Но если вам нужны более конкретные данные?

Если вы являетесь мастером по запросам OpenStreetMap, вы сможете использовать плагин QuickOSM также с вашими личными запросами.

QuickOSM имеет невероятный анализатор данных, который в сочетании с удивительным механизмом запросов Overpass, позволит вам загружать данные с вашими конкретными потребностями.

Например: мы хотим загрузить горные вершины, относящиеся к определенной горной местности, известной как `Dolomites `_ .

Вы не можете решить эту задачу с помощью вкладки *Quick query*, вы должны более четко прописать свой собственный запрос. Давайте попробуем это сделать.

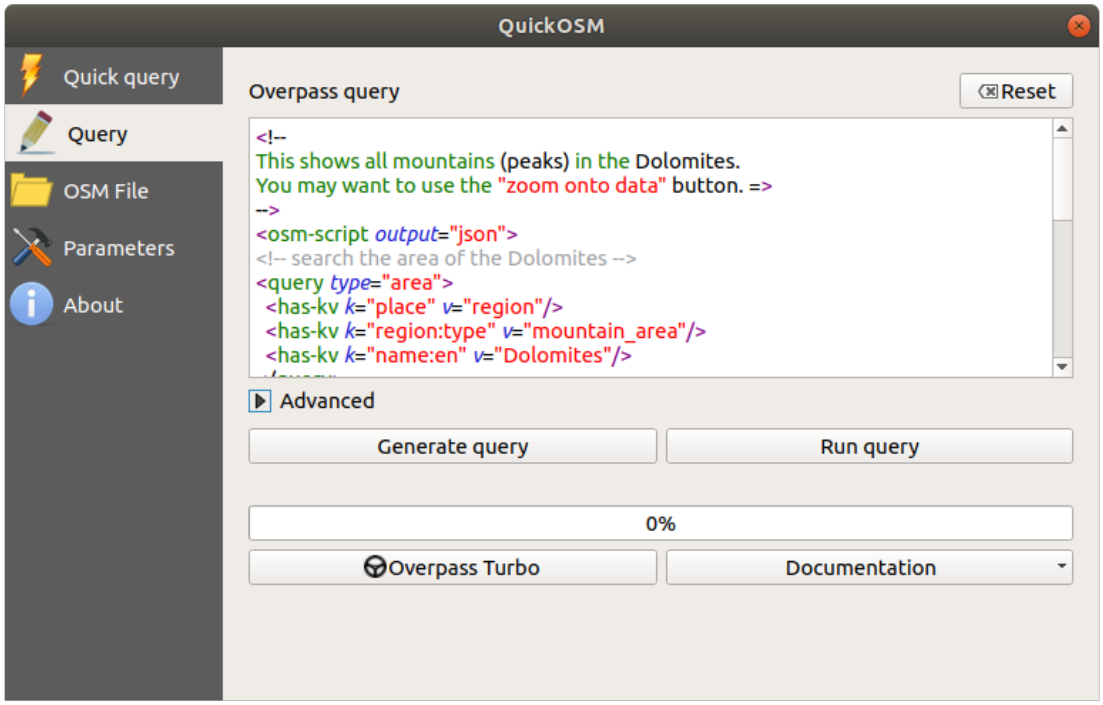
1. Начните новый проект.
2. Откройте плагин QuickOSM и кликните на вкладку *Query*.

3. Скопируйте и вставьте следующий код в холст запроса:

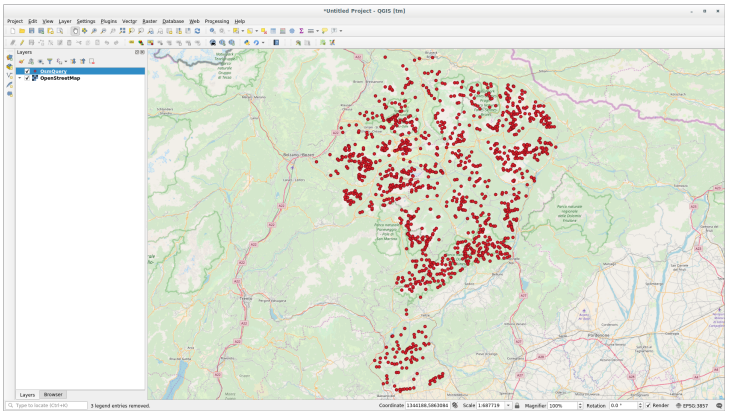
```
  <!--
This shows all mountains (peaks) in the Dolomites.
You may want to use the "zoom onto data" button. =>
-->
<osm-script output="json">
<!-- search the area of the Dolomites -->
<query type="area">
  <has-kv k="place" v="region"/>
  <has-kv k="region:type" v="mountain_area"/>
  <has-kv k="name:en" v="Dolomites"/>
</query>
<print mode="body" order="quadtile"/>
<!-- get all peaks in the area -->
<query type="node">
  <area-query/>
  <has-kv k="natural" v="peak"/>
</query>
<print mode="body" order="quadtile"/>
<!-- additionally, show the outline of the area -->
<query type="relation">
  <has-kv k="place" v="region"/>
  <has-kv k="region:type" v="mountain_area"/>
  <has-kv k="name:en" v="Dolomites"/>
</query>
<print mode="body" order="quadtile"/>
<recurse type="down"/>
<print mode="skeleton" order="quadtile"/>
</osm-script>
```

Примечание: Этот запрос записан на языке `xml`. Если вы больше привыкли работать с `Overpass QL`, вы можете написать запрос на этом языке.

4. И кликните на кнопку *Run Query*:



Слой по горным вершинам будет загружен и показан в QGIS:



Вы можете писать сложные запросы с использованием Языка запросов `Overpass Query language `_. Посмотрите какие-нибудь примеры и попробуйте изучить язык запросов.

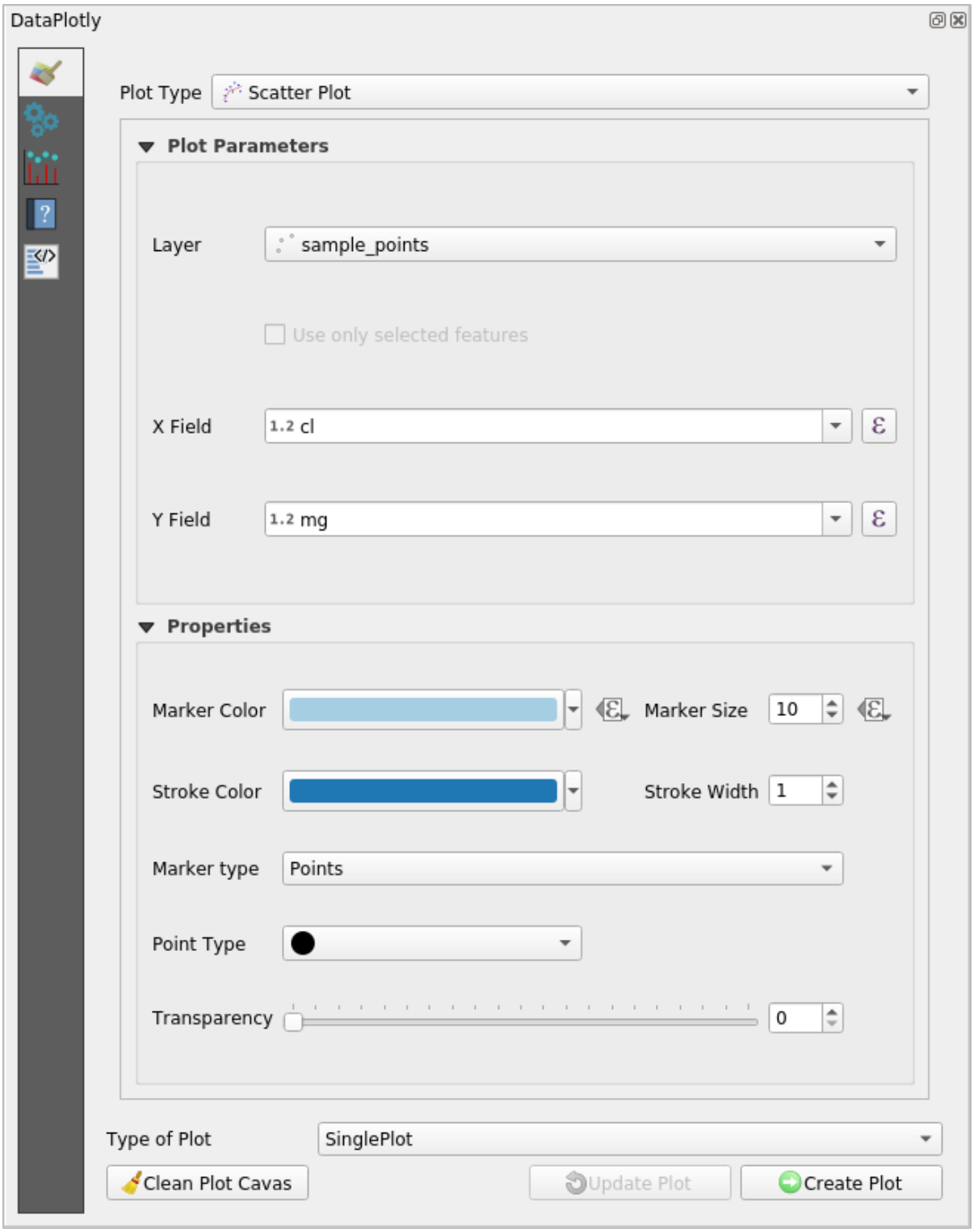
9.2.4 Идем дальше: Плагин DataPlotly

The **DataPlotly** plugin allows you to create D3 plots of vector attributes data thanks to the plotly library.

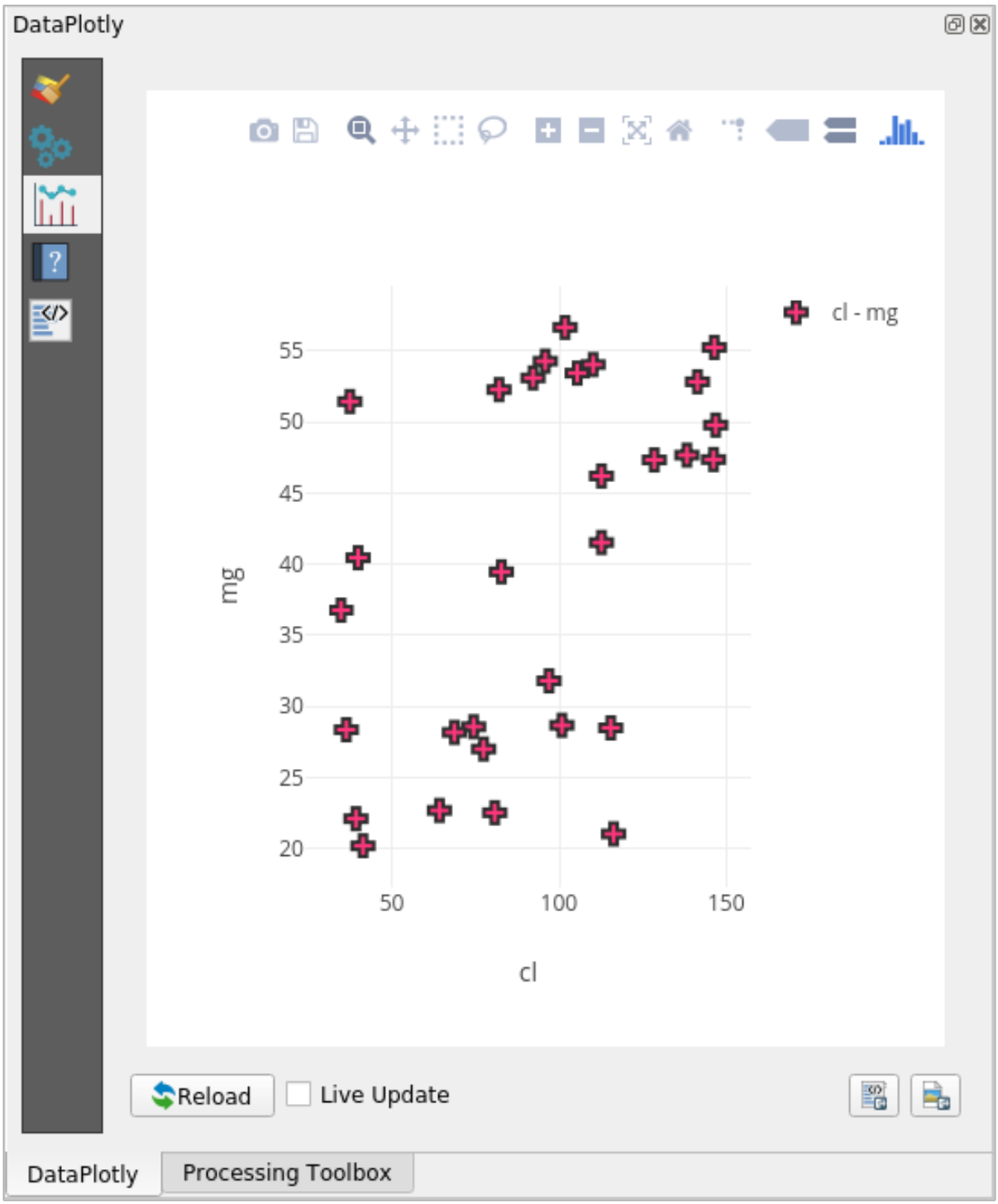
1. Начните новый проект.
2. Загрузите слой *sample_points* из папки `exercise_data/plugins`.
3. Установите плагин, следуя инструкциям, описанным в *Идем дальше: Установка новых плагинов* поиск *Data Plotly*.
4. Откройте плагин, кликнув на новый значок на панели инструментов или в меню *Plugins* ► *Data Plotly*.

В следующем примере мы создаем простой `Scatter Plot` двух полей слоя *sample_points*. В панели DataPlotly:

1. Выберите *sample_points* в фильтре слоя, *cl* для *X Field* и *mg* для *Y Field*:



2. Если хотите, вы можете изменить цвета, тип маркера, прозрачность и многие другие настройки: попробуйте изменить некоторые параметры, чтобы создать график ниже.



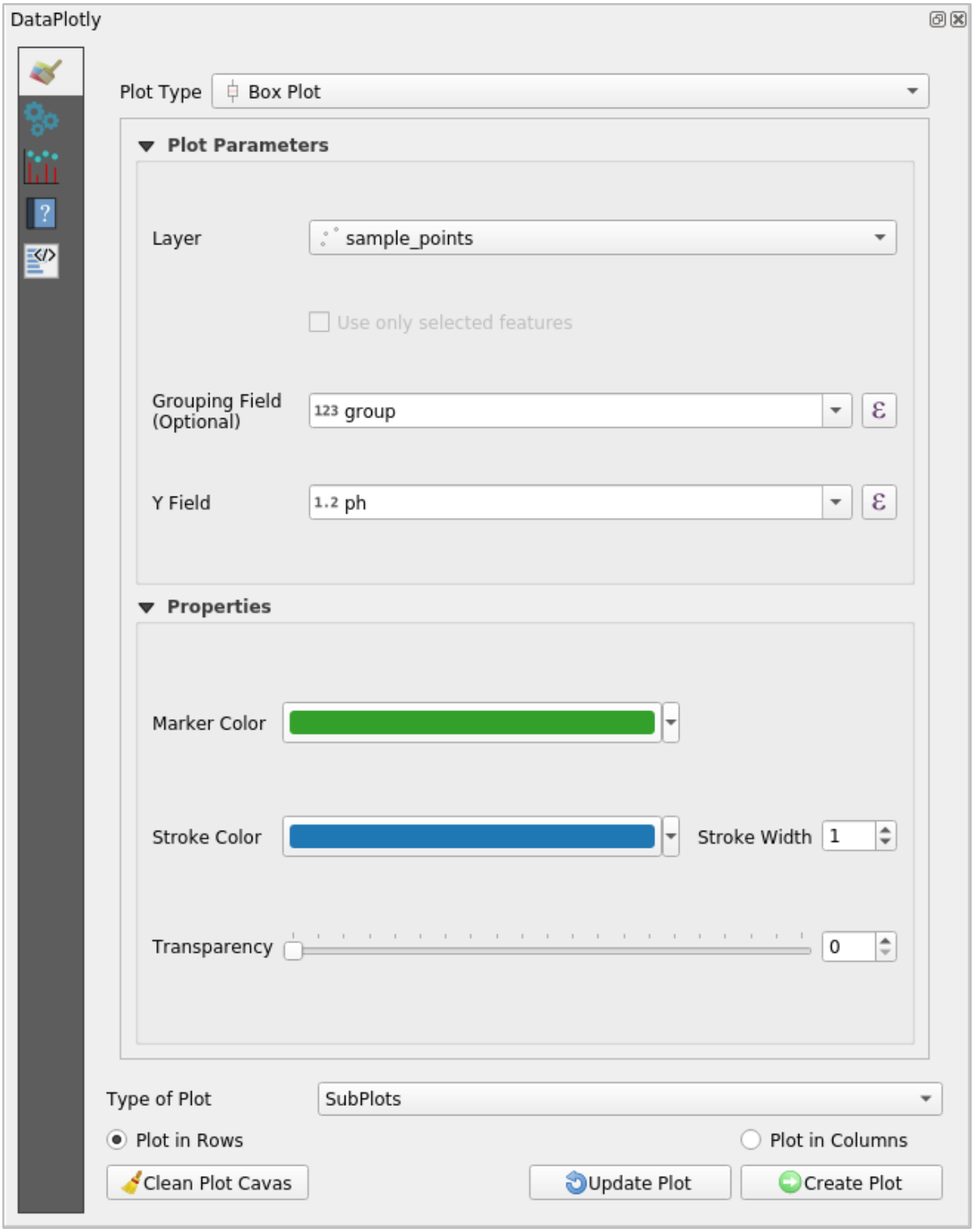
3. После того, как вы настроите все параметры, кликните на кнопку *Create Plot* для создания графика.

График является интерактивным: то есть вы можете использовать все верхние кнопки для изменения размера, перемещения или увеличения/уменьшения масштаба изображения. Кроме того, каждый элемент графика является интерактивным: при нажатии или выборе одной или нескольких точек на графике соответствующие точки будут выбраны на холсте графика.

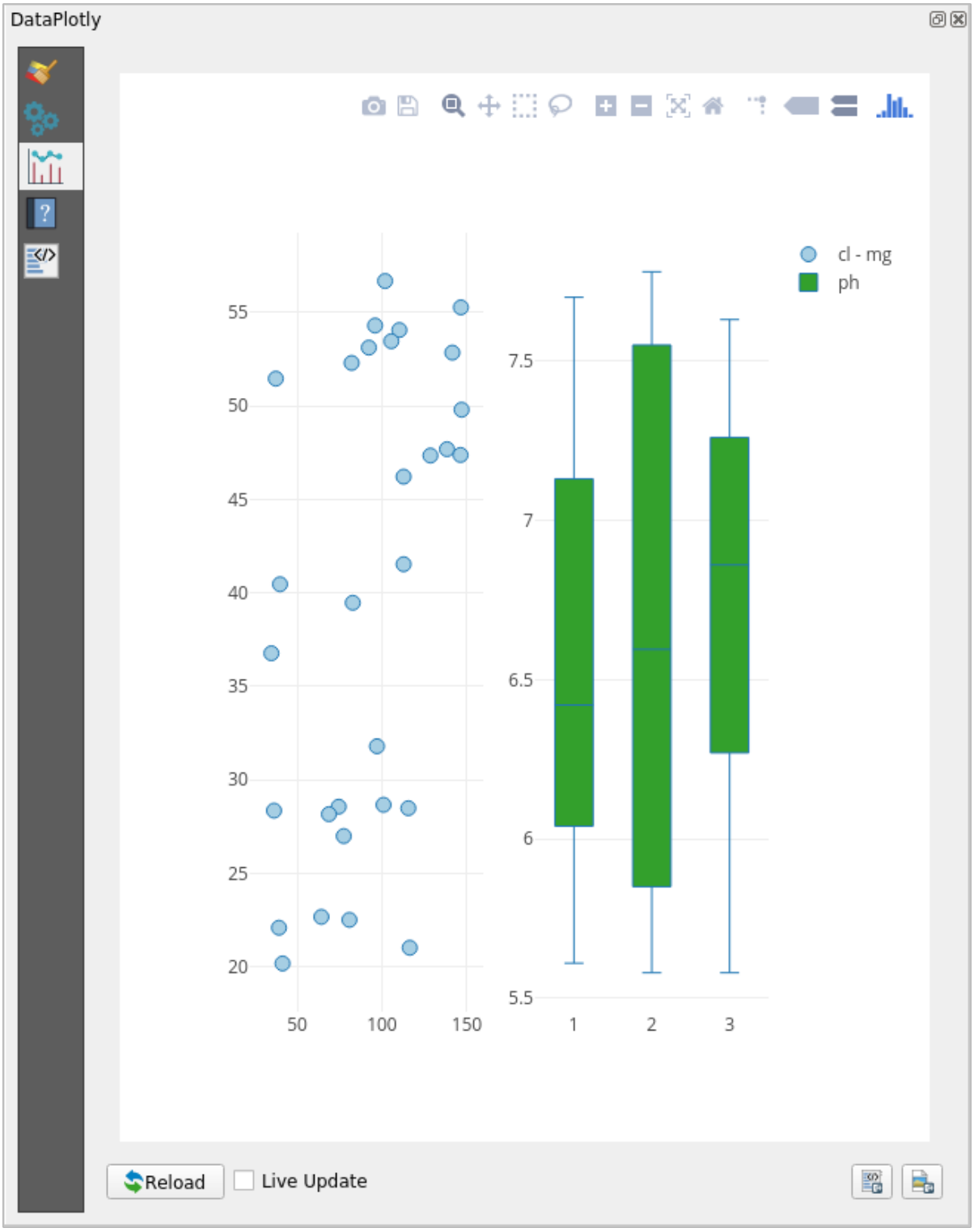
Вы можете сохранить график как статичное изображение `png` или как файл `html`, нажав на кнопку  или на кнопку  в правом нижнем углу графика.

Есть еще кое-что. Иногда лучше иметь два (или более) графика, которые показывают разные типы графиков с разными переменными на одной и той же странице. Давай сделаем это!

1. Надо вернуться на вкладку основных настроек графика, нажав на кнопку  в верхнем левом углу панели плагина.
2. Измените *Plot Type* на *Box Plot*.
3. Выберите *group* как *Grouping Field* и *ph* как *Y Field*.
4. В нижней части панели измените *Type of Plot* из *SinglePlot* на *SubPlots* и пусть опция по умолчанию *Plot in Rows* будет выбрана.



5. Когда закончите, кликните на кнопку *Create Plot* для рисования графика.



Теперь на одной странице графика у вас есть два вида графика т.е. точечный и коробчатый. У вас по-прежнему есть возможность кликнуть каждый элемент графика и выбрать соответствующие объекты на холсте карты.

Совет: У каждого графика есть собственная справочная страница, доступная во вкладке . Попробуйте

изучить все типы графиков и посмотрите все остальные доступные настройки.

9.2.5 В заключении

Есть множество полезных плагинов, доступных для QGIS. Используя встроенные инструменты для установки и управления этими плагинами, вы сможете найти новые плагины и оптимально их использовать.

9.2.6 Что дальше?

Далее мы изучим, как использовать слои, которые размещены на удаленных серверах в режиме реального времени.

ГЛАВА 10

Модуль: Онлайн ресурсы

При рассмотрении источников данных для карты не обязательно ограничиваться данными, которые вы сохранили на вашем компьютере. Существуют онлайн источники данных, из которых вы можете загружать данные.

В этом модуле мы рассмотрим два типа веб-услуг GIS: Сервисы веб-карт - Web Mapping Services (WMS) и веб-сервисы объектов - Web Feature Services (WFS).

10.1 Занятие: Сервисы веб-карт

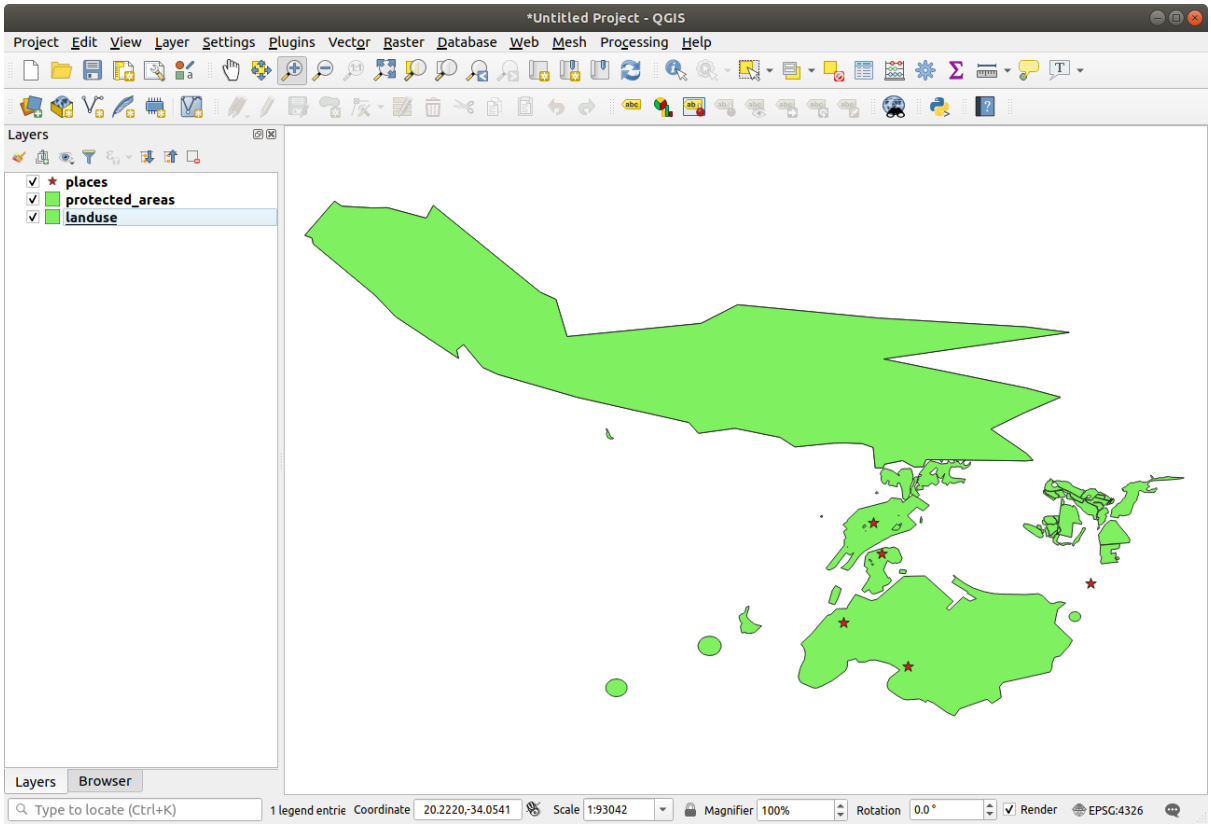
Сервисы веб-карт (WMS) - это сервис, размещенный на удаленном сервере. Как и в случае с веб-сайтом, вы можете получить к нему доступ, если у вас есть соединение с сервером. Используя QGIS, вы можете загрузить WMS прямо на свою существующую карту.

Как вы помните из урока по плагинам, можно, например, загрузить новое растровое изображение из Google. Тем не менее, это разовая операция: после того, как вы загрузили изображение, оно не меняется. WMS отличается тем, что это интерактивный сервис, который автоматически обновляет свой вид при панорамировании или масштабировании карты.

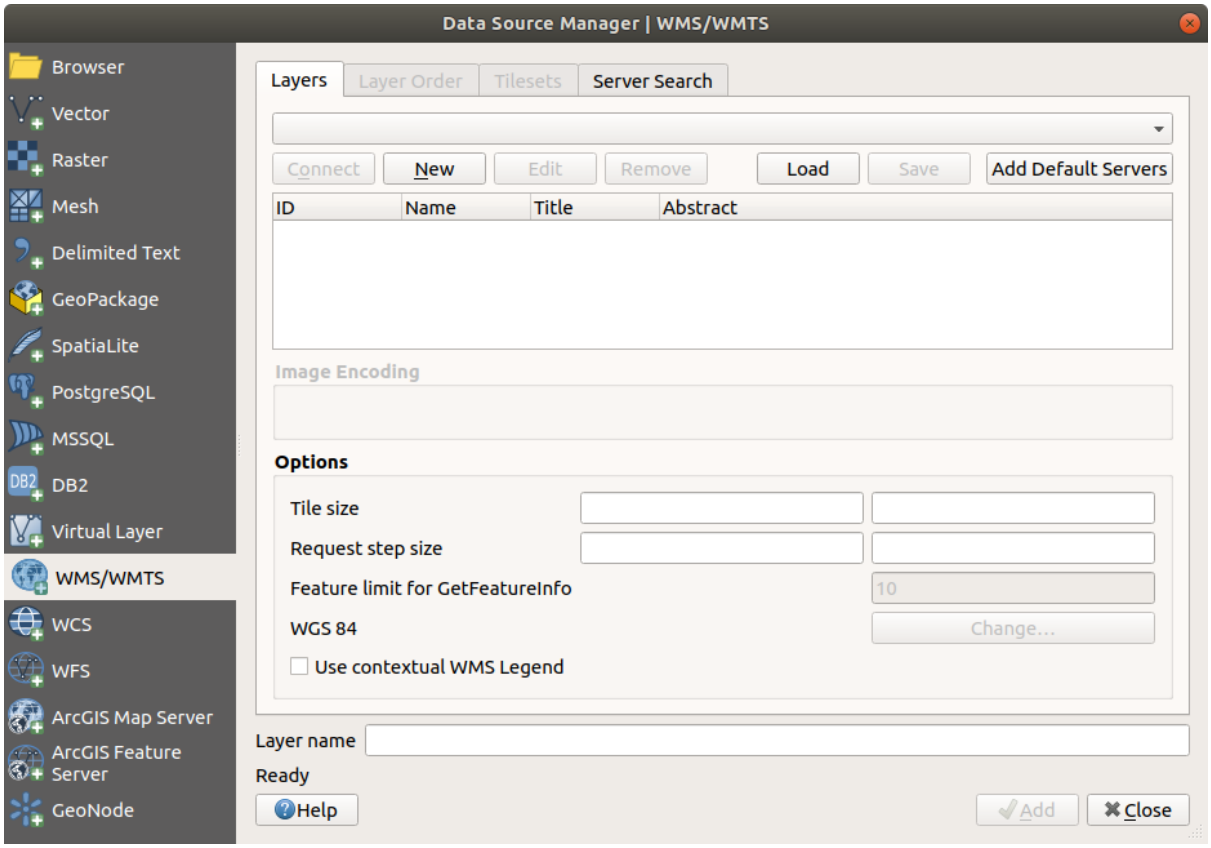
Цель этого урока: Использовать WMS и понять ее ограничения.

10.1.1  Идем дальше: Загружаем слои WMS

Для выполнения данного упражнения вы можете использовать либо базовую карту, которую вы создали в начале курса, или же просто создать новую карту и загрузить в нее некоторые существующие слои. В приведенном примере мы использовали новую карту и загрузили исходные слои *places*, *landuse* и *protected_areas* и скорректировали условные обозначения:



1. Загрузите эти слои на новую карту или используйте свою исходную карту с только этими слоями.
2. Перед тем, как добавить слой WMS, отключите проекцию «на лету» (вкладка *Project* ► *Properties…* ► *CRS*, отметить *No CRS (or unknown/non-Earth projection)*. Это может привести к тому, что слои больше не будут правильно перекрываться, но не волнуйтесь: мы это исправим позже.
3. Для того, чтобы добавить слои WMS, кликните на кнопку для того, чтобы открыть диалоговое окно *Data Source Manager* и включить вкладку *WMS/WMTS*.

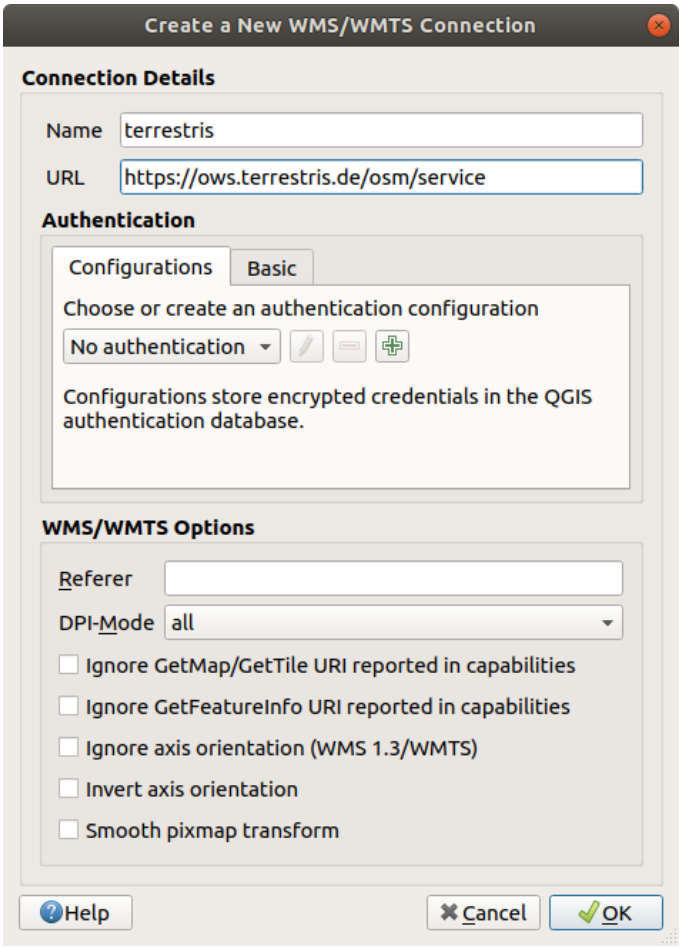


Вспомните, как вы подключались к базе данных SpatiaLite или GeoPackage в начале курса. Слои *landuse*, *buildings* и *roads* хранятся в базе данных. Для того, чтобы использовать эти слои, вам надо было сперва подключиться к базе данных. WMS используется аналогичным образом, за исключением того, что слои находятся на удаленном сервере.

4. Для того, чтобы создать новое подключение к WMS, надо кликнуть на кнопку *New*.

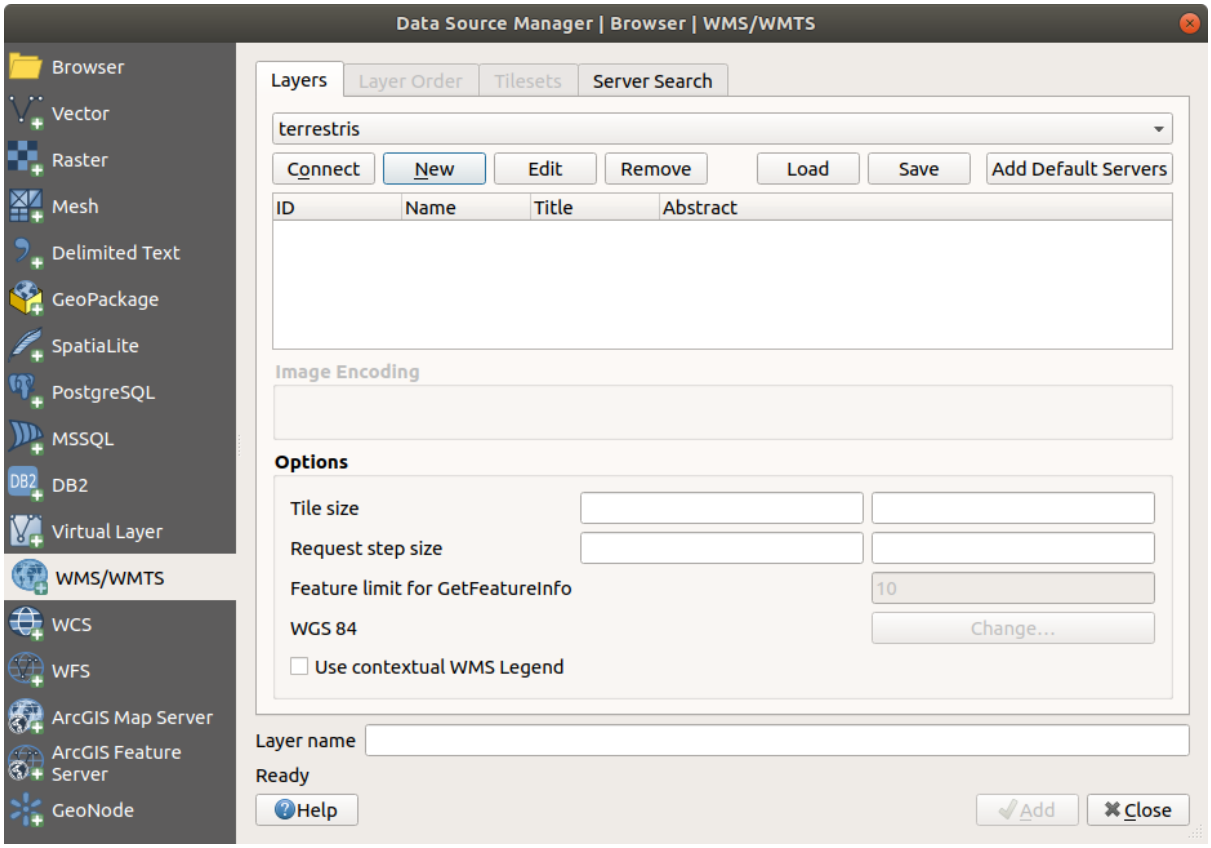
   Для продолжения работы вам нужен адрес WMS. Существует несколько бесплатных серверов WMS. Один из них `terrestris `_ , который использует набор данных `OpenStreetMap `_ .

5. Для того, чтобы воспользоваться этим WMS, настройте ее в вашем текущем диалоговом окне следующим образом:

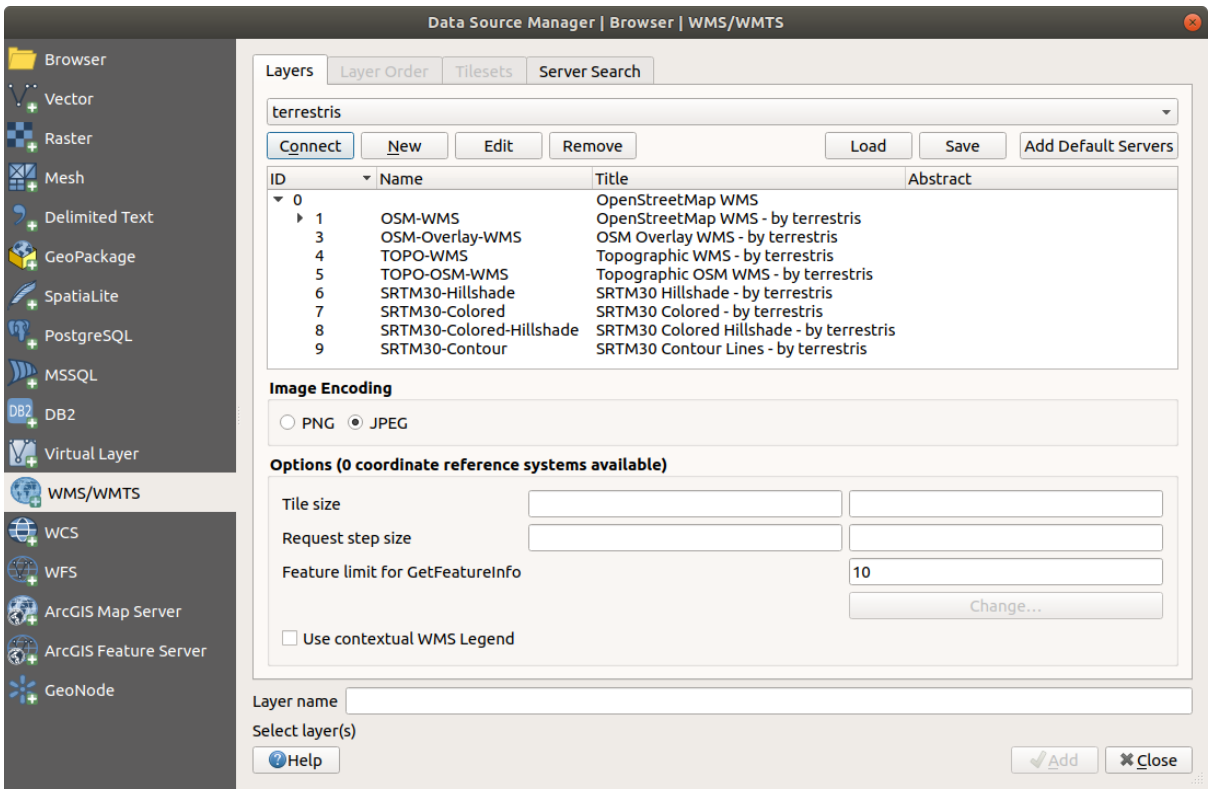


- Значение поля *Name* должно быть `terrestris`.
- Значение поля *URL* должно быть `https://ows.terrestris.de/osm/service`.

6. Кликните на кнопку *OK*. Вы должны увидеть в списке новый WMS-сервер:

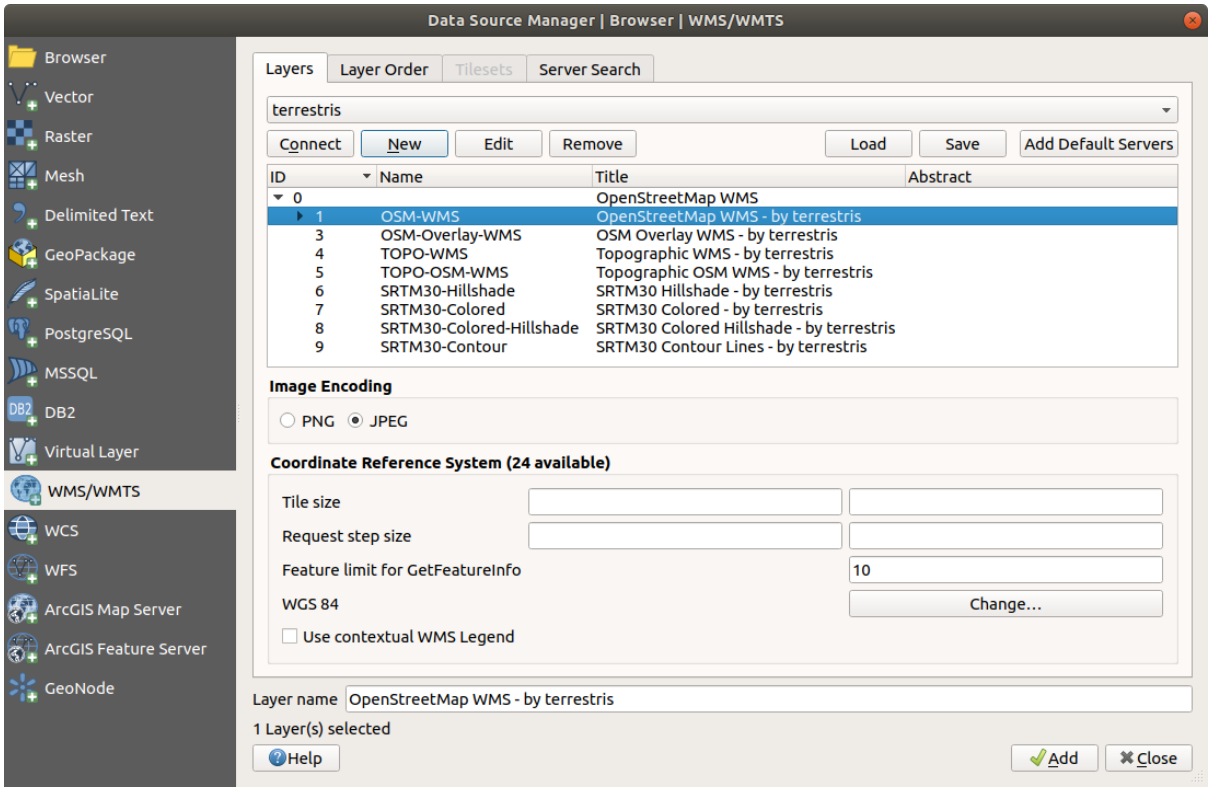


7. Кликните на *Connect*. В списке ниже вы должны увидеть эти новые загруженные записи:



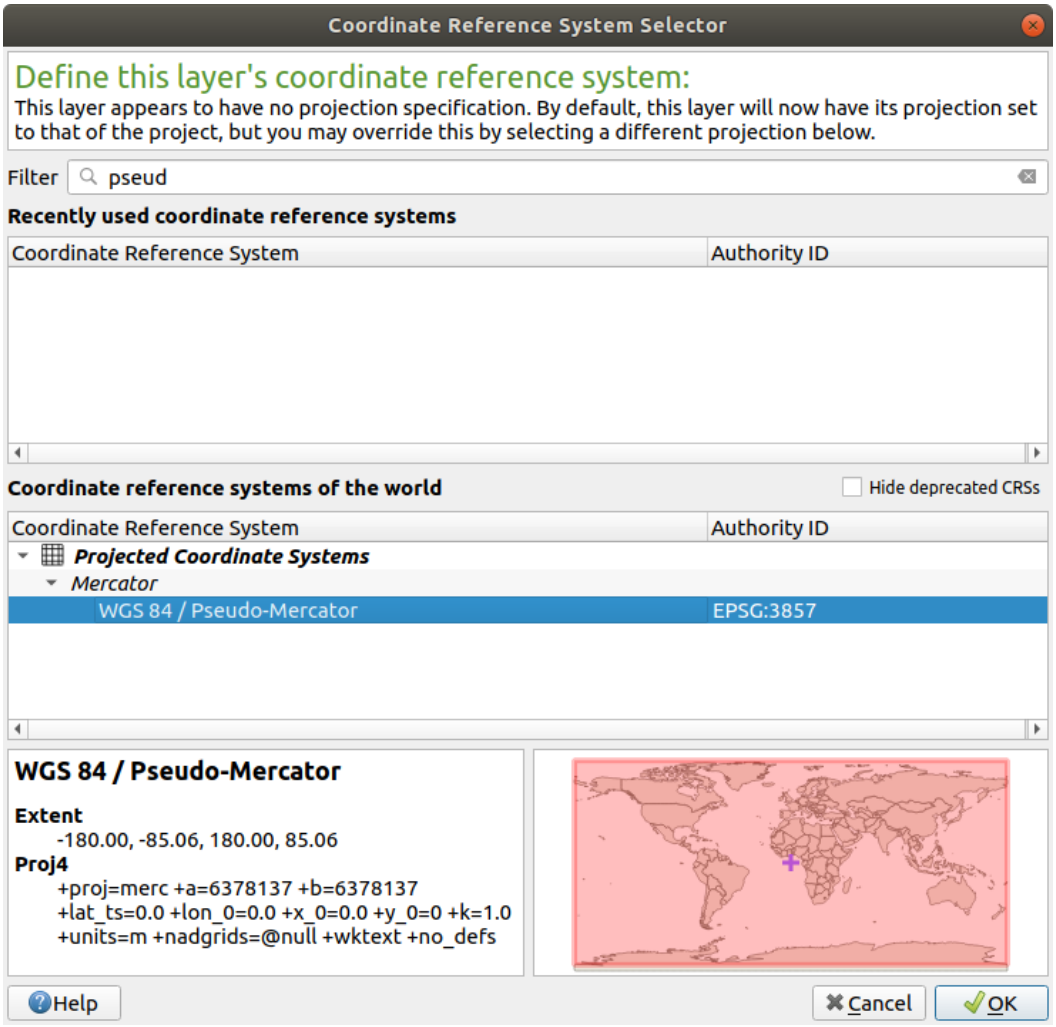
Это все слои, размещенные на этом сервере WMS.

8. Кликните один раз на слой *OSM-WMS*. Появится его *Coordinate Reference System*:



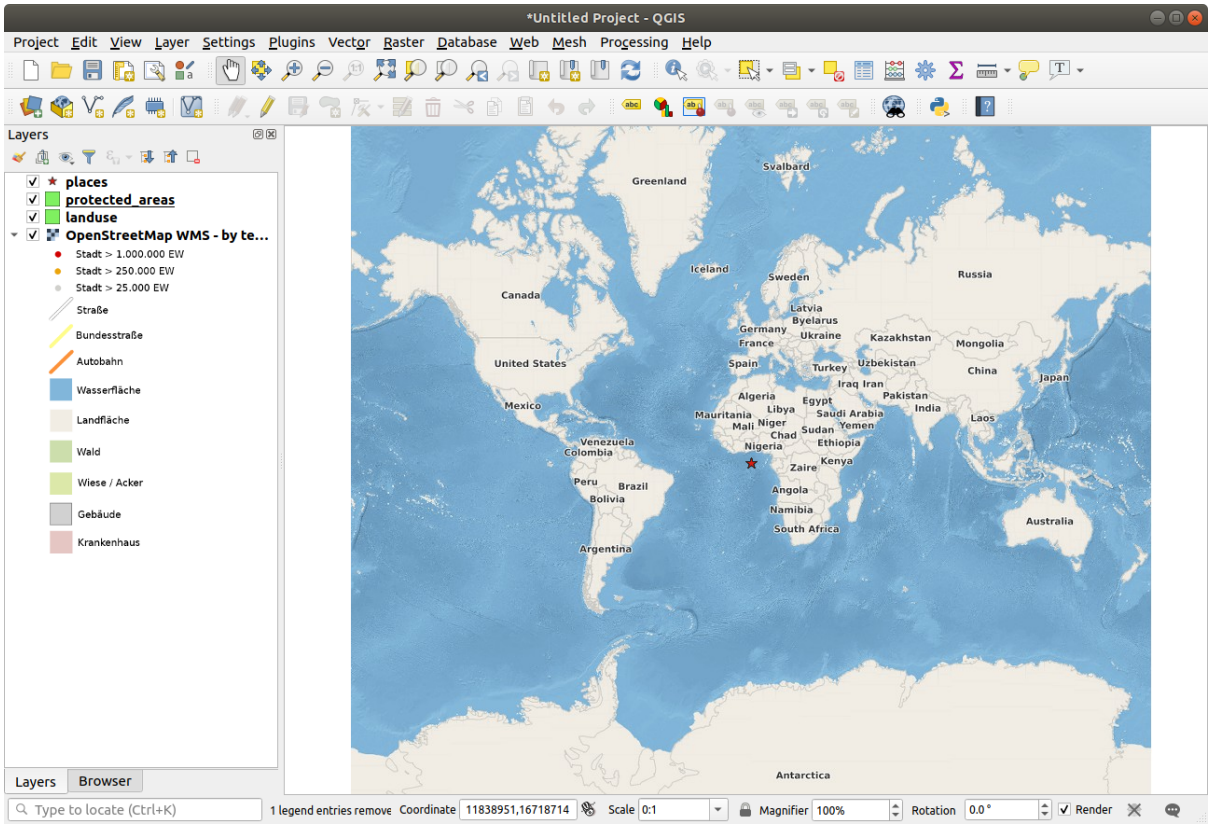
Так как мы не используем `WGS 84` для нашей карты, давайте посмотрим все CRS, из которых нам нужно выбрать.

1. Кликните на кнопку *Change...*. Вы увидите стандартное диалоговое окно *Coordinate Reference System Selector*.
2. Нам нужна *проецированная* CRS, так что давайте выберем *WGS 84 / Pseudo Mercator*.
    1. Введите значение `pseudo` в поле *Filter*:
    2. Выберите *WGS 84 / Pseudo Mercator* (с epsg: 3857) из списка.



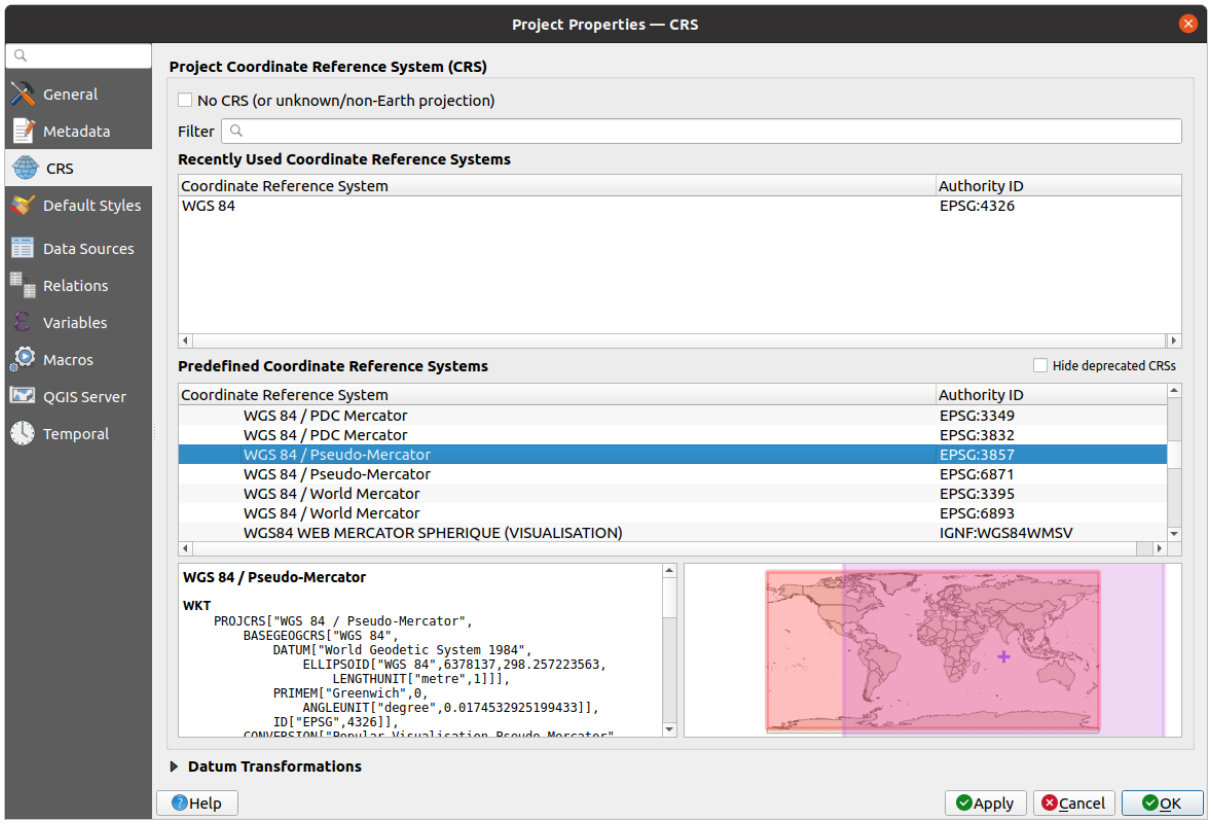
3. Кликните на кнопку *OK*. Система координат, связанная с записью, изменилась.

9. Кликните на *Add* и появится новый слой на вашей карте как *OpenStreetMap WMS - by terrestris*.
10. Закройте диалоговое окно *Data Source Manager*, если оно не закрылась автоматически.
11. На панели *Layers* кликните и перетащите в конец списка.
12. Уменьшите масштаб для того, чтобы получить общий вид слоев. Вы заметите, что ваши слои расположены неправильно (рядом с западной Африкой). Это произошло потому, что проецирование «на лету» отключено.



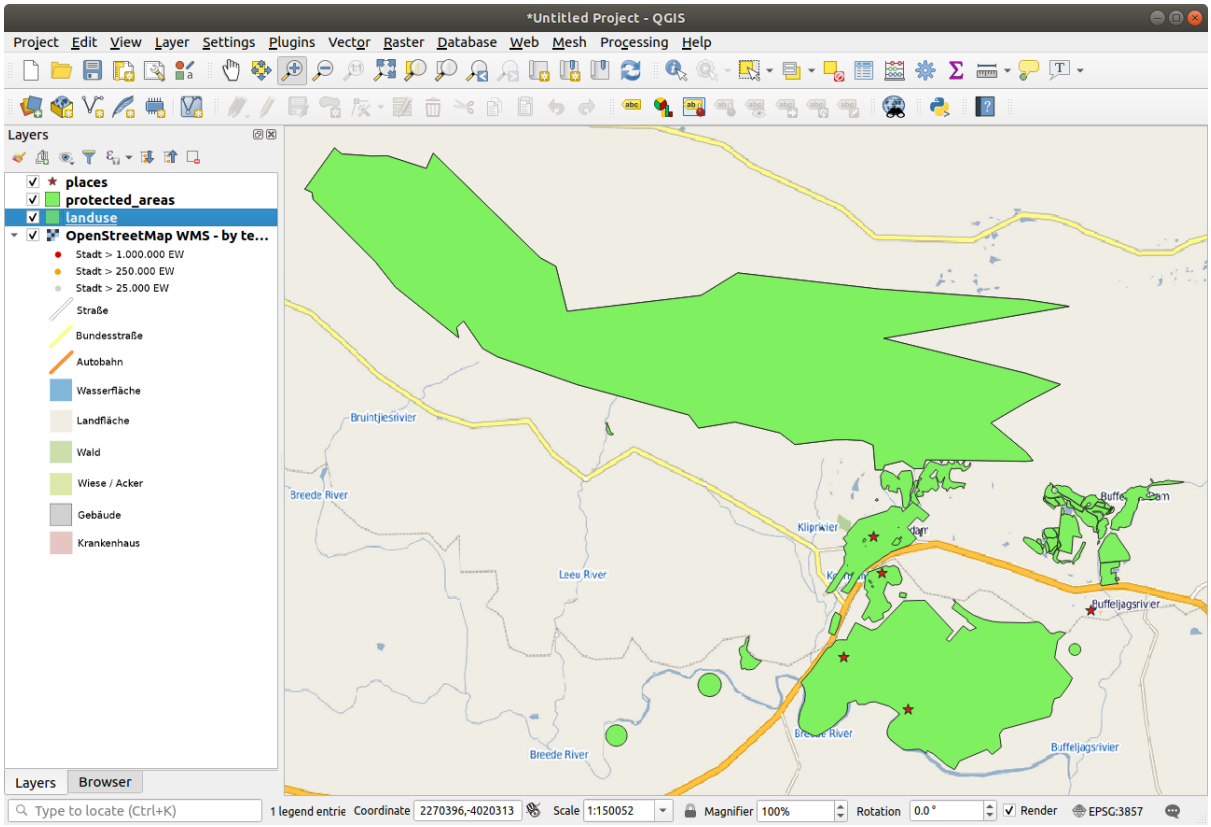
13. Давайте снова включим пере-проецирование, но используя ту же проекцию, что и в слое *OpenStreetMap WMS* т.е. *WGS 84 / Pseudo Mercator*.

    1. Откройте вкладку *Project* ► *Properties…* ► *CRS*.
    2. Снимите флажок *No CRS (or unknown/non-Earth projection)*.
    3. Выберите *WGS 84 / Pseudo Mercator* из списка.



4. Кликните на кнопку *OK*.

14. Now right-click on one of your own layers in the *Layers* panel and click *Zoom to layer(s)*. You should see the Swellendam area:



Обратите внимание, как перекрываются улицы слоя WMS и наши улицы. Это хороший знак!

Сущность и ограничения WMS

К настоящему времени вы, возможно, заметили, что этот слой WMS на самом деле имеет много функций. Здесь есть улицы, реки, заповедники и так далее. Даже если он выглядит так, как будто он состоит из векторов, он больше смахивает на растр, но вы не можете изменить его условные обозначения. Почему?

Так работает WMS: это карта, похожая на обычную бумажную карту, которую вы получаете в виде изображения. У вас есть векторные слои, которые QGIS отображает как карту. Но при использовании WMS эти векторные слои находятся на сервере WMS, который отображает их как карту и выдает эту карту вам как изображение. QGIS может показать это изображение, но не может изменять его условные обозначения, потому что все это обрабатывается на сервере.

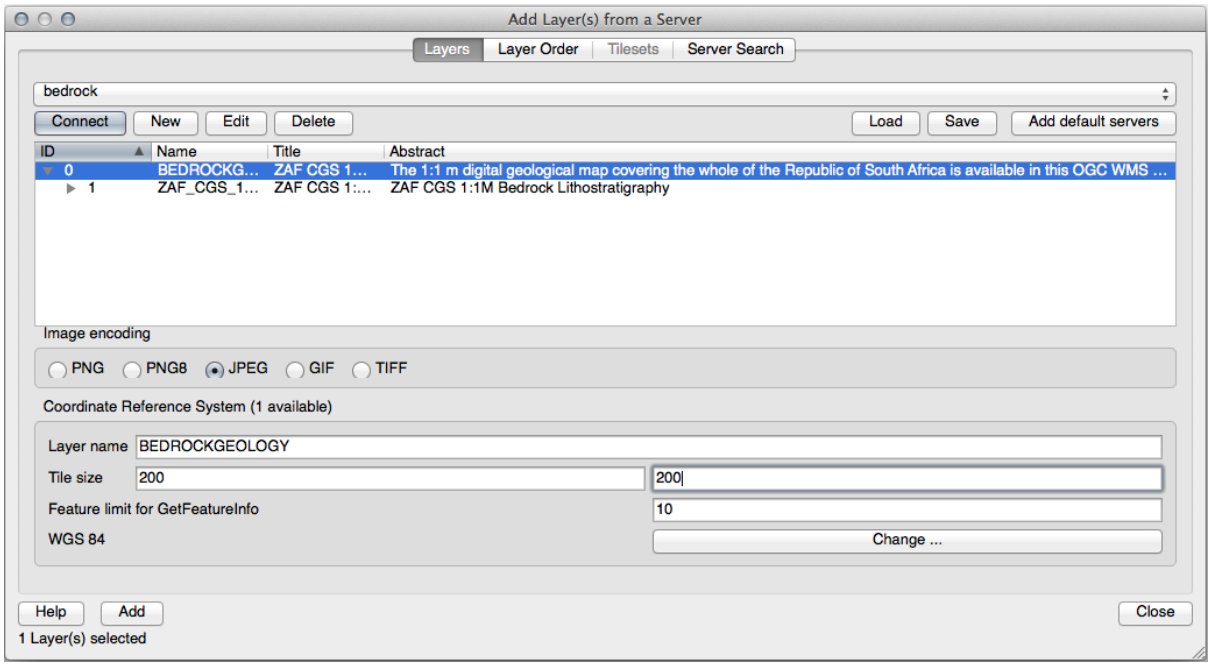
Это дает вам несколько преимуществ, вам не нужно беспокоиться об условных обозначениях. Они уже отработаны и на них можно посмотреть на любой грамотно спроектированной WMS.

С другой стороны, вы не сможете изменить условные обозначения, если они вам не нравятся, и если что-то меняется на сервере WMS, они также поменяются на вашей карте. Поэтому иногда вам захочется использовать вместо этого веб-сервисы объектов (WFS), которые предоставляют векторные слои отдельно, а не как часть карты в стиле WMS.

Этот вопрос будет рассмотрено в следующем уроке. Сначала давайте добавим еще один слой WMS из сервера WMS *terrestris*.

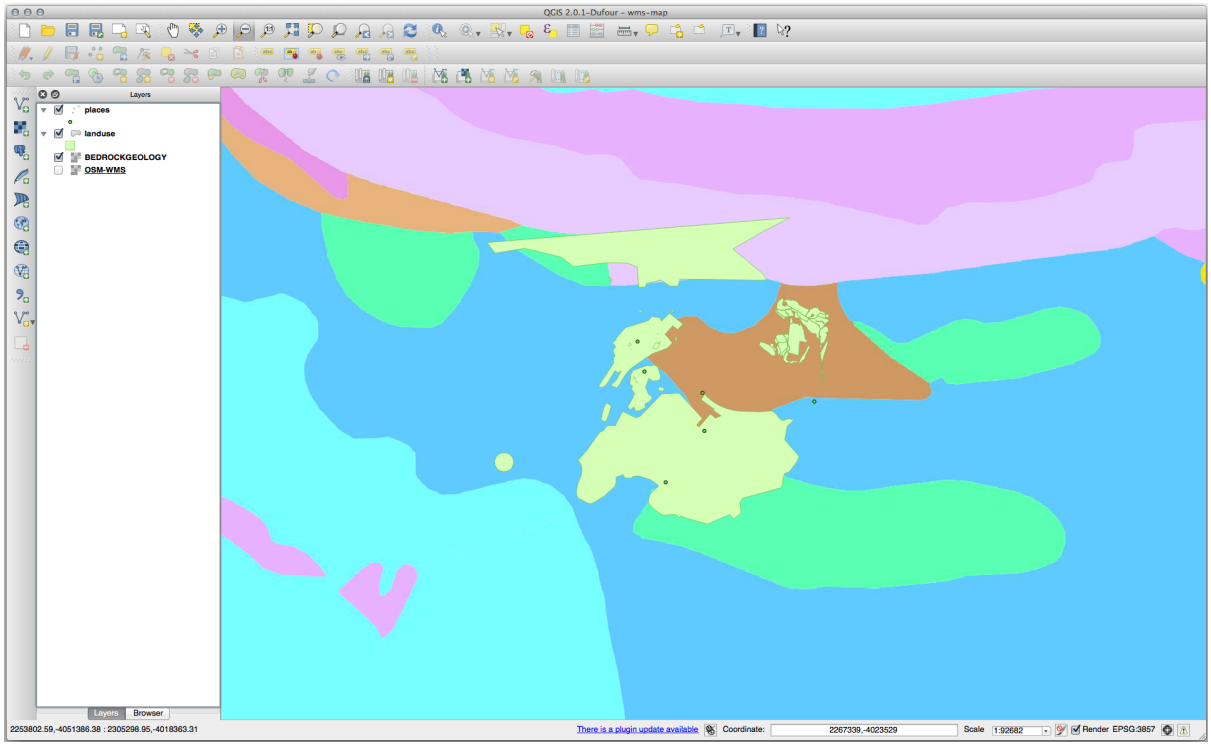
10.1.2  Попробуйте сами

1. Надо скрыть слой *OSM-WSM* в панели *Layers*.
2. Добавьте «ZAF CGS 1M Bedrock Lithostratigraphy» WMS сервер в: URL: `http://196.33.85.22/cgi-bin/ZAF_CGS_Bedrock_Geology/wms`.
3. Загрузите слой *BEDROCKGEOLOGY* на карту (вы также можете использовать кнопку *Layer* ► *Add Layer* ►  *Add WMS/WMTS Layer...* для того, чтобы открыть диалоговое окно the Data Source Manager). Надо проверить, находится ли она той же *WGS 84 / World Mercator* проекции как и остальная часть вашей карты!
4. Вы можете настроить его *Encoding* на *JPEG* и его *Tile size* опцию на `200` на `200`, чтобы она загружалась быстрее:



Answer

Your map should look like this (you may need to re-order the layers):

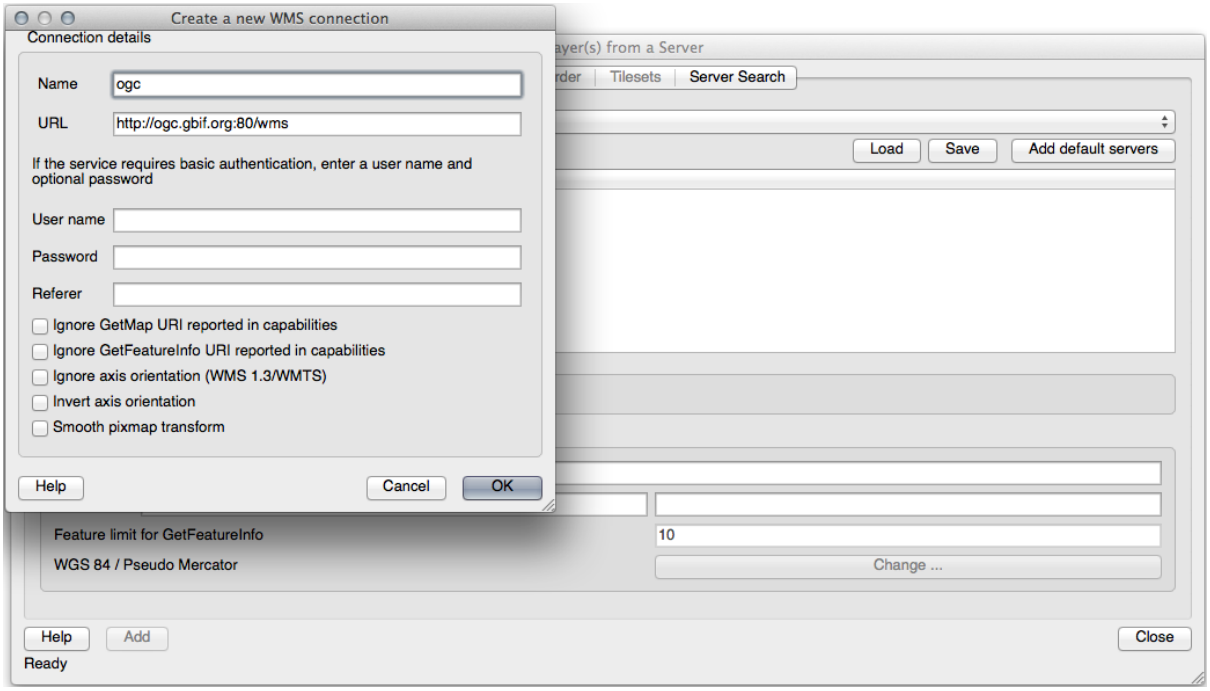


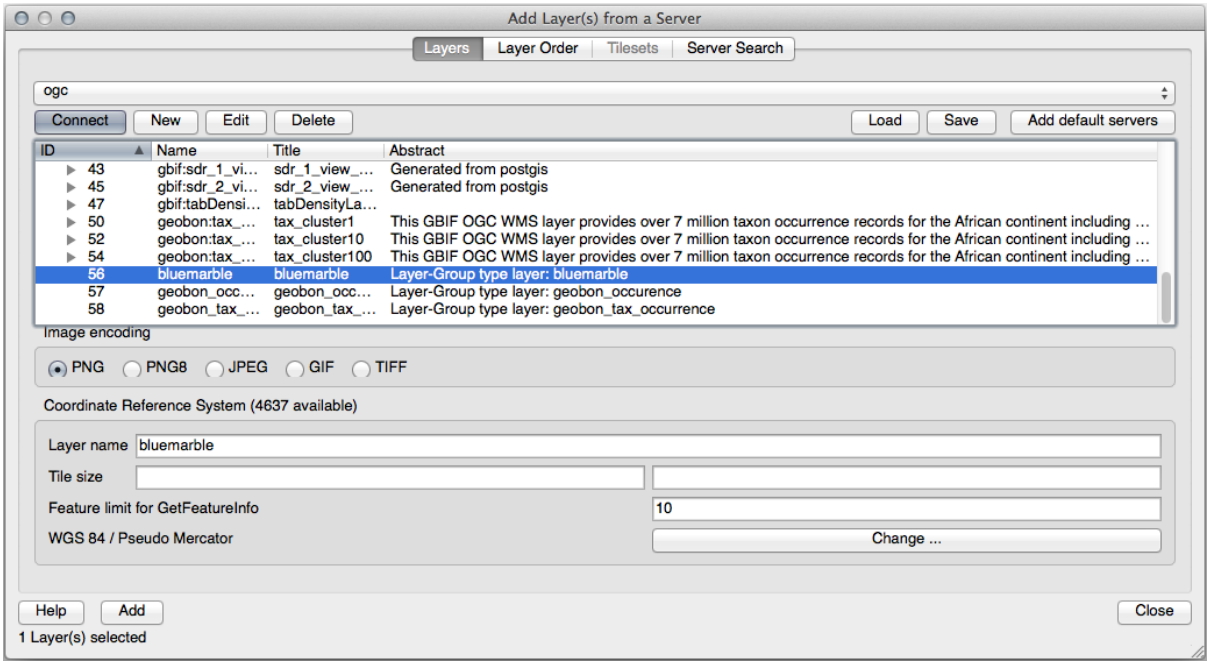
10.1.3  Попробуйте сами

1. Скройте все остальные слои WMS, чтобы они не отображались в фоновом режиме без надобности.
2. Добавьте «OGC» WMS-сервер на: `http://ogc.gbif.org:80/wms`.
3. Добавьте слой *bluemarble*.

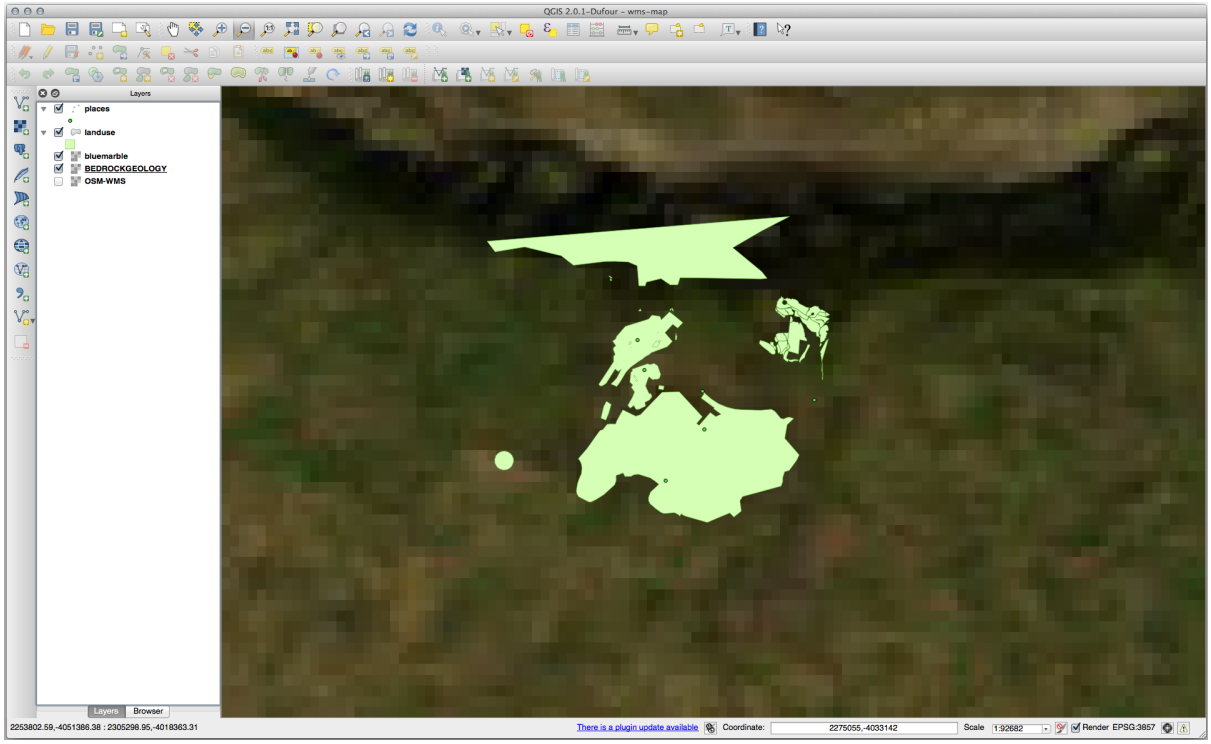
Answer

- Use the same approach as before to add the new server and the appropriate layer as hosted on that server:





- If you zoom into the Swellendam area, you'll notice that this dataset has a low resolution:



Therefore, it's better not to use this data for the current map. The Blue Marble data is more suitable at global or national scales

---

10.1.4 Попробуйте сами

Сложность использования WMS заключается в том, чтобы найти хороший (бесплатный) сервер.

- Найдите новую WMS на `directory.spatineo.com `_ (или еще где-то в интернете) без надобности оплаты каких-либо услуг или без ограничений и он должен охватывать всю территорию исследования Swellendam.

  Помните, что для использования WMS вам нужен лишь адрес сайта (и желательно какое-то описание).

Answer

You may notice that many WMS servers are not always available. Sometimes this is temporary, sometimes it is permanent. An example of a WMS server that worked at the time of writing is the *World Mineral Deposits* WMS at http://apps1.gdr.nrcan.gc.ca/cgi-bin/worldmin_en-ca_ows. It does not require fees or have access constraints, and it is global. Therefore, it does satisfy the requirements. Keep in mind, however, that this is merely an example. There are many other WMS servers to choose from.

10.1.5 В заключении

Используя WMS, вы можете добавлять неактивные карты в качестве фона для ваших существующих картографических данных.

10.1.6 Дополнительные материалы для чтения

- `Spatineo Directory `_
- `OpenStreetMap.org list of WMS servers `_

10.1.7 Что дальше?

После того как вы добавили неактивную карту в качестве фона, вы будете рады узнать, что вы также можете добавлять объекты (например, другие векторные слои, которые вы добавляли ранее). Добавление объектов с удаленных серверов возможно с помощью веб-сервисов объектов (WFS). Это тема следующего урока.

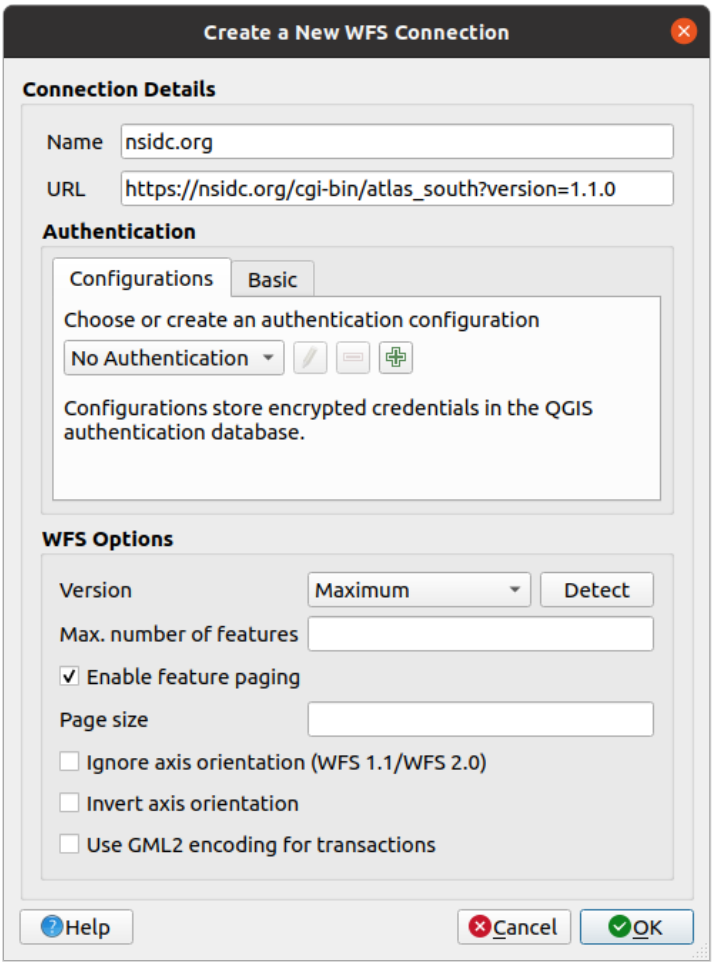
10.2 Занятие: Веб-сервисы объектов

Веб-сервисы объектов (WFS) дают своим пользователям данные ГИС в форматах, которые можно загрузить непосредственно в QGIS. В отличие от WMS, которая предоставляет вам только карту, которую вы не можете редактировать, WFS дает вам доступ к самим объектам.

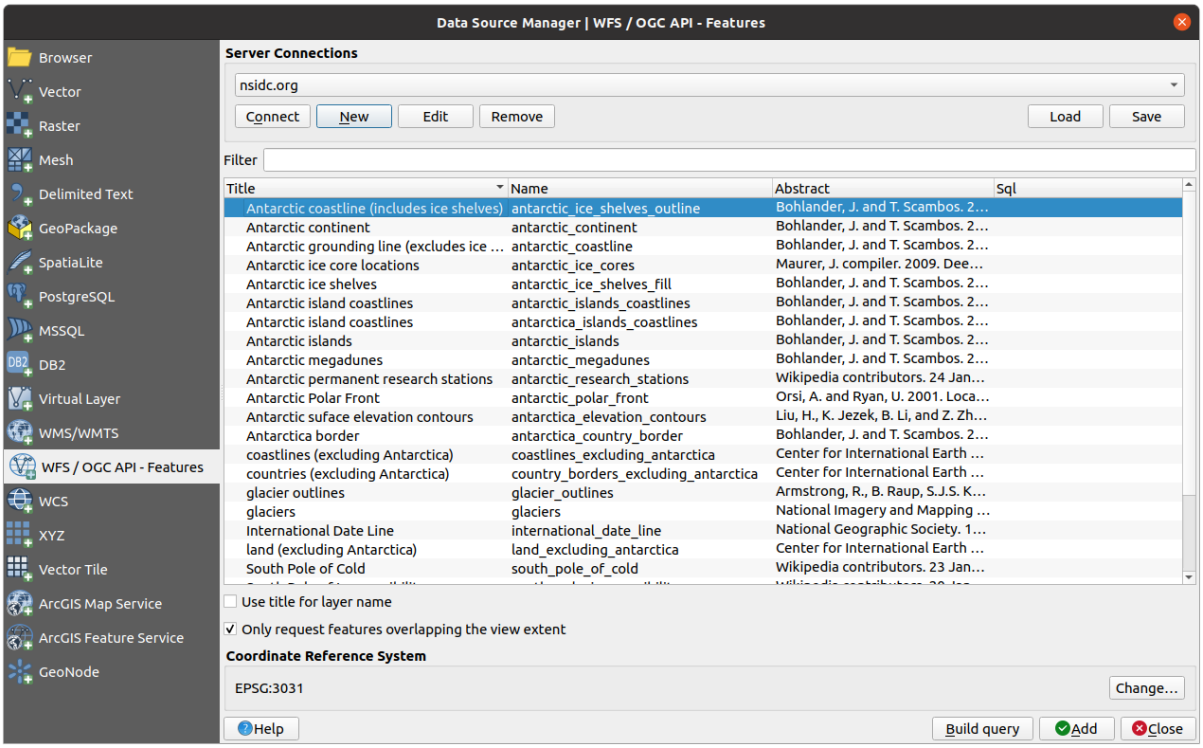
Цель этого урока: Использовать WFS и понять, чем он отличается от WMS.

10.2.1  Идем дальше: Загружаем слои WFS

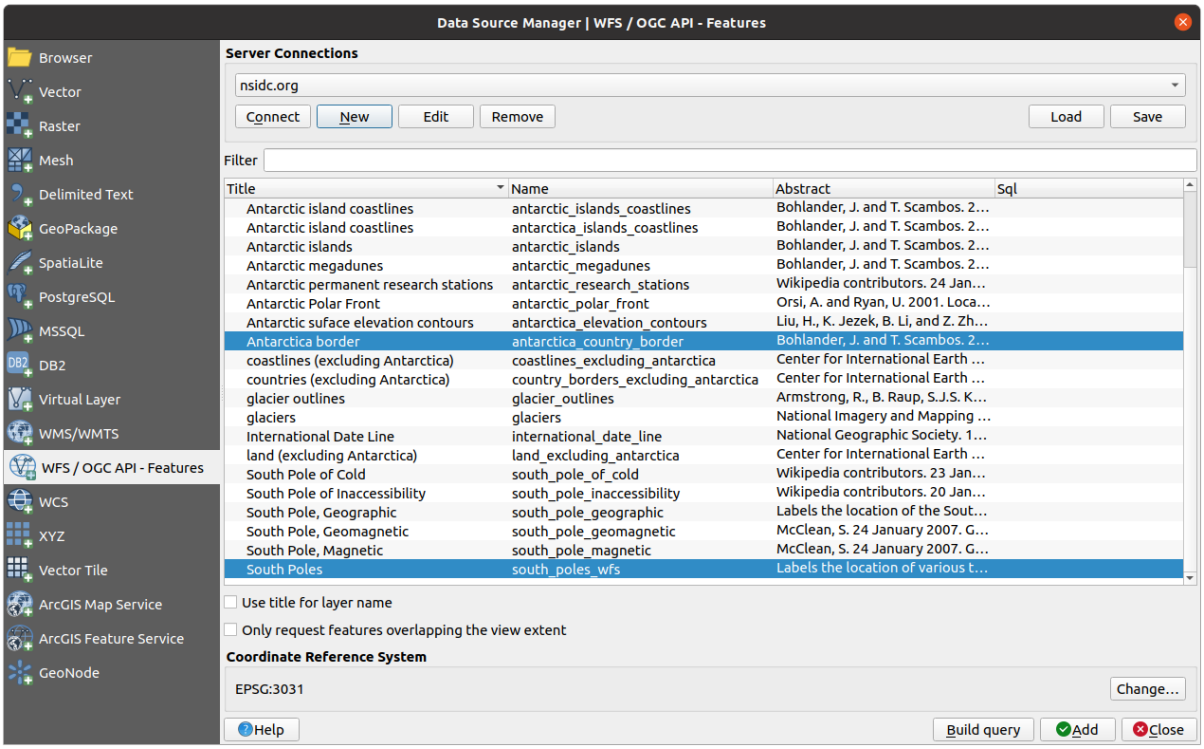
1. Запустите новую карту. Вы делаете это для демонстрационных целей и не будете ее сохранять.
2. Кликните на кнопку  Open Data Source Manager.
3. Включите вкладку  *WFS / OGC API - Features*.
4. Кликните на кнопку *New*.
5. В появившемся диалоговом окне надо ввести *Name* в виде `nsidc.org` и адрес сайта *URL* `https://nsidc.org/cgi-bin/atlas_south?version=1.1.0`.



6. Кликните на кнопку *OK* и появится новое соединение в вашем *Server connections*.
7. Кликните на *Connect*. Появится список доступных слоев:

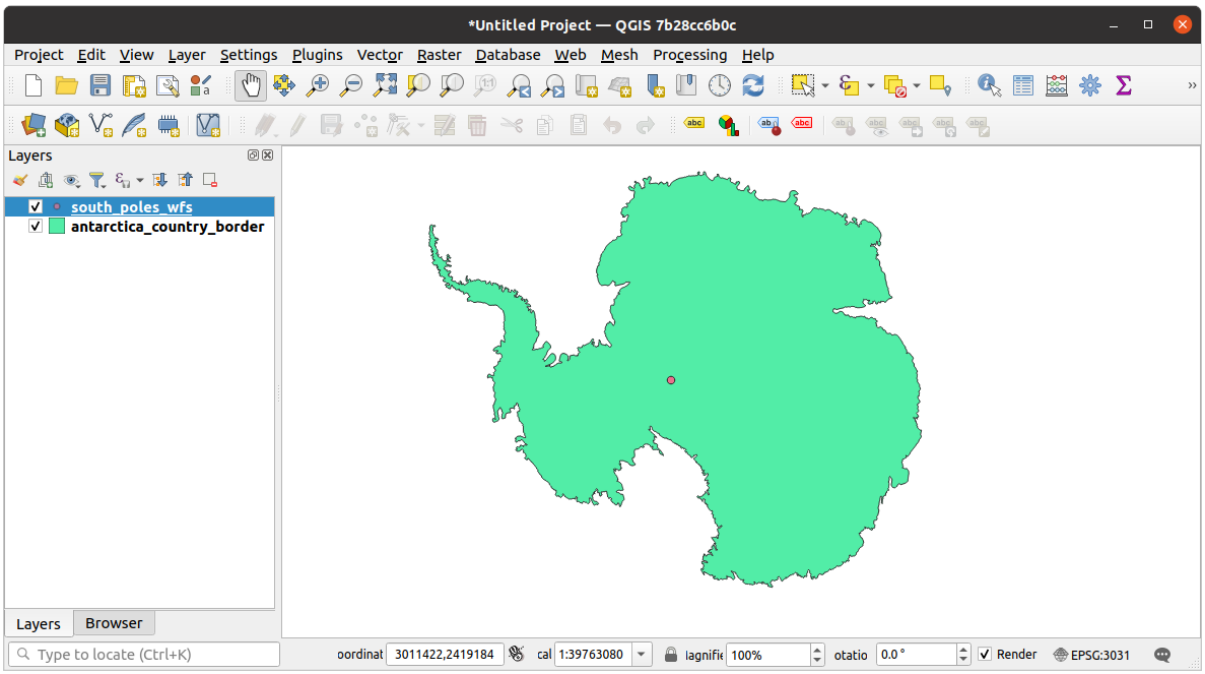


8. Снимите флажок *Only request features overlapping the view extent* внизу списка слоев, так как текущий холст вашей карты может не покрывать интересующую нас местность: Антарктида.
9. Найдите слой *antarctica_country_border*. Вы можете использовать ячейку *Filter* на верхней части.
10. Кликните на слой, чтобы выбрать его:
11. Найдите и выберите также слой *south_poles_wfs*. Вам надо удерживать `Ctrl`.



12. Кликните на *Add*.

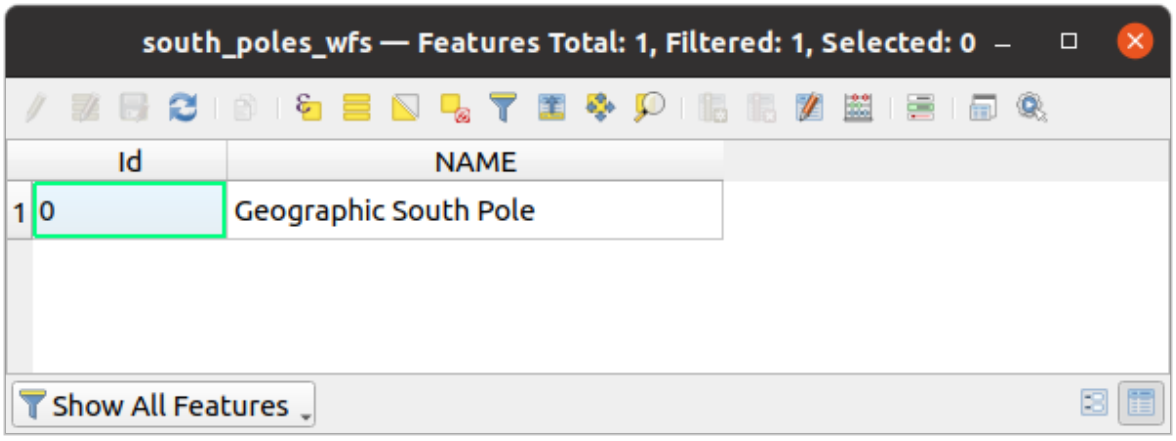
Загрузка слоев может занять некоторое время. После загрузки они появятся на карте, показывая очертания Антарктиды и несколько точек над ней.



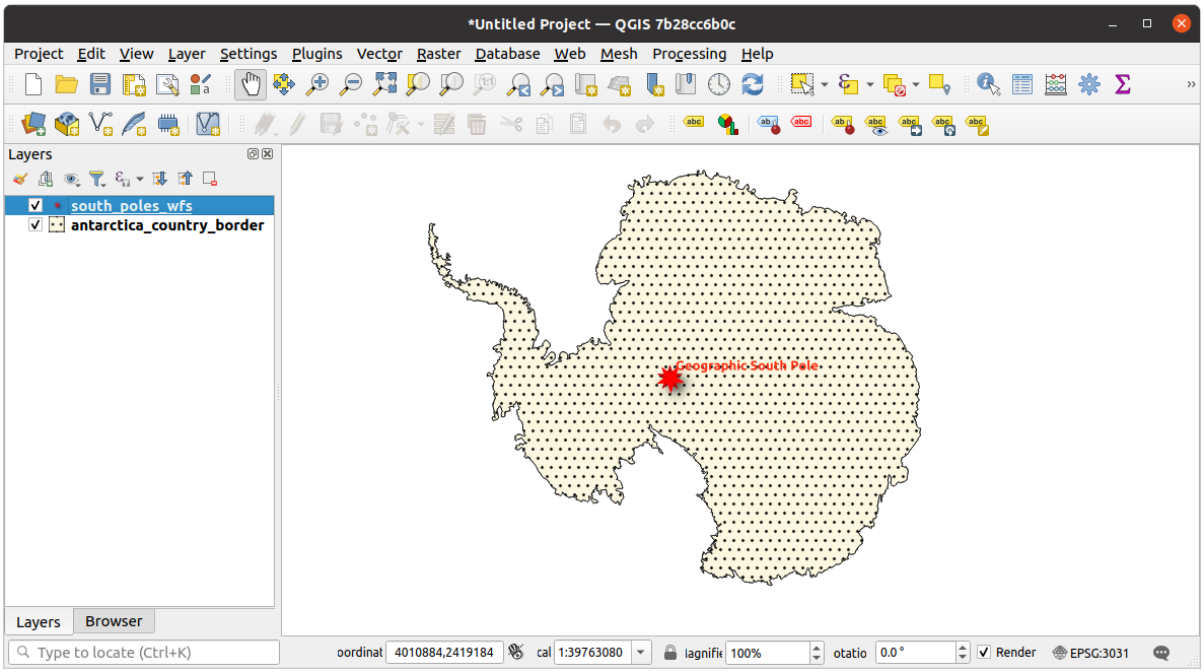
В чем отличие от слоя WMS?

13. Выберите любой из слоев и вы заметите, что включены инструменты выбора объектов и таблицы атрибутов. Это векторные слои.

14. Выберите слой *south_poles_wfs* и откройте его таблицу атрибутов. Вы должны увидеть это:



Так как точки имеют атрибуты, мы можем пометить их, а также изменить их условные обозначения. Вот пример:



Отличия от слоев WMS

Веб-сервис объектов возвращает сам слой, а не только карту, визуализированную из него. Это дает вам прямой доступ к данным, а значит, вы можете изменять их условные обозначения и запускать функции анализа этих данных. Тем не менее, это происходит за счет передачи гораздо большего объема данных. Это становится особенно очевидным, если загружаемые вами слои имеют сложные формы, много атрибутов или много объектов; или если вы просто загружаете много слоев. Из-за этого слои WFS обычно загружаются очень долго.

10.2.2 Идем дальше: Запрашиваем слои WFS

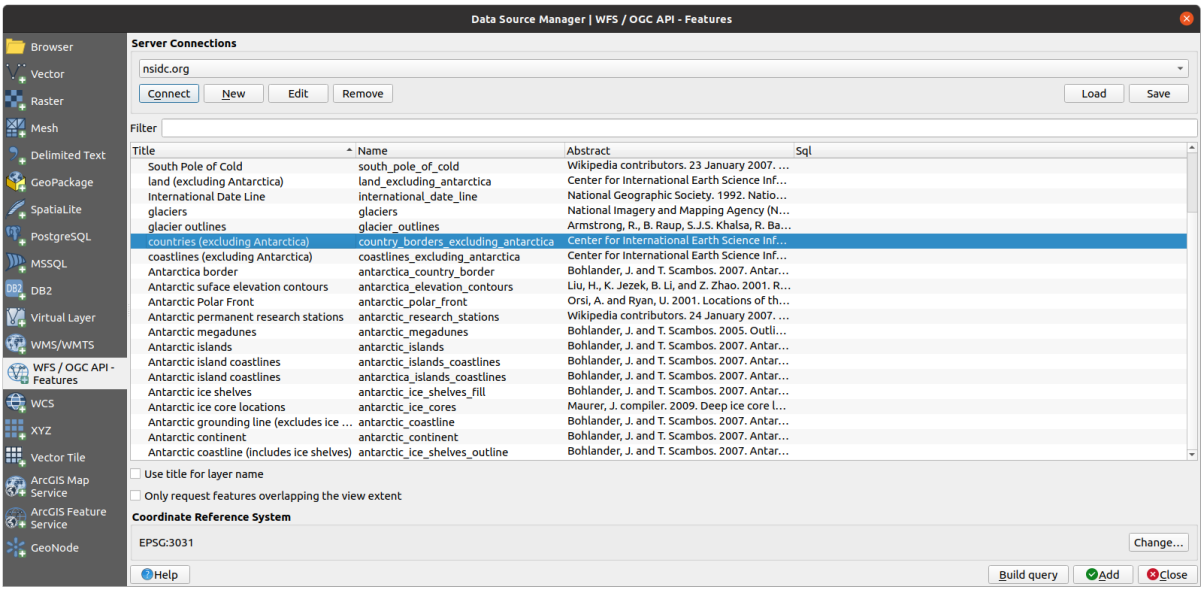
Хотя и можно сделать запрос слоя WFS после его загрузки, но чаще всего более эффективно запросить его перед загрузкой. Таким образом, вы запрашиваете только те объекты, которые вам нужны, а это означает, что вы используете гораздо меньшую пропускную способность.

Например, на сервере WFS, который мы сейчас используем, есть слой, который называется *countries (excluding Antarctica)*. Предположим, что мы хотим знать, где находится Южная Африка по отношению к слою *south_poles_wfs* (и, возможно, также слой *antarctica_country_border*), который уже был загружен.

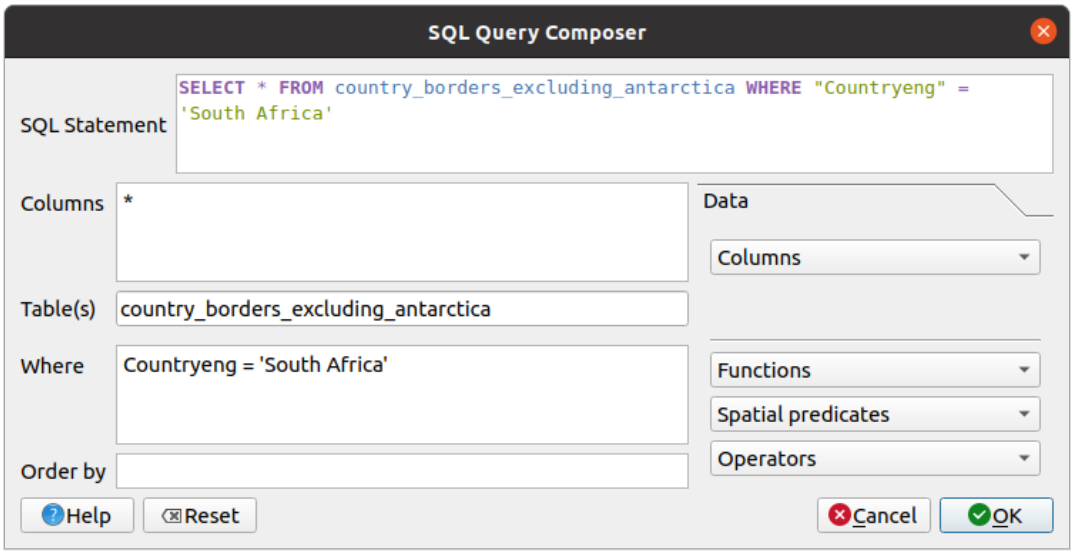
Есть два способа сделать это. Вы можете загрузить весь слой *countries* …, а затем создайте запрос, как обычно, после его загрузки. Но передача данных по всем странам мира и последующее использование данных только по Южной Африке кажется немного расточительным с точки зрения пропускной способности. В зависимости от скорости вашего подключения, загрузка этого набора данных может занять несколько минут.

Альтернативным вариантом является создание запроса в качестве фильтра еще до загрузки слоя с сервера.

1. Включите вкладку *WFS / OGC API Features* в диалоговом окне *Data Source Manager*.
2. Подключитесь к серверу, который мы использовали раньше, и вы должны увидеть список доступных слоев.
3. Найдите и дважды кликните по слою *countries (excluding Antarctica)*. Название слоя такое `country_borders_excluding_antarctica`. Вы также можете выбрать слой и кликнуть на кнопку *Build query* в нижней части диалогового окна:

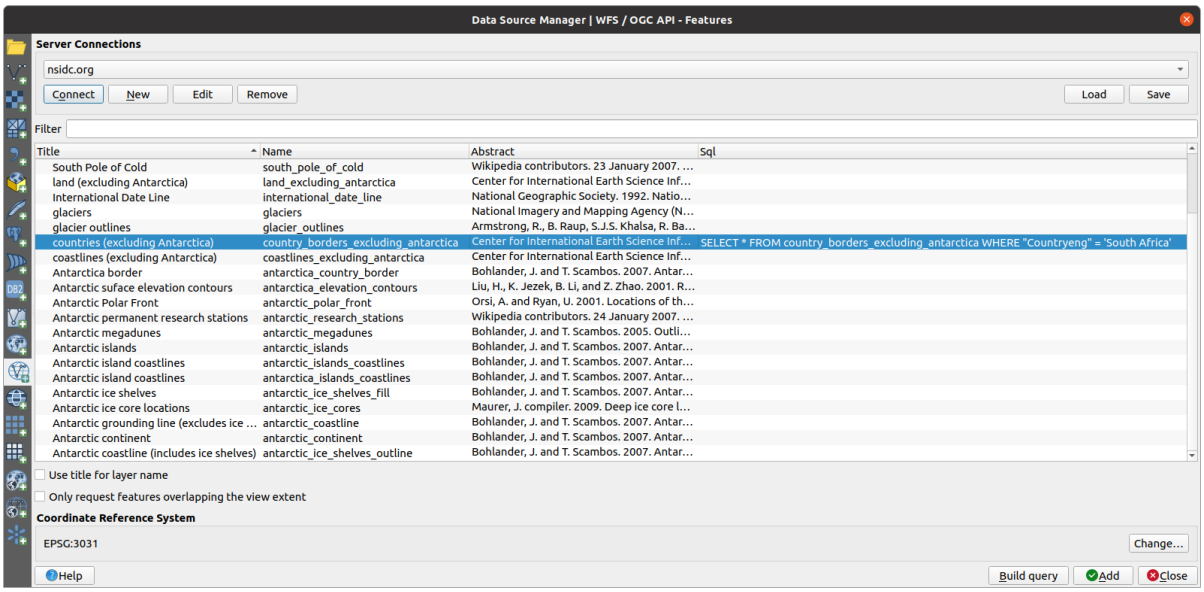


4. В появившемся диалоговом окне введите следующий запрос `SELECT * FROM country_borders_excluding_antarctica WHERE "Countryeng" = 'South Africa'` в ячейке *SQL Statement*.

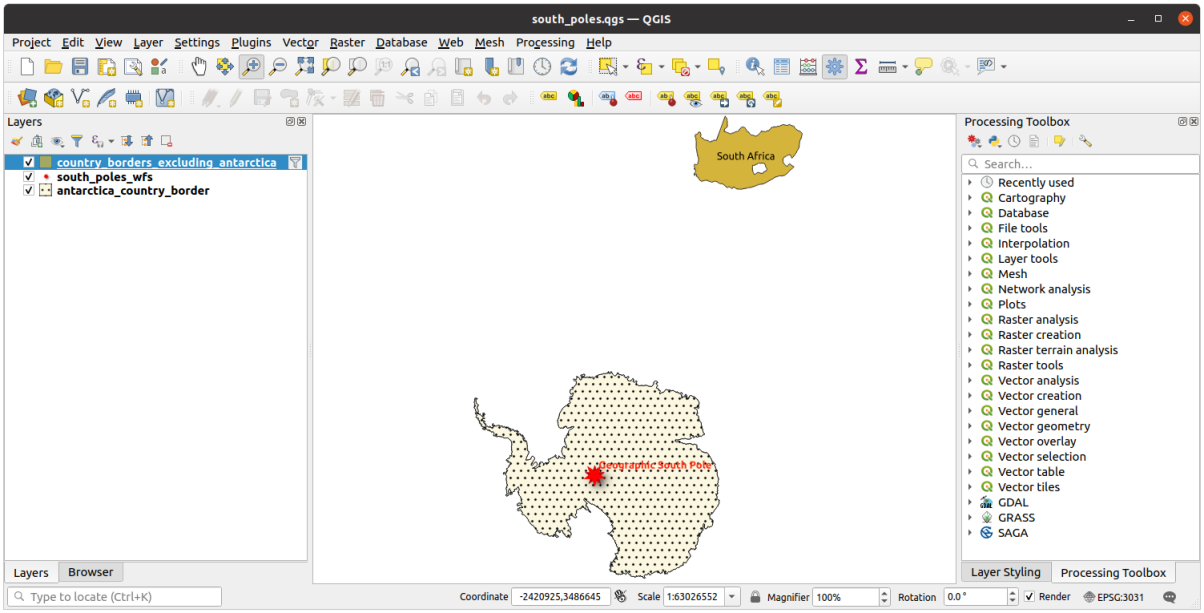


5. Кликните на кнопку *OK*.

6. Использованное выражение будет отображено как *Sql* значение целевого слоя:



7. Кликните на *Add* с выбранным слоем, как было показано выше. Только страна с `Countryeng` значением `South Africa` загрузится из этого слоя:



Вы заметили  значок рядом со слоем `country_borders_excluding_antarctica`? Это указывает на то, что загруженный слой отфильтрован и не отображает в проекте все свои объекты.

8. Вам не обязательно это делать, но если вы попробуете оба метода, вы заметите, что это намного быстрее, чем загружать все страны перед их фильтрацией!

О доступности WFS

Редко можно найти объекты, хранимые в WFS, которые вам нужны, если ваши потребности очень специфичны. Причина, по которой WFS относительно редки, заключается в том, что для описания объекта в целом необходимо передавать большие объемы данных. Поэтому не очень рентабельно размещать WFS вместо WMS, которая предоставляет только изображения.

Поэтому наиболее распространенный тип WFS, с которым вы столкнетесь, вероятно, будет находиться в локальной сети или даже на вашем собственном компьютере, а не в Интернете.

10.2.3 В заключении

Слои WFS предпочтительнее чем слои WMS, если вам нужен прямой доступ к атрибутам и геометрии слоев. Тем не менее, с учетом объема данных, которые необходимо загрузить (что приводит к проблемам со скоростью, а также к отсутствию легкодоступных серверов WFS), не всегда возможно использовать WFS вместо WMS.

10.2.4 Что дальше?

Далее вы увидите, как использовать QGIS Server для предоставления услуг OGC.

ГЛАВА 11

Module: QGIS Server

Module contributed by Tudor Bărăscu.

In this module, we'll cover how to install and use QGIS Server.

To learn more about QGIS Server, read the QGIS-Server-manual.

11.1 Lesson: Install QGIS Server

The goal for this lesson: To learn how to install **QGIS Server** on Debian Stretch. With negligible variations you can also follow it for any Debian based distribution like Ubuntu and its derivatives.

Примечание: In Ubuntu you can use your regular user, prepending `sudo` to commands requiring admin permissions. In Debian you can work as admin (`root`), without using `sudo`.

11.1.1  Follow Along: Install from packages

In this lesson we're going to do only the install from packages as shown here .

Install QGIS Server with:

```
apt install qgis-server --no-install-recommends --no-install-suggests

# if you want to install server plugins, also:
apt install python-qgis
```

QGIS Server should be used in production without QGIS Desktop (with the accompanying X Server) installed on the same machine.

11.1.2  Follow Along: QGIS Server Executable

The QGIS Server executable is `qgis_mapserv.fcgi`. You can check where it has been installed by running `find / -name 'qgis_mapserv.fcgi'` which should output something like `/usr/lib/cgi-bin/qgis_mapserv.fcgi`.

Optionally, if you want to do a command line test at this time you can run the `/usr/lib/cgi-bin/qgis_mapserv.fcgi --version` command which should output something like:

```
QGIS 3.21.0-Master 'Master' (1c70953f1e)
QGIS code revision 1c70953f1e
Qt version 5.15.2
Python version 3.9.5
GDAL/OGR version 3.2.2
PROJ version 7.2.1
EPSG Registry database version v10.008 (2020-12-16)
GEOS version 3.9.0-CAPI-1.16.2
SQLite version 3.34.1
OS Ubuntu 21.04
```

We'll see later on how to make WMS requests.

11.1.3  HTTP Server Configuration

In order to access on the installed QGIS server from an Internet Browser we need to use an HTTP server. The Apache HTTP Server installation process is detailed in httpserver section.

Примечание: If you installed QGIS Server without running an X Server (included in Linux Desktop) and if you also want to use the `GetPrint` command then you should install a fake X Server and tell QGIS Server to use it. You can do that by following the Xvfb installation process.

11.1.4  Follow Along: Create another virtual host

Let's create another Apache virtual host pointing to QGIS Server. You can choose whatever name you like (`coco.bango`, `super.duper.training`, `example.com`, etc.) but for simplicity sake we're going to use `myhost`.

- Let's set up the `myhost` name to point to the localhost IP by adding `127.0.0.1 x` to the `/etc/hosts` with the following command: `sh -c "echo '127.0.0.1 myhost' >> /etc/hosts"` or by manually editing the file with `gedit /etc/hosts`.
- We can check that `myhost` points to the localhost by running in the terminal the `ping myhost` command which should output:

```
qgis@qgis:~$ ping myhost
PING myhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.029 ms
```

- Let's try if we can access QGIS Server from the `myhost` site by doing: `curl http://myhost/cgi-bin/qgis_mapserv.fcgi` or by accessing the url from your Debian box browser. You will probably get:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /cgi-bin/qgis_mapserv.fcgi was not found on this server.</p>
<hr>
<address>Apache/2.4.25 (Debian) Server at myhost Port 80</address>
</body></html>
```

- Apache doesn't know that he's supposed to answer requests pointing to the server named `myhost`. In order to setup the virtual host the simplest way would be to make a `myhost.conf` file in the `/etc/apache2/sites-available` directory that has the same content as `qgis.demo.conf` except for the `ServerName` line that should be `ServerName myhost`. You could also change where the logs go as otherwise the logs for the two virtual hosts would be shared but this is optional.
- Let's now enable the virtual host with `a2ensite myhost.conf` and then reload the Apache service with `service apache2 reload`.
- If you try again to access the http://myhost/cgi-bin/qgis_mapserv.fcgi url you'll notice everything is working now!

11.1.5 In Conclusion

You learned how to install different QGIS Server versions from packages, how to configure Apache with QGIS Server, on Debian based Linux distros.

11.1.6 What's Next?

Now that you've installed QGIS Server and it's accessible through the HTTP protocol, we need to learn how to access some of the services it can offer. The topic of the next lesson is to learn how to access QGIS Server WMS services.

11.2 Lesson: Serving WMS

The data used for this exercise are available in the `qgis-server-tutorial-data` subdirectory of the *training data* you downloaded. For convenience and to avoid possible permissions problems, we will assume that these files are stored in `/home/qgis/projects` directory. Hence, adapt the following instructions to your path.

The demo data contains a QGIS project named `world.qgs` that is already prepared to be served with QGIS Server. If you want to use your own project or you want to learn how a project is prepared see the Creatingwmsfromproject section.

Примечание: This module presents the URLs so that the audience can easily distinguish the parameters and parameters values. While the normal format is:

```
...&field1=value1&field2=value2&field3=value3
```

this tutorial uses:

```
&field1=value1
&field2=value2
&field3=value3
```

Pasting them into Mozilla Firefox works properly but other web browsers like Chrome may add unwanted spaces between the `field:parameter` pairs. So, if you encounter this issue you can either use Firefox or modify the URLs so that they're in one line format.

Let's make a WMS GetCapabilities request in the web browser or with curl:

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetCapabilities
&map=/home/qgis/projects/world.qgs
```

In the Apache config from the previous lesson the `QGIS_PROJECT_FILE` variable sets the default project to `/home/qgis/projects/world.qgs`. However, in the above request we made use of the **map** parameter to be explicit and to show it can be used to point at any project. If you delete the **map** parameter from the above request QGIS Server will output the same response.

By pointing any WMS client to the `GetCapabilities` URL, the client gets in response an XML document with metadata of the Web Map Server's information, e.g. what layers does it serve, the geographical coverage, in what format, what version of WMS etc.

As QGIS is also a ogc-wms you can create a new WMS server connection with the help of the above GetCapabilities url. See the *Занятие: Сервисы веб-карт* or the ogc-wms-servers section on how to do it.

By adding the `countries` WMS layer to your QGIS project you should get an image like the one below:

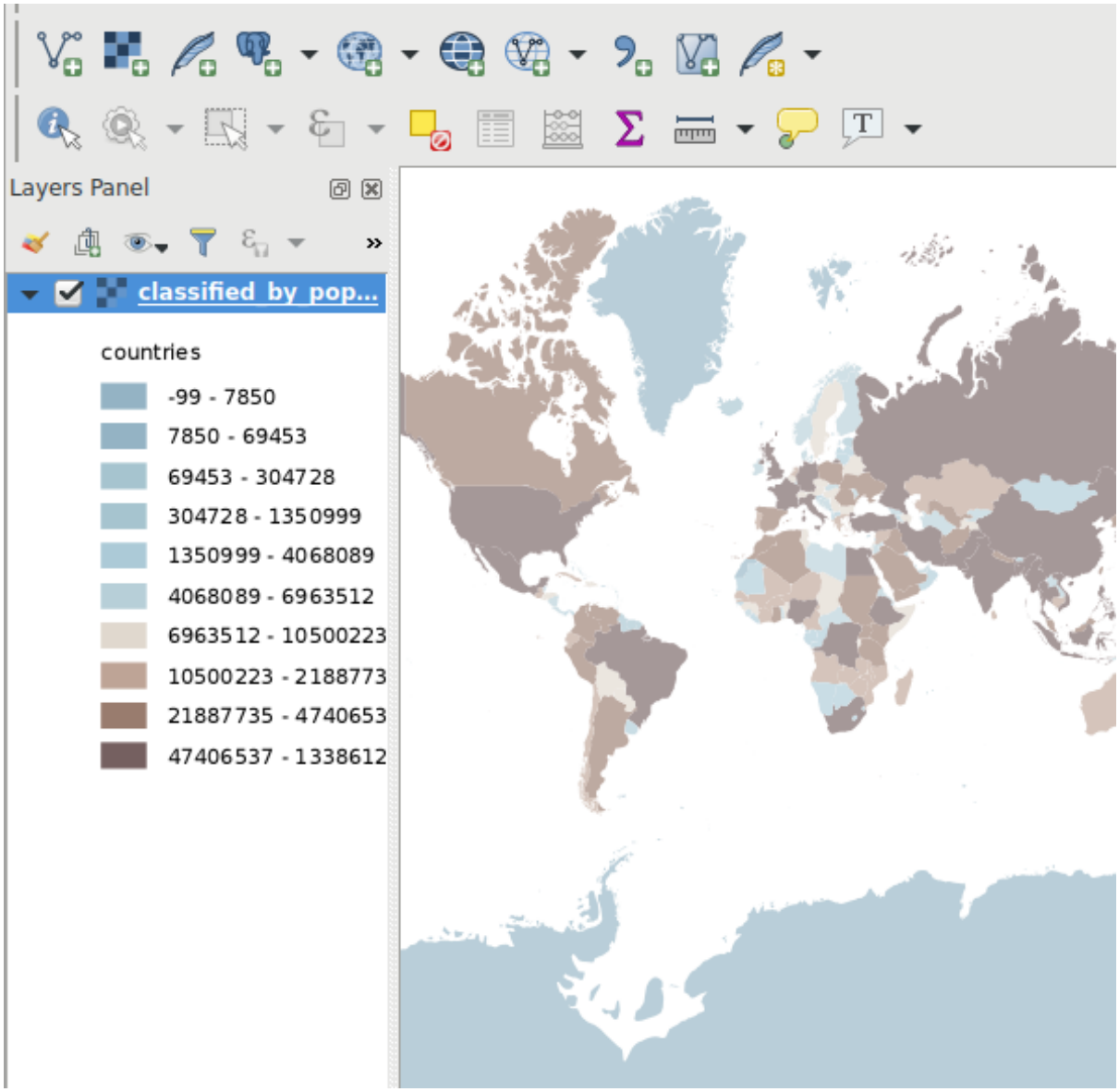


Рис. 11.1: QGIS Desktop consuming the QGIS Server countries layer WMS service

Примечание: QGIS Server serves layers that are defined in the `world.qgs` project. By opening the project with QGIS you can see there are multiple styles for the countries layer. QGIS Server is also aware of this and you can choose the style you want in your request. The `classified_by_population` style was chosen in the above image.

11.2.1 Логгирование

When you're setting up a server, the logs are always important as they show you what's going on. We have setup in the `*.conf` file the following logs:

- QGIS Server log at `/logs/qgisserver.log`
- `qgisplatform.demo` Apache access log at `qgisplatform.demo.access.log`
- `qgisplatform.demo` Apache error log at `qgisplatform.demo.error.log`

The log files are simply text files so you can use a text editor to check them out. You can also use the `tail` command in a terminal: `sudo tail -f /logs/qgisserver.log`.

This will continuously output in the terminal what's written in that log file. You can also have three terminals opened for each of the log files like so:

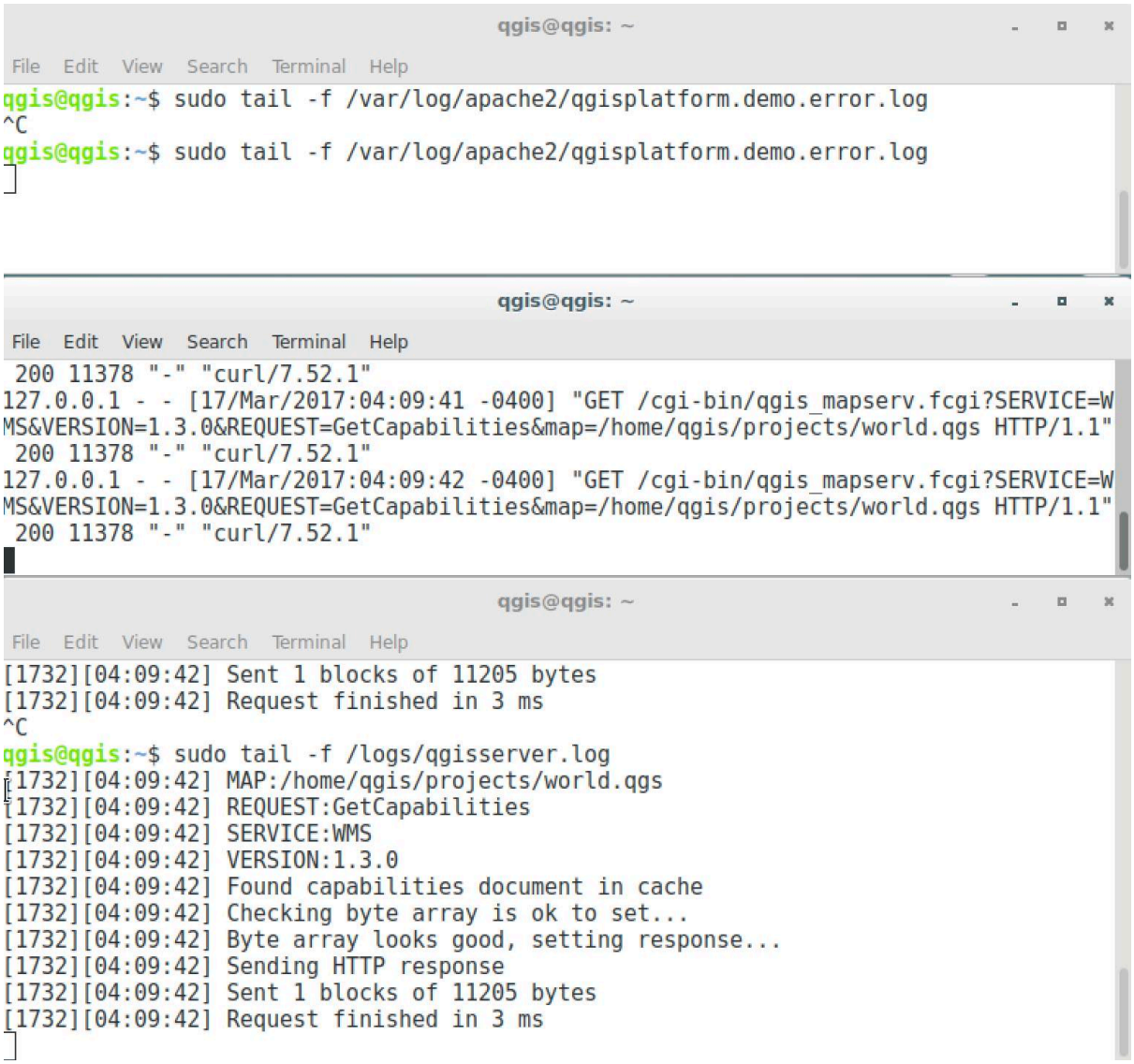


Рис. 11.2: Using the `tail` command to visualise QGIS Server logs output

When you use QGIS Desktop to consume the QGIS Server WMS services you will see all the requests QGIS sends to the Server in the access log, the errors of QGIS Server in the QGIS Server log etc.

Примечание:

- If you look at the logs in the following sections you should get a better understanding on what's happening.
- By restarting Apache while looking in the QGIS Server log you can find some extra pointers on how things work.

11.2.2 GetMap requests

In order to display the `countries` layer, QGIS Desktop, like any other WMS client, is using `GetMap` requests.

A simple request looks like:

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?MAP=/home/qgis/projects/world.qgs
&SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&BBOX=-432786,4372992,3358959,7513746
&SRS=EPSG:3857
&WIDTH=665
&HEIGHT=551
&LAYERS=countries
&FORMAT=image/jpeg
```

The above request should output the following image:

Figure: simple GetMap request to QGIS Server

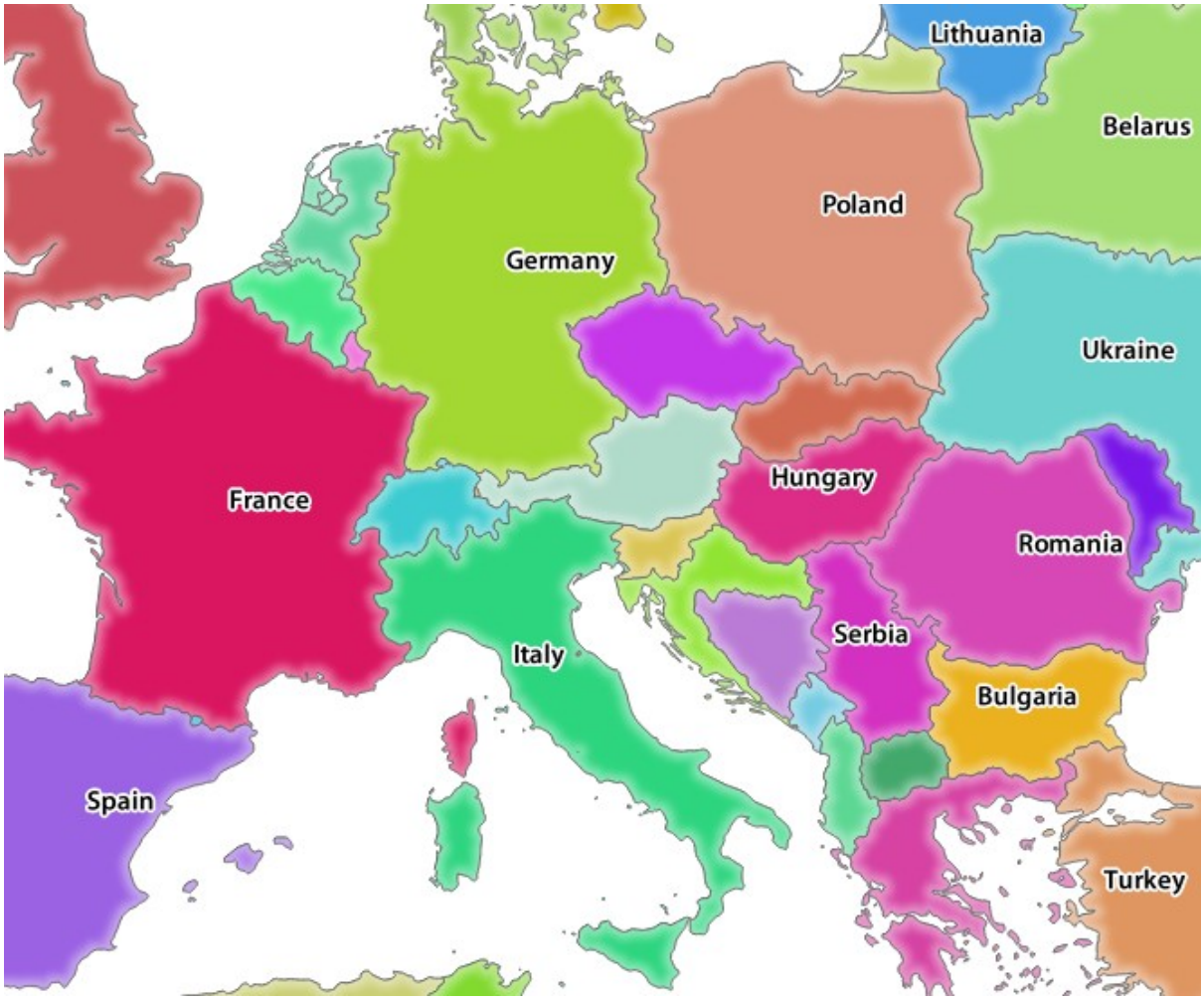


Рис. 11.3: Qgis Server response after a simple GetMap request

11.2.3 Try Yourself Change the Image and Layers parameters

Based on the request above, let's replace the `countries` layer with another.

In order to see what other layers are available you could open up the `world.qgs` project in QGIS and look at its contents. Keep in mind though that the WMS clients don't have access to the QGIS project, they just look at the capabilities document contents.

Also, there's a configuration option so that some of the layers existing in the QGIS project are ignored by QGIS when serving the WMS service.

So, you could look at the layer list when you point QGIS Desktop to the `GetCapabilities` URL or you could try yourself finding other layer names in the `GetCapabilities` XML response.

One of the layer names that you could find and works is `countries_shapeburst`. You may find others but keep in mind some may not be visible at such a small scale so you could get a blank image as response.

You can also play around with others parameters from above, like changing the returned image type to `image/png`.

11.2.4  Follow Along: Use Filter, Opacities and Styles parameters

Let's do another request that adds another layer, some basic parameters, **FILTER** and **OPACITIES**, but also uses the standard STYLES parameter.

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?MAP=/home/qgis/projects/world.qgs
&SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&BBOX=-432786,4372992,3358959,7513746
&SRS=EPSG:3857
&WIDTH=665
&HEIGHT=551
&FORMAT=image/jpeg
&LAYERS=countries,countries_shapeburst
&STYLES=classified_by_name,blue
&OPACITIES=255,30
&FILTER=countries:"name" IN ( 'Germany' , 'Italy' )
```

The above request should output the following image:

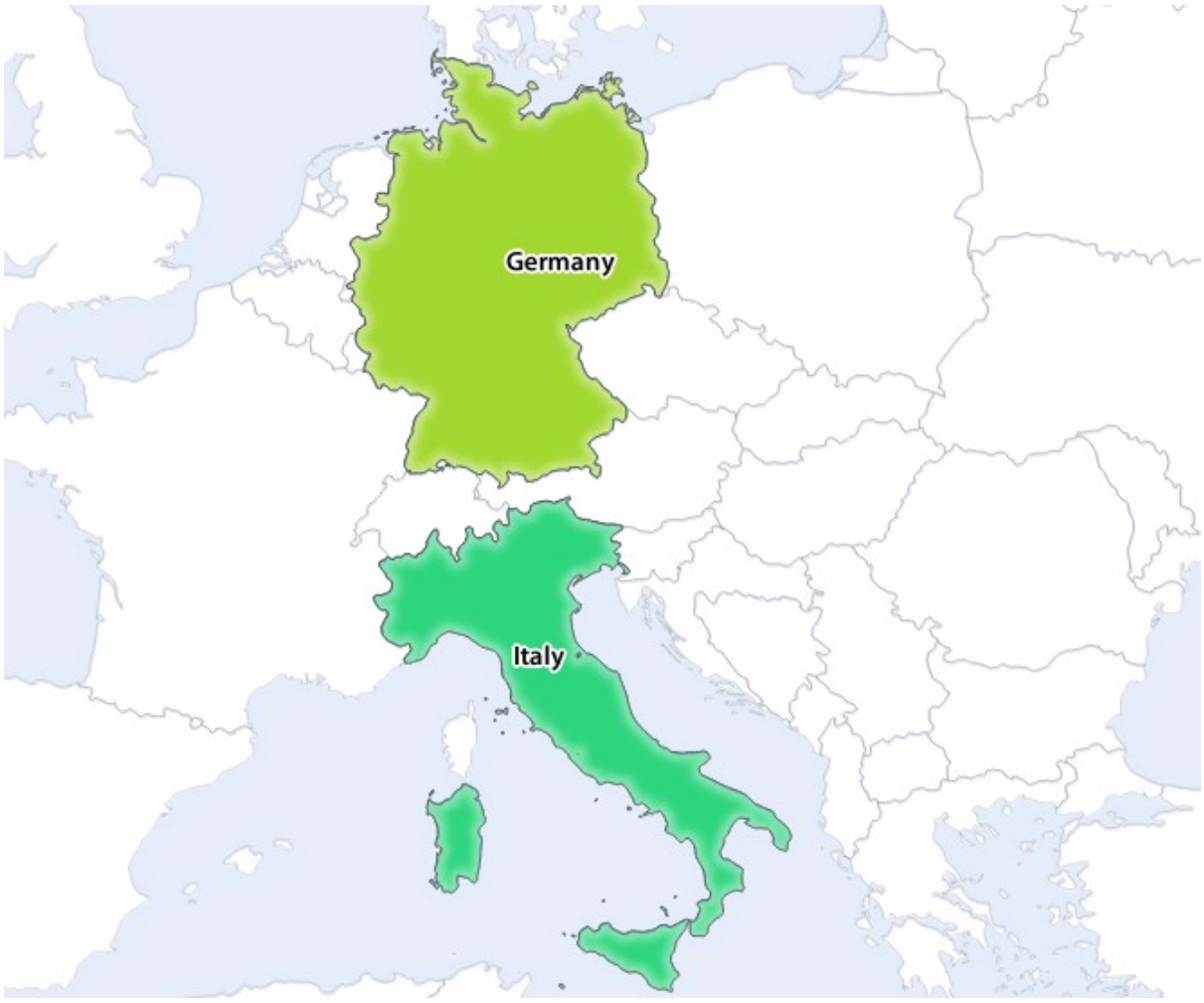


Рис. 11.4: Response to a GetMap request with FILTER and OPACITIES parameters

As you can see from the above image, among other things, we told QGIS Server to render only **Germany** and **Italy** from the countries layer.

11.2.5  Follow Along: Use Redlining

Let's do another GetMap request that makes use of the redlining feature and of the **SELECTION** parameter detailed in the Basics section:

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?MAP=/home/qgis/projects/world.qgs
&SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&BBOX=-432786,4372992,3358959,7513746
&SRS=EPSG:3857
&WIDTH=665
&HEIGHT=551
&LAYERS=countries,countries_shapeburst
&FORMAT=image/jpeg
&HIGHLIGHT_GEOM=POLYGON((590000 6900000, 590000 7363000, 2500000 7363000, 2500000␣
↪6900000, 590000 6900000))
&HIGHLIGHT_SYMBOL=<StyledLayerDescriptor><UserStyle><Name>Highlight</Name>
↪<FeatureTypeStyle><Rule><Name>Symbol</Name><LineSymbolizer><Stroke><SvgParameter␣
↪name="stroke">%233a093a</SvgParameter><SvgParameter name="stroke-opacity">1</
↪SvgParameter><SvgParameter name="stroke-width">1.6</SvgParameter></Stroke></
↪LineSymbolizer></Rule></FeatureTypeStyle></UserStyle></StyledLayerDescriptor>
&HIGHLIGHT_LABELSTRING=QGIS Tutorial
&HIGHLIGHT_LABELSIZE=30
&HIGHLIGHT_LABELCOLOR=%23000000
&HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFF
&HIGHLIGHT_LABELBUFFERSIZE=3
&SELECTION=countries:171,65
```

Pasting the above request in your web browser should output the following image:



Рис. 11.5: Response to a request with the REDLINING feature and SELECTION parameter

You can see from the above image that the countries with the 171 and 65 ids were highlighted in yellow (Romania and France) by using the **SELECTION** parameter and we used the **REDLINING** feature to overlay a rectangle with the **QGIS Tutorial** label.

11.2.6 GetPrint requests

One very nice feature of QGIS Server is that it makes use of the QGIS Desktop print layouts. You can learn about it in the wms_getprint section.

If you open the `world.qgs` project with QGIS Desktop you will find a print layout named `Population distribution`. A simplified `GetPrint` request that exemplifies this amazing feature is:

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?map=/home/qgis/projects/world.qgs
&SERVICE=WMS
&VERSION=1.3.0&
REQUEST=GetPrint
&FORMAT=pdf
&TRANSPARENT=true
&SRS=EPSG:3857
&DPI=300
&TEMPLATE=Population distribution
&map0:extent=-432786,4372992,3358959,7513746
&LAYERS=countries
```

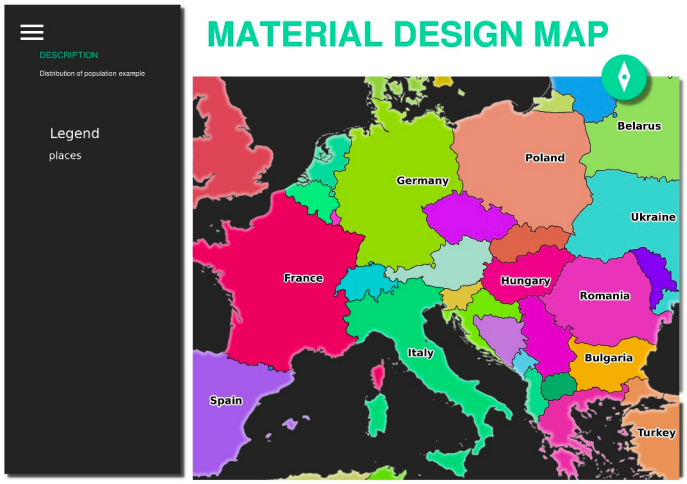


Рис. 11.6: Shows the pdf resulted from the above GetPrint request

Naturally, it's hard to write your `GetMap`, `GetPrint` etc. requests.

QGIS Web Client or QWC is a Web client project that can work alongside QGIS Server so that you can publish your projects on the Web or help you create QGIS Server requests for a better understanding about the possibilities.

You can install it like this:

- As user `qgis` go to the home directory with `cd /home/qgis`.
- Download the QWC project from here and unzip it.
- Make a symbolic link to the `/var/www/html` directory as it's the `DocumentRoot` that we've setup in the virtual host configuration. If you unzipped the archive under `/home/qgis/Downloads/QGIS-Web-Client-master` we can do that with `sudo ln -s /home/qgis/Downloads/QGIS-Web-Client-master /var/www/html/`.
- Access http://qgisplatform.demo/QGIS-Web-Client-master/site/qgiswebclient.html?map=/home/qgis/projects/world.qgs from your Web browser.

Now you should be able to see the Map as in the following figure:



Рис. 11.7: QGIS Web Client consuming the world.qgs project

If you click the Print button in QWC you can interactively create `GetPrint` requests. You can also click the `?` icon in the QWC to access the available help so that you can better discover the QWC possibilities.

11.2.7 In Conclusion

You learned how use QGIS Server to provide WMS Services.

11.2.8 What's Next?

Next, you'll see how to use QGIS as a frontend for the famous GRASS GIS.

ГЛАВА 12

Module: GRASS

GRASS (Geographic Resources Analysis Support System) is a well-known open source GIS with a wide array of useful GIS functions. It was first released in 1984, and has seen much improvement and additional functionality since then. QGIS allows you to make use of GRASS“ powerful GIS tools directly.

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. Hence, make sure you have installed QGIS Desktop with Grass support.

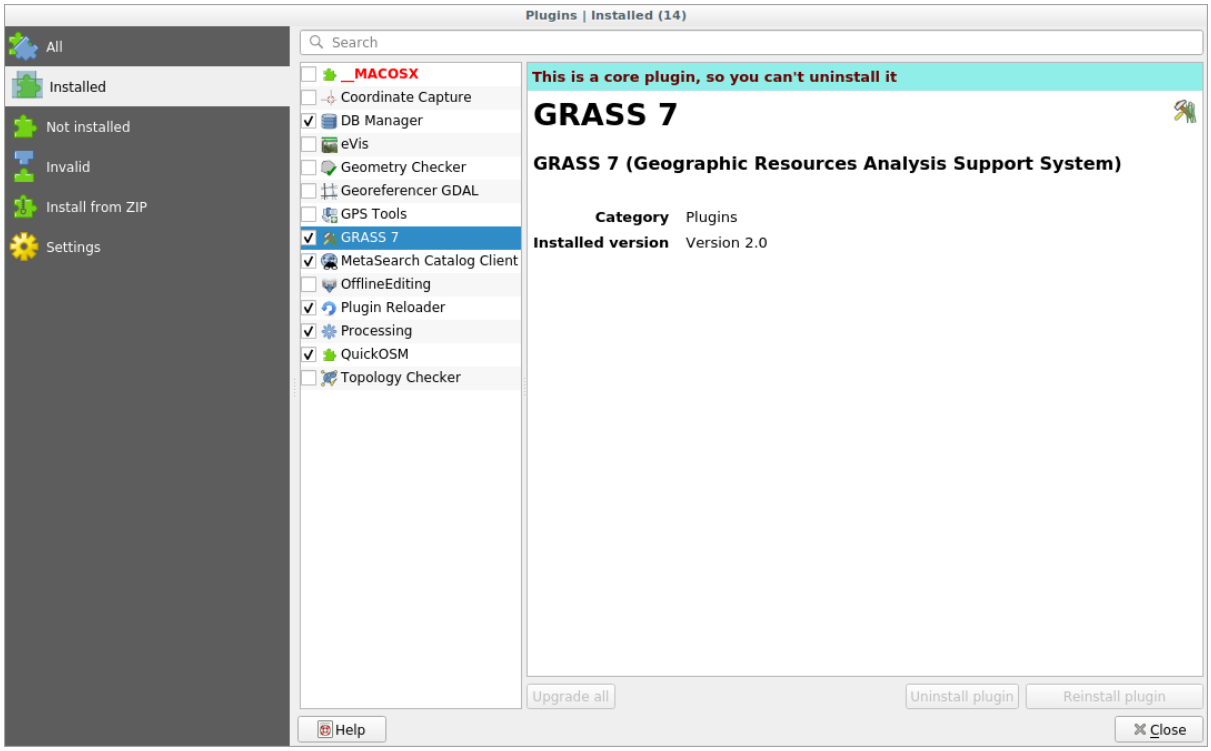
To open a QGIS session with GRASS available on Windows you have to click on the `QGIS Desktop with GRASS` icon.

The goal for this lesson: To begin a GRASS project in QGIS.

12.1.1  Follow Along: Start a New GRASS Session

To launch GRASS from within QGIS, you need to activate it as with any other plugin:

1. First, open a new QGIS project.
2. In the *Plugin Manager*, enable *GRASS* in the list:



The GRASS toolbar and the GRASS panel will appear:

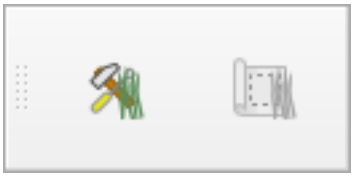


Рис. 12.1: GRASS toolbar



Рис. 12.2: GRASS Panel

The GRASS panel is not active because, 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.

The GRASS database has a straightforward structure, even if at a first look it seems very complicated. The most important thing you should know is that the upper level of the database is the `Location`. Each `Location` can contain different `Mapset`: in **every** `Mapset` you will find the `PERMANENT` Mapset because it is created by default by GRASS. Each `Mapset` contains the data (raster, vector, etc) in a particular structure, but don't worry, GRASS will take care of this for you.

Just remember: `Location` contains `Mapset` that contains the data. For more information visit the GRASS website.

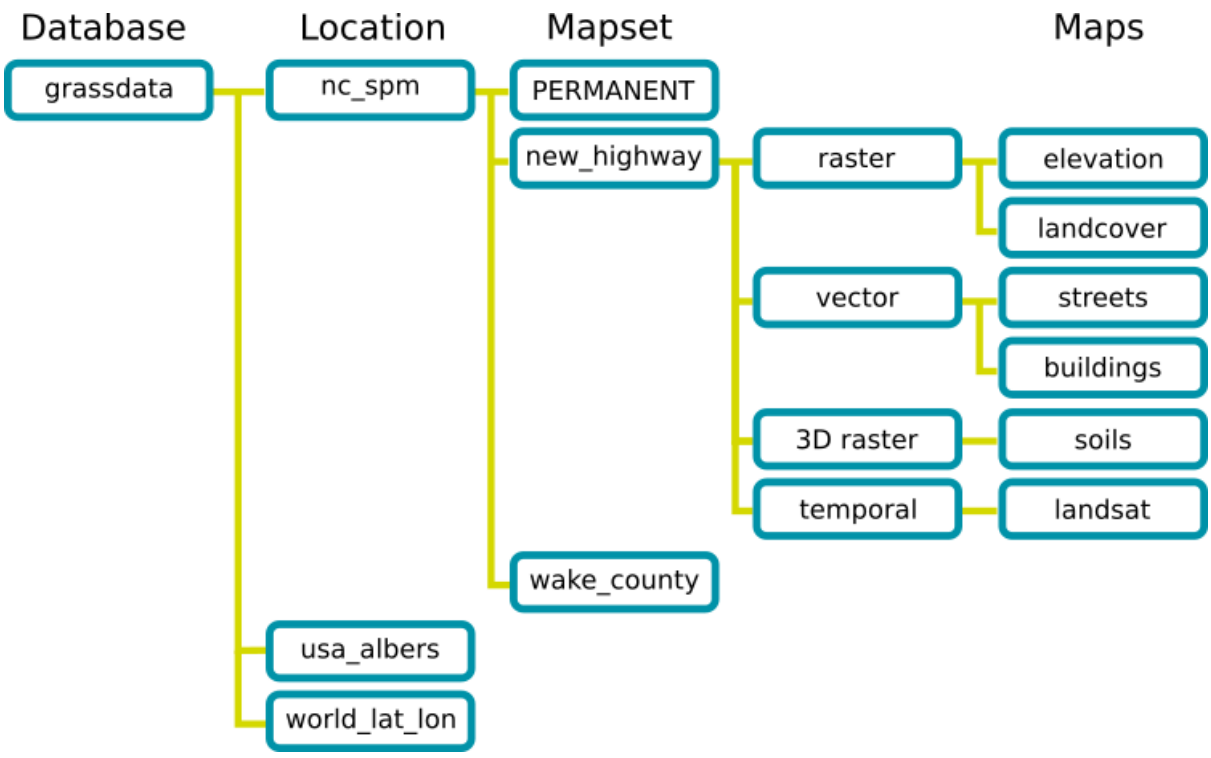
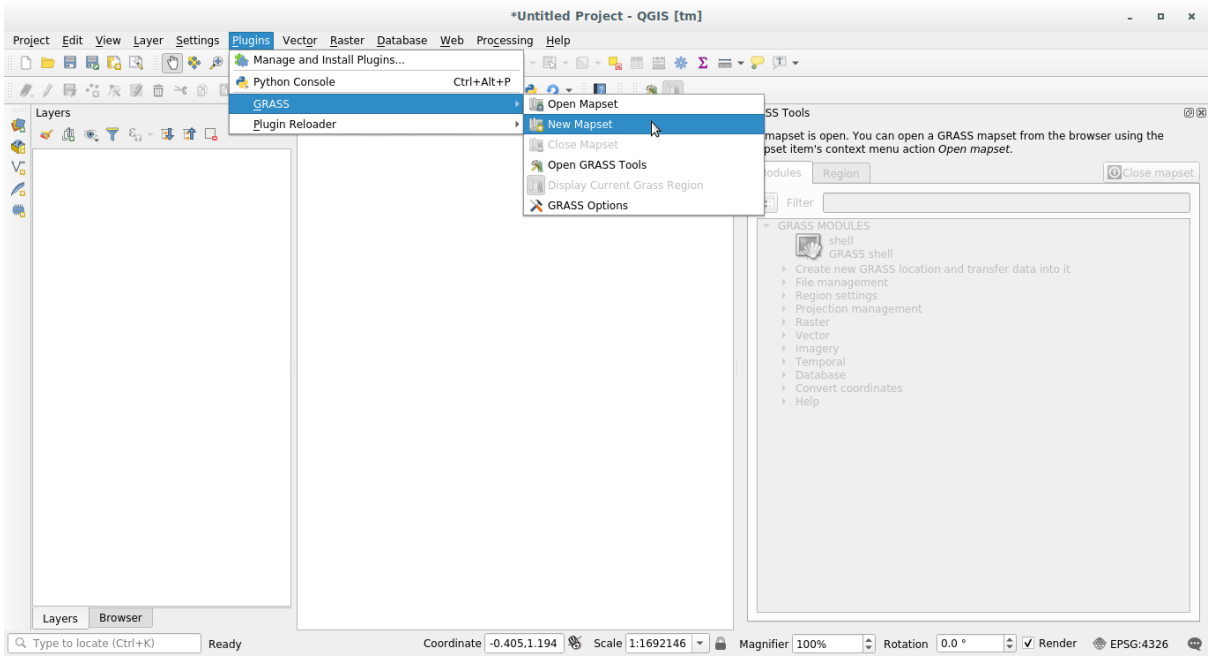


Рис. 12.3: GRASS database structure (from GRASS docs)

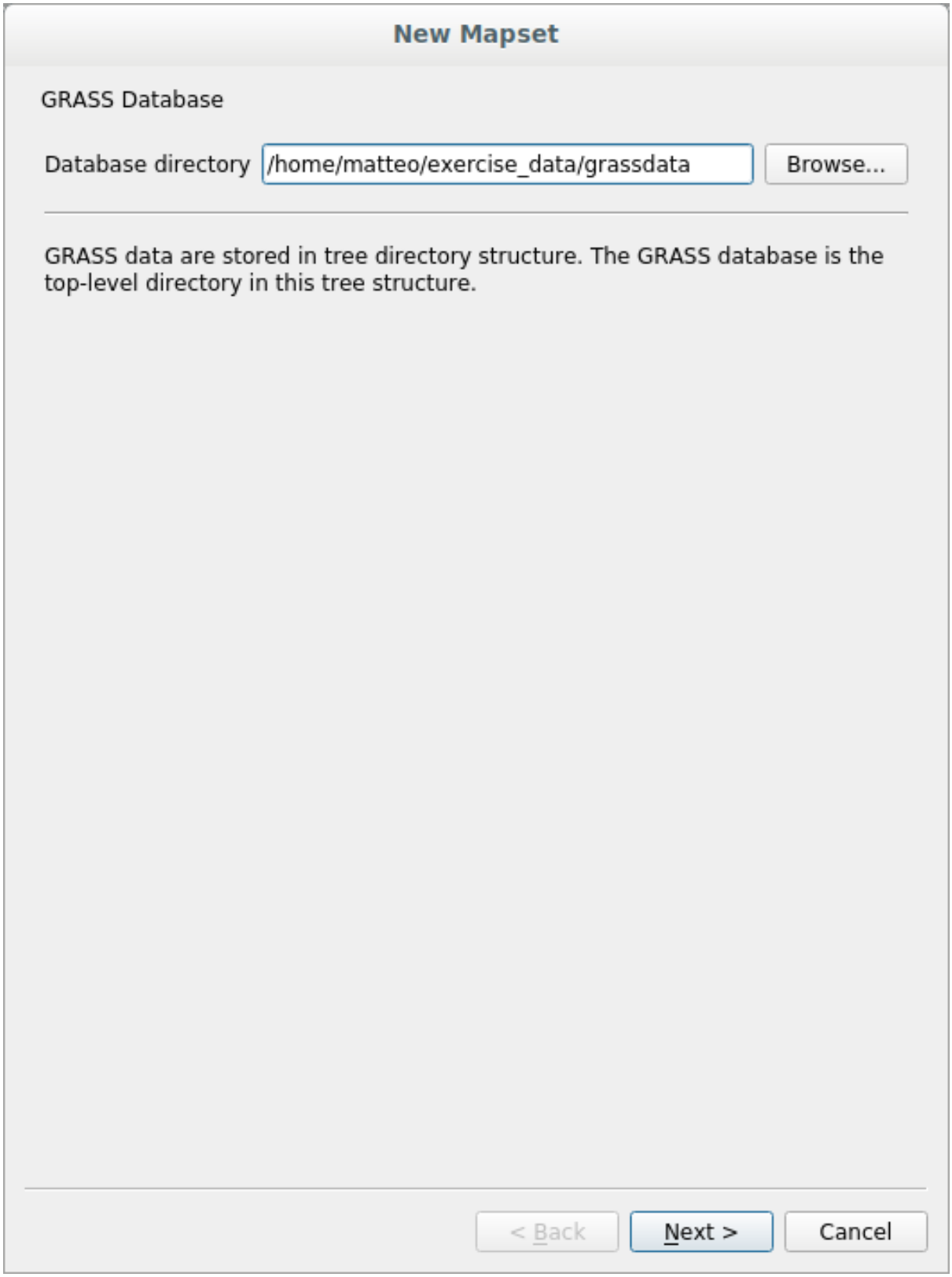
12.1.2 Follow Along: Start a New GRASS Project

1. Click on the *Plugins –> GRASS –> New Mapset* menu:



   You'll be asked to choose the location of the GRASS database.

2. Set it as the directory that will be used by GRASS to set up its database:



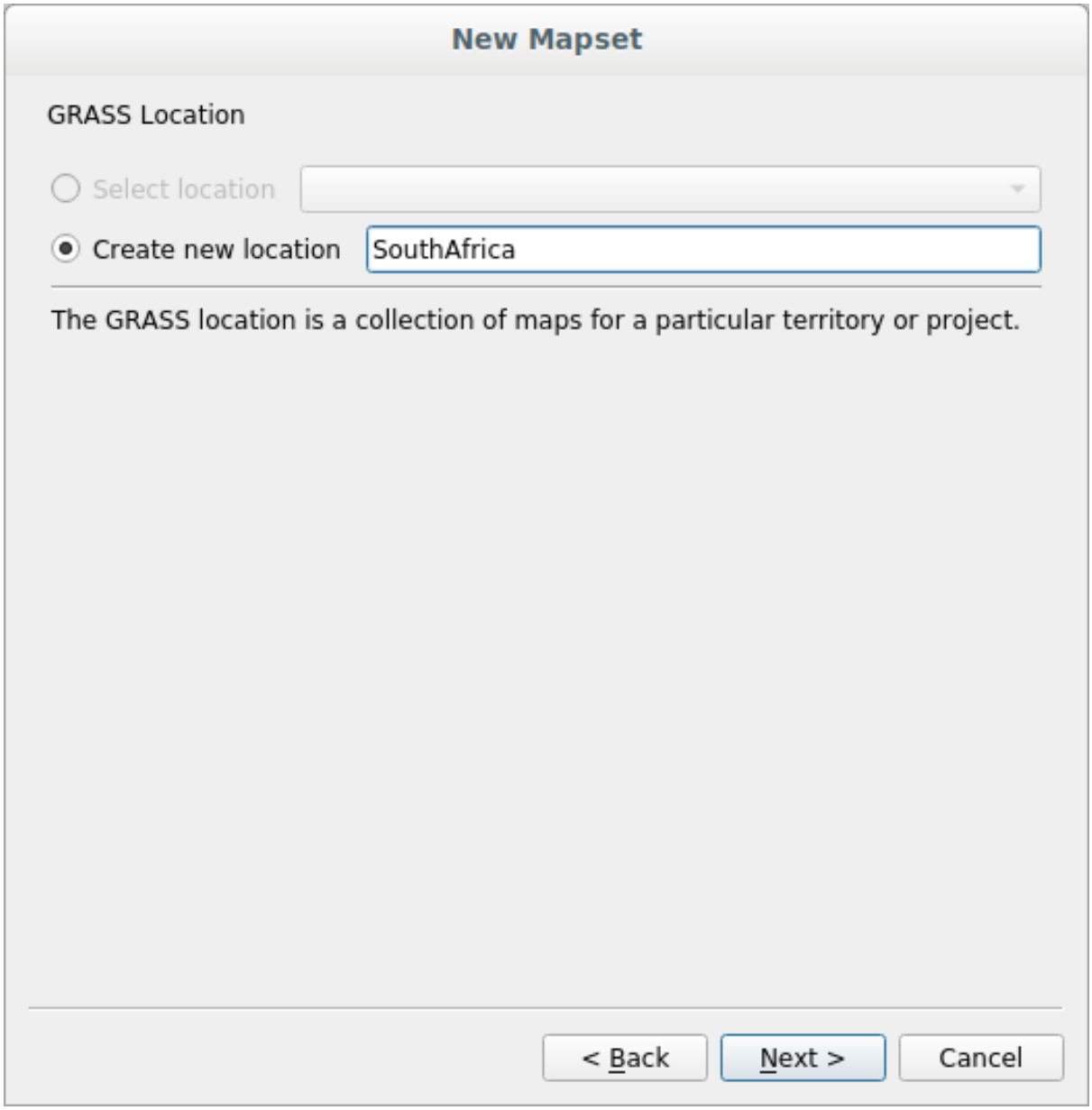
3. Click *Next*.

GRASS needs to create a `Location`, which describes the maximum extent of the geographic area you'll be working in, also known as Grass `Region`.

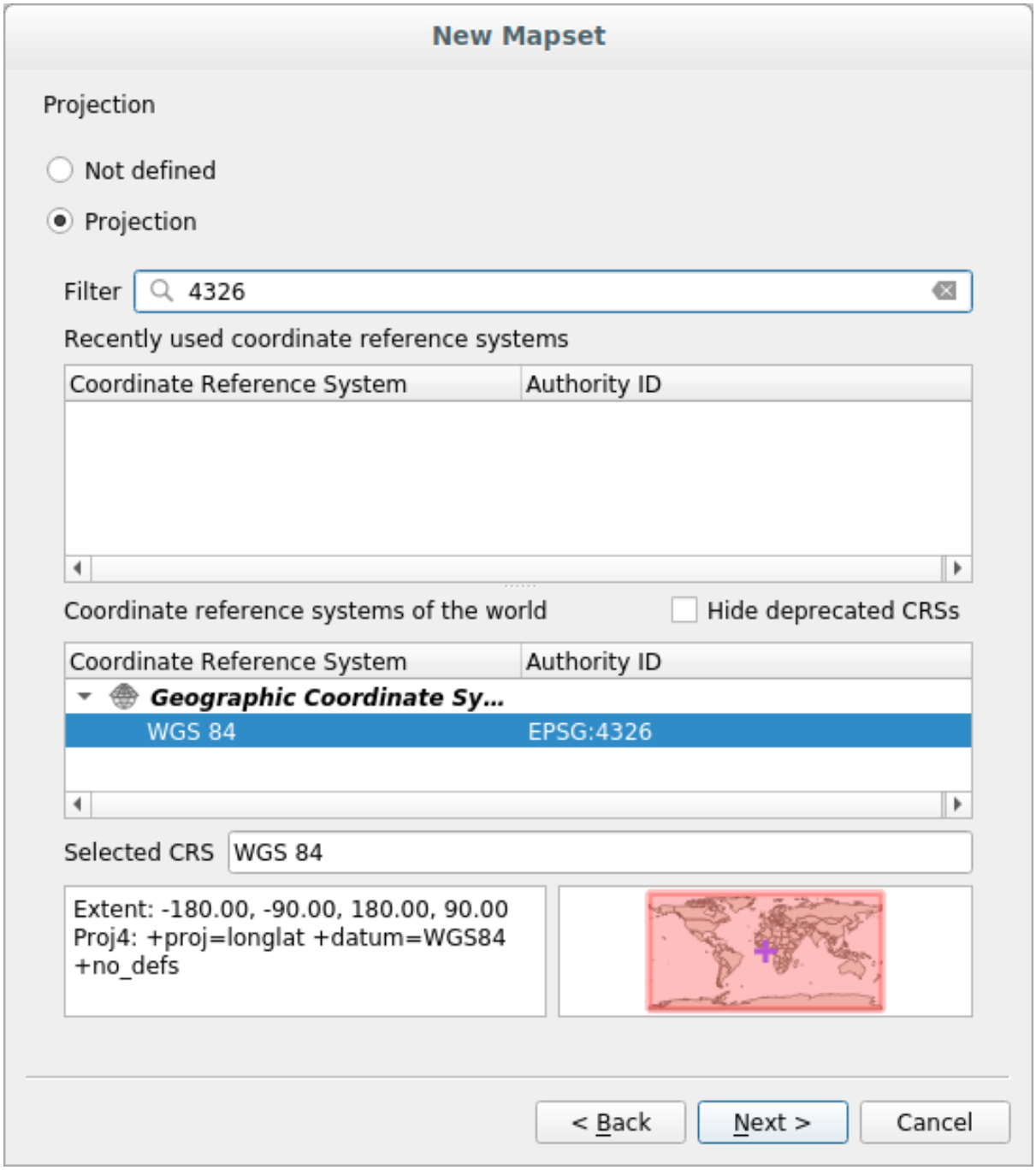
Примечание: the Region is extremely important for GRASS because it describes the area in which all layers will

be taken into account for GRASS. Everything that is outside will not be considered. Don't worry, you can always change the extent of the GRASS Region after the Location has been created

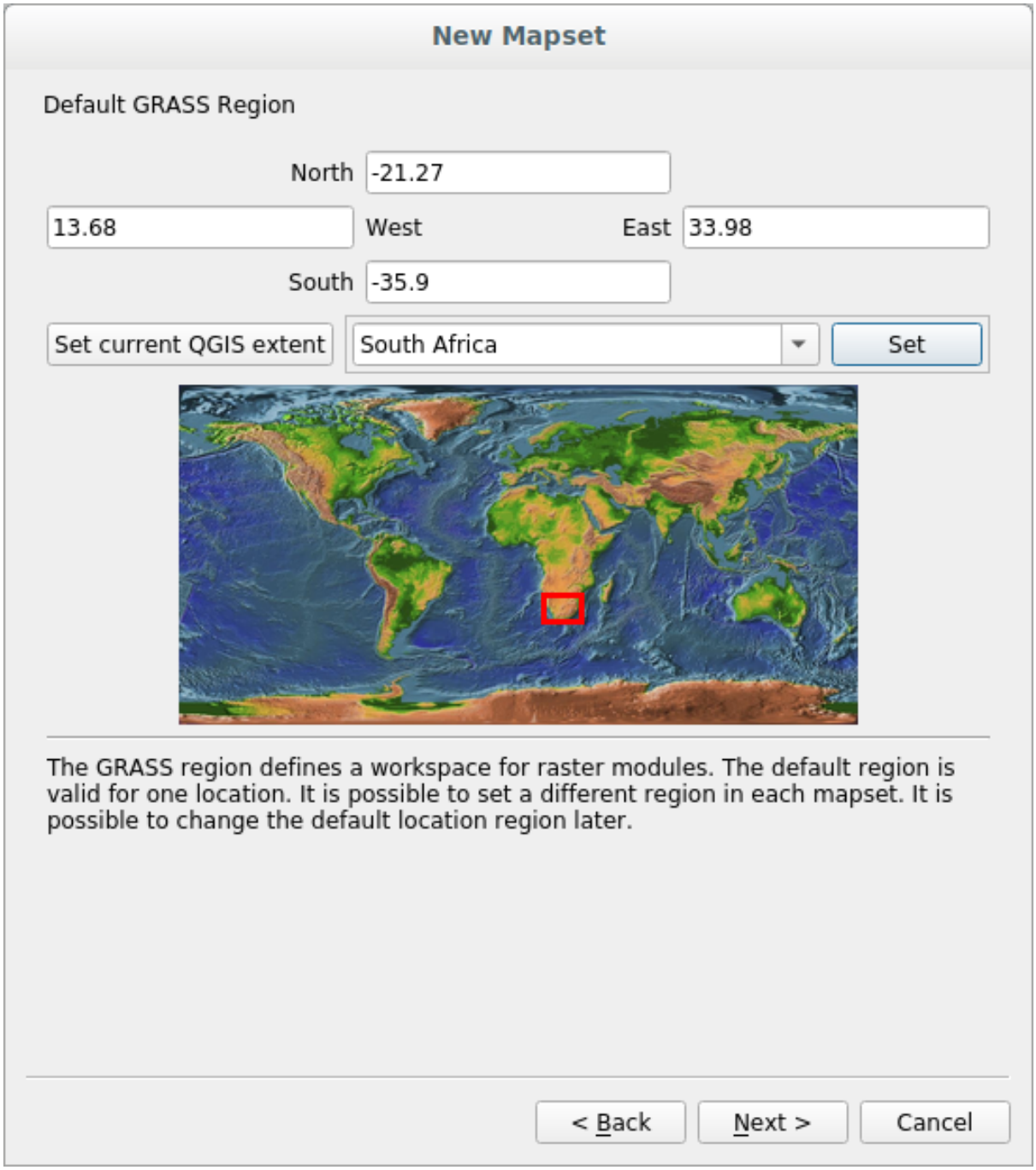
1. Call the new location `SouthAfrica`:



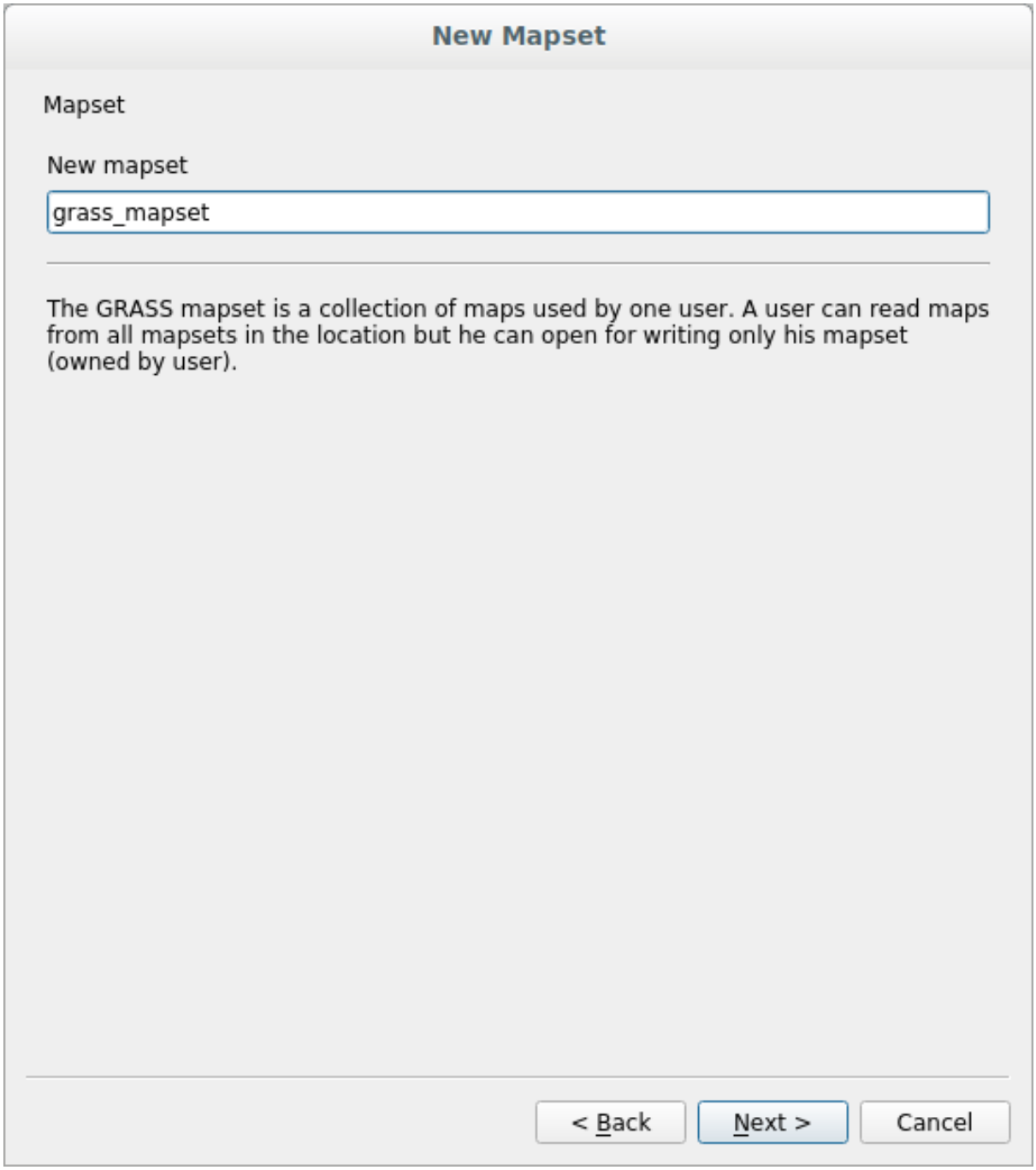
2. Click *Next*.
3. We'll be working with `WGS 84`, so search for and select this CRS:



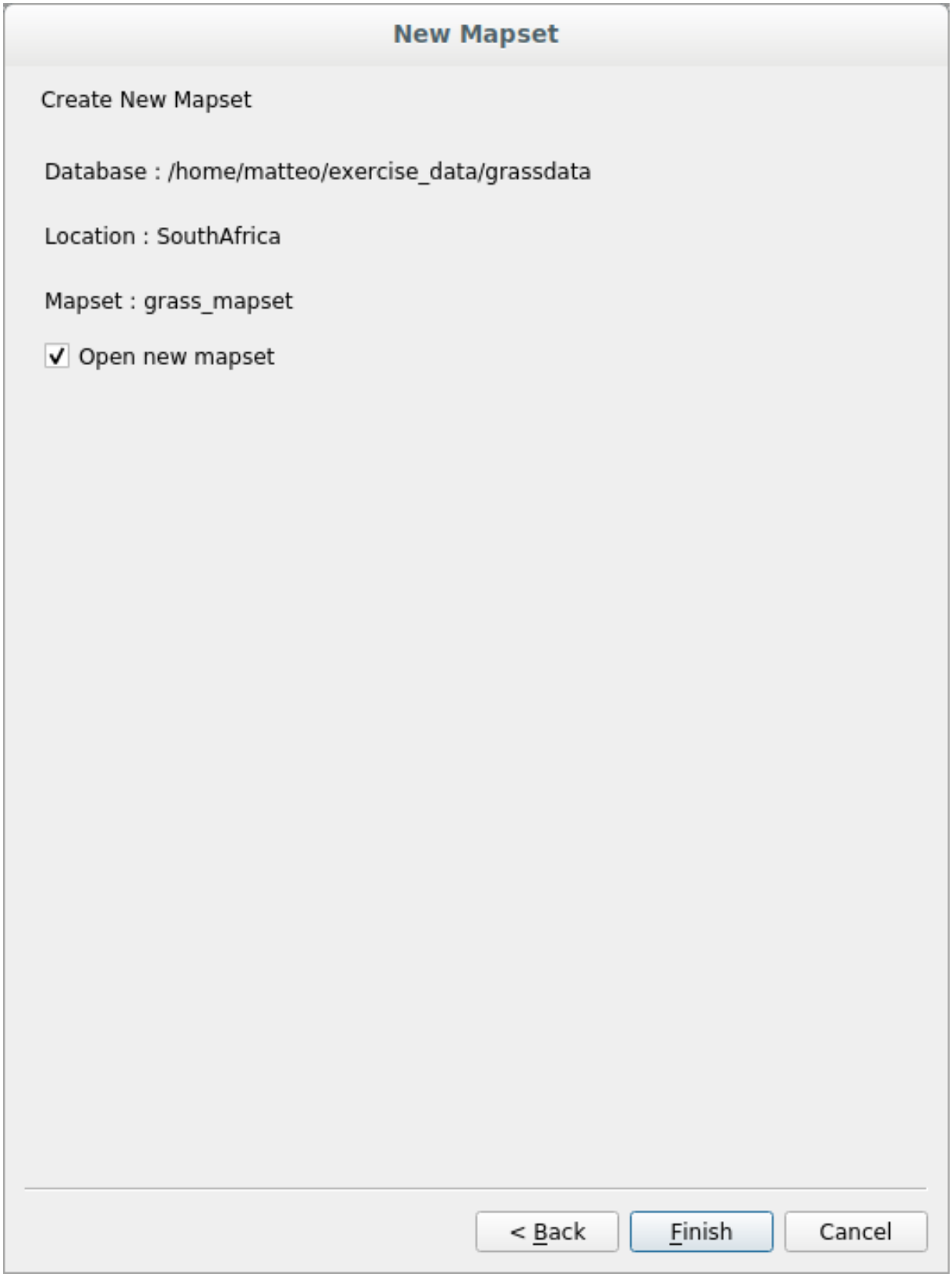
4. Click *Next*.
5. Now select the region *South Africa* from the dropdown and click *Set*:



6. Click *Next*.
7. Create a mapset, which is the map file that you'll be working with.



Once you're done, you'll see a dialog asking with a summary of all the information entered.



8. Click *Finish*.
9. Click *OK* on the success dialog.

You will see that the GRASS Panel will become active and you can start to use all GRASS tools.

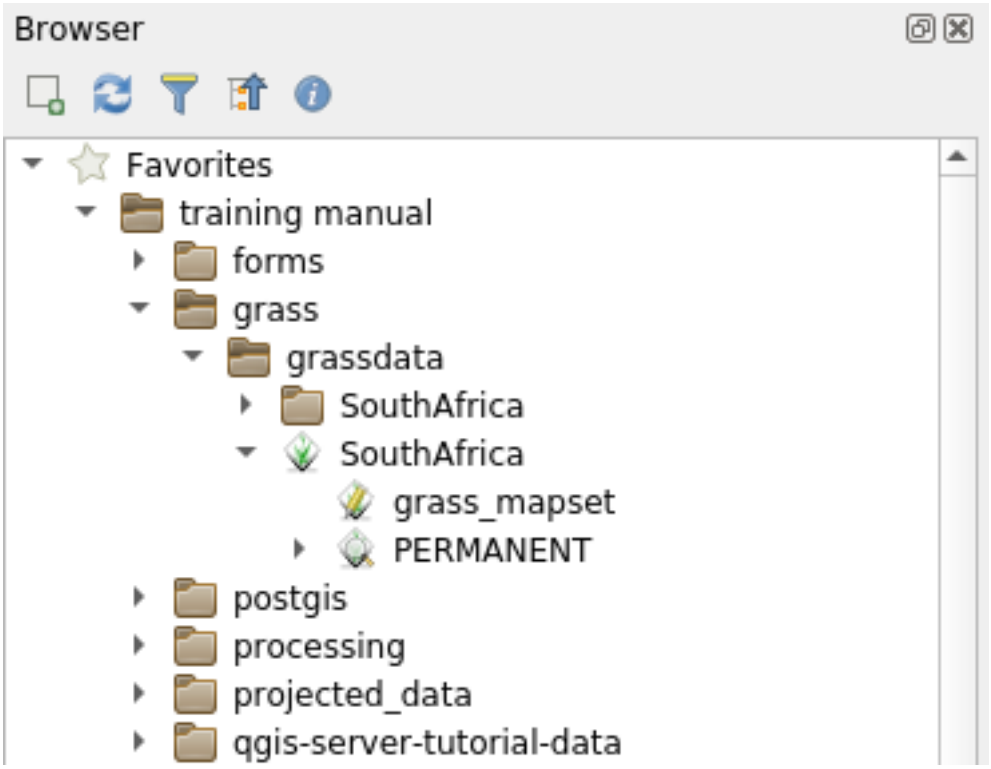
12.1.3 Follow Along: Loading Vector Data into GRASS

You have now a blank map and before you can start to use all the GRASS tools you have to load data into the GRASS database, specifically into the `Mapset`. You cannot use GRASS tools with layer that are not loaded into a GRASS `Mapset`.

There are many different ways to load data in the GRASS database. Let's start with the first one.

Follow Along: Load data using the QGIS Browser

In section *Панель браузера* we saw that the easiest and quickest way to load the data in QGIS is the Browser Panel. GRASS data are recognized from the QGIS Browser as *real* GRASS data and you can notice it because you will see the GRASS icon next to the GRASS Mapset. Moreover you will see the icon next to the Mapset that is opened.

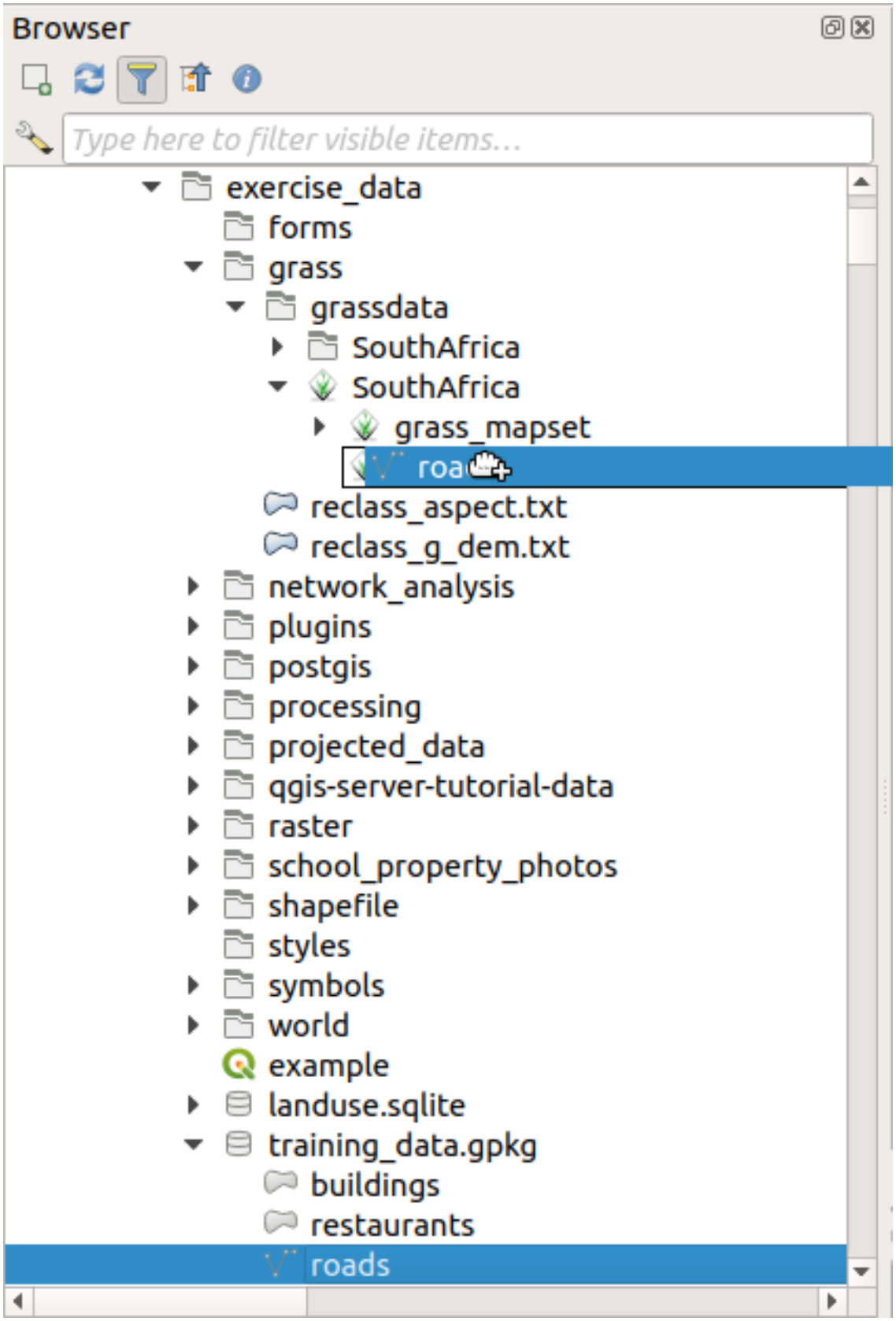


Примечание: You will see a replication of the GRASS Location as normal folder: GRASS `Mapset` data are those within the folder

You can easily **drag and drop** layers from a folder to the GRASS Mapset.

Let's try to import the `roads` layer into the `grass_mapset` Mapset of the `SouthAfrica` Location.

Go to the Browser, and simply drag the `roads` layer from the `training_data.gpkg` GeoPackage file into the `grass_mapset` Mapset.



That's it! If you expand the Mapset you will see the imported `roads` layer. You can now load in QGIS the imported layer like all the other layers.

Совет: You can also load layers from the Layer Legend Panel to Mapset in the Browser Panel. This will speed up

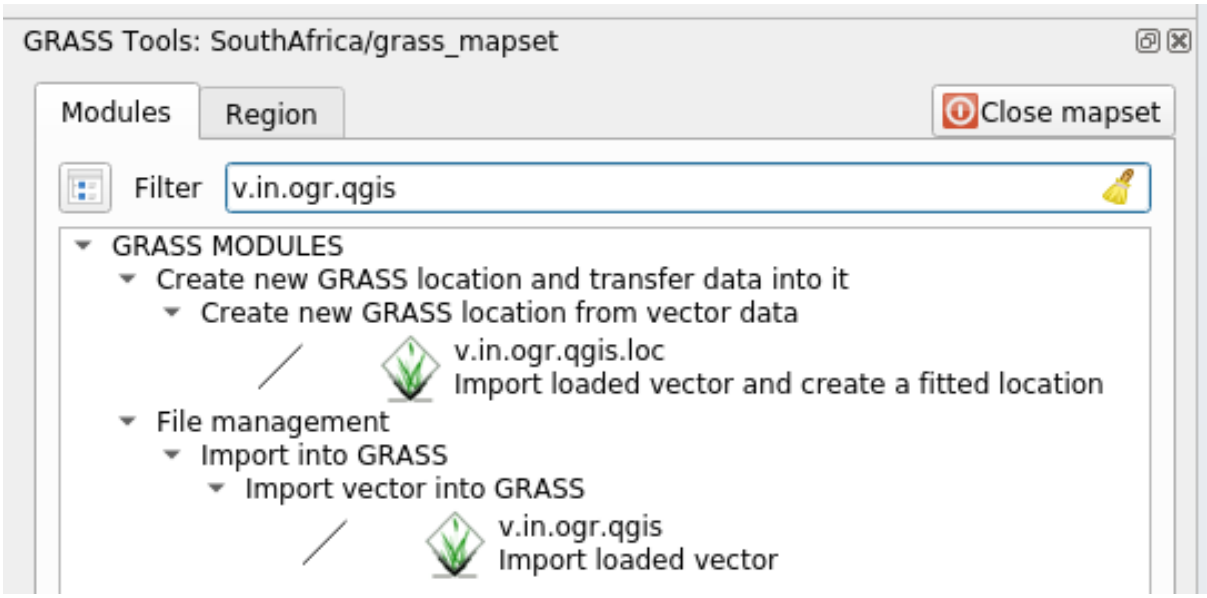
incredibly your workflow!

Follow Along: Load data using the GRASS Panel

We will use now the *long* method to load the `rivers.shp` layer into the same Mapset.

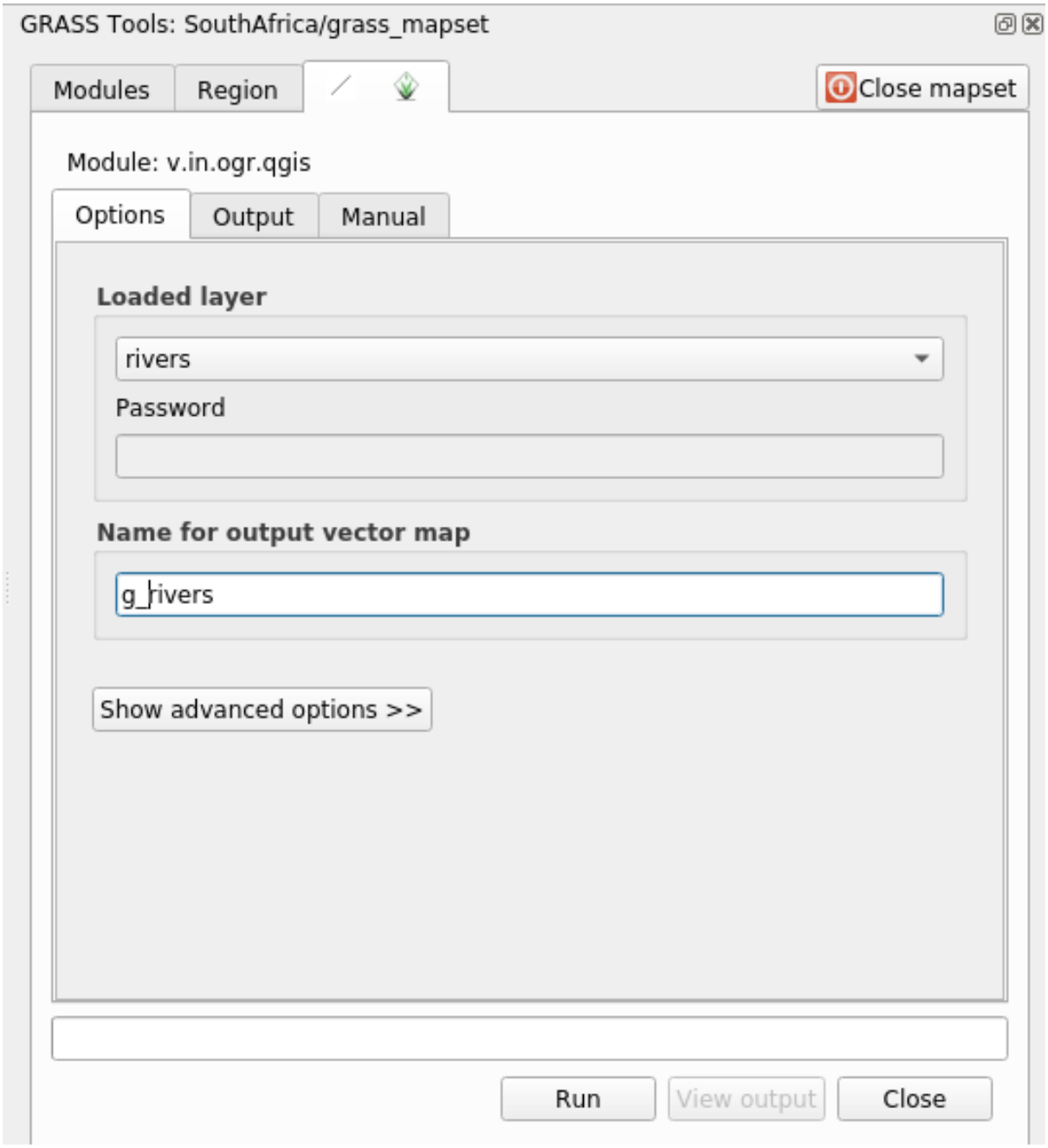
1. Load data into QGIS as usual. Use the `rivers.shp` dataset (found in the `exercise_data/shapefile/` folder)
2. As soon as it is loaded, click on the *Filter* box of the `GRASS Panel` and find the vector import tool by entering the term `v.in.ogr.qgis`:

> **Предупреждение:** There are 2 similar tools: `v.in.ogr.qgis` and `v.in.ogr.qgis.loc`. We are looking for the **first** one.



The `v` stands for *vector*, `in` means 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.

3. Once you've found this tool, click on it to bring up the tool itself. Choose the *rivers* layer in the *Loaded Layer* box and type and name it `g_rivers` to prevent confusion:



Примечание: 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.

4. Click *Run* to begin the import.
5. When it's done, click the *View output* button to see the newly imported GRASS layer in the map.
6. Close first the import tool (click the *Close* button to the immediate right of *View output*), then close the *GRASS Tools* window.
7. Remove the original *rivers* layer.

Now you are left with only the imported GRASS layer as displayed in your QGIS map.

12.1.4  Follow Along: Loading Raster Data into GRASS

You can import a raster layer in the same ways we imported vector layers.

We are going to import in the GRASS Mapset the layer `srtm_41_19_4326.tif`.

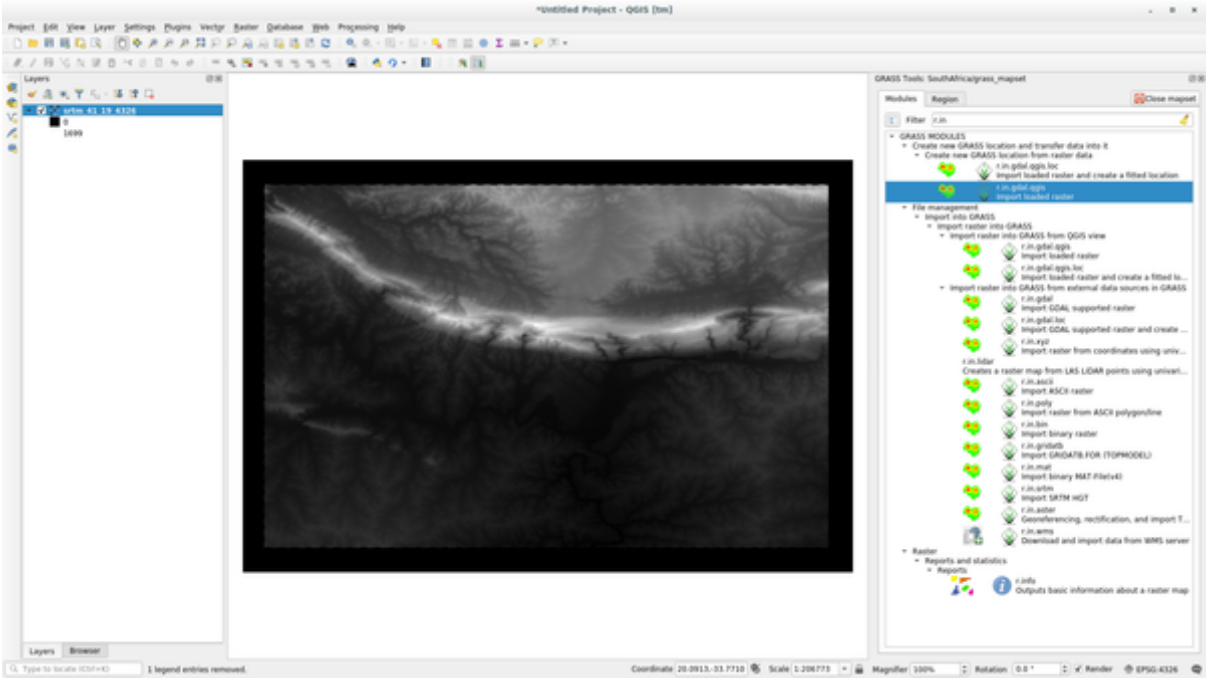
Примечание: the raster layer is already in the correct CRS, `WGS 84`. If you have layers in different CRS you must reproject them in the same CRS of the GRASS Mapset

---

1. Load the `srtm_41_19_4326.tif` layer in QGIS
2. Open the *GRASS Tools* dialog again.
3. Click on the *Modules List* tab.
4. Search for `r.in.gdal.qgis` and double click the tool to open the tool's dialog.
5. Set it up so that the input layer is `srtm_41_19_4326.tif` and the output is `g_dem`.



6. Click *Run*.
7. When the process is done, click *View output*.
8. *Close* the current tab, and then *Close* the whole dialog.



9. You may now remove the original `srtm_41_19_4326.tif` layer.

12.1.5 Try Yourself Add Layers to Mapset

Try to import in the GRASS Mapset the vector layers `water.shp` and `places.shp` from the `exercise_data/shapefile/` folder. As we did for `rivers` rename the imported layer as `g_water` and `g_places` to avoid confusion

---

Answer

You can add layers (both vector and raster) into a GRASS Mapset by drag and drop them in the Browser (see *Follow Along: Load data using the QGIS Browser*) or by using the `v.in.gdal.qgis` for vector and `r.in.gdal.qgis` for raster layers.

12.1.6 Open an existing GRASS Mapset

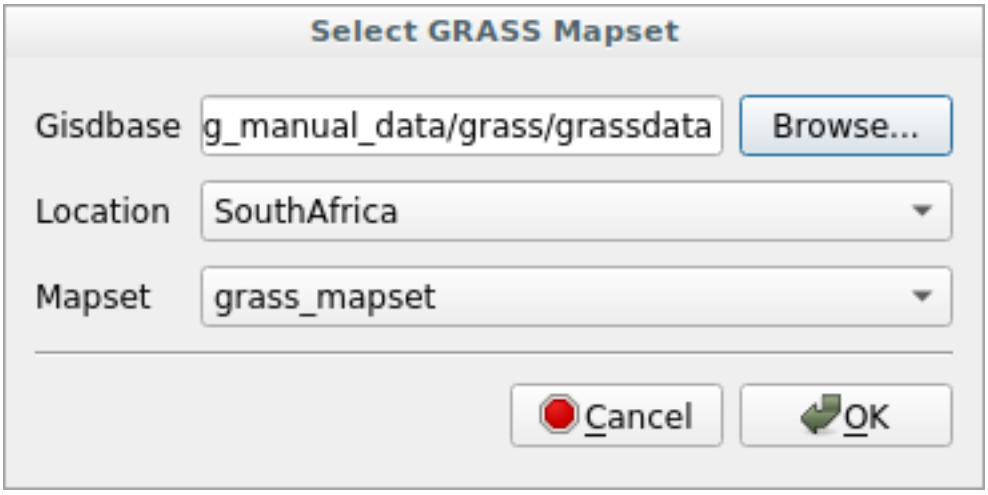
If you have an existing GRASS Mapset you can easily reopen it in another session of QGIS.

You have several method to open a GRASS Mapset, let's explore some of them.

Let's close the Mapset by clicking on the *Close Mapset* button of the *GRASS Tools* window.

Follow Along: Using the GRASS plugin

1. Click on the *Plugins –> GRASS –> Open Mapset* menu next to the *Plugins –> GRASS –> New Mapset* menu that we saw in the previous section.
2. Browse to the GRASS database folder: be careful! You must choose the parent folder, not the GRASS Mapset one. Indeed GRASS will read all the `Locations` of the database and all the `Mapsets` of each `Location`:



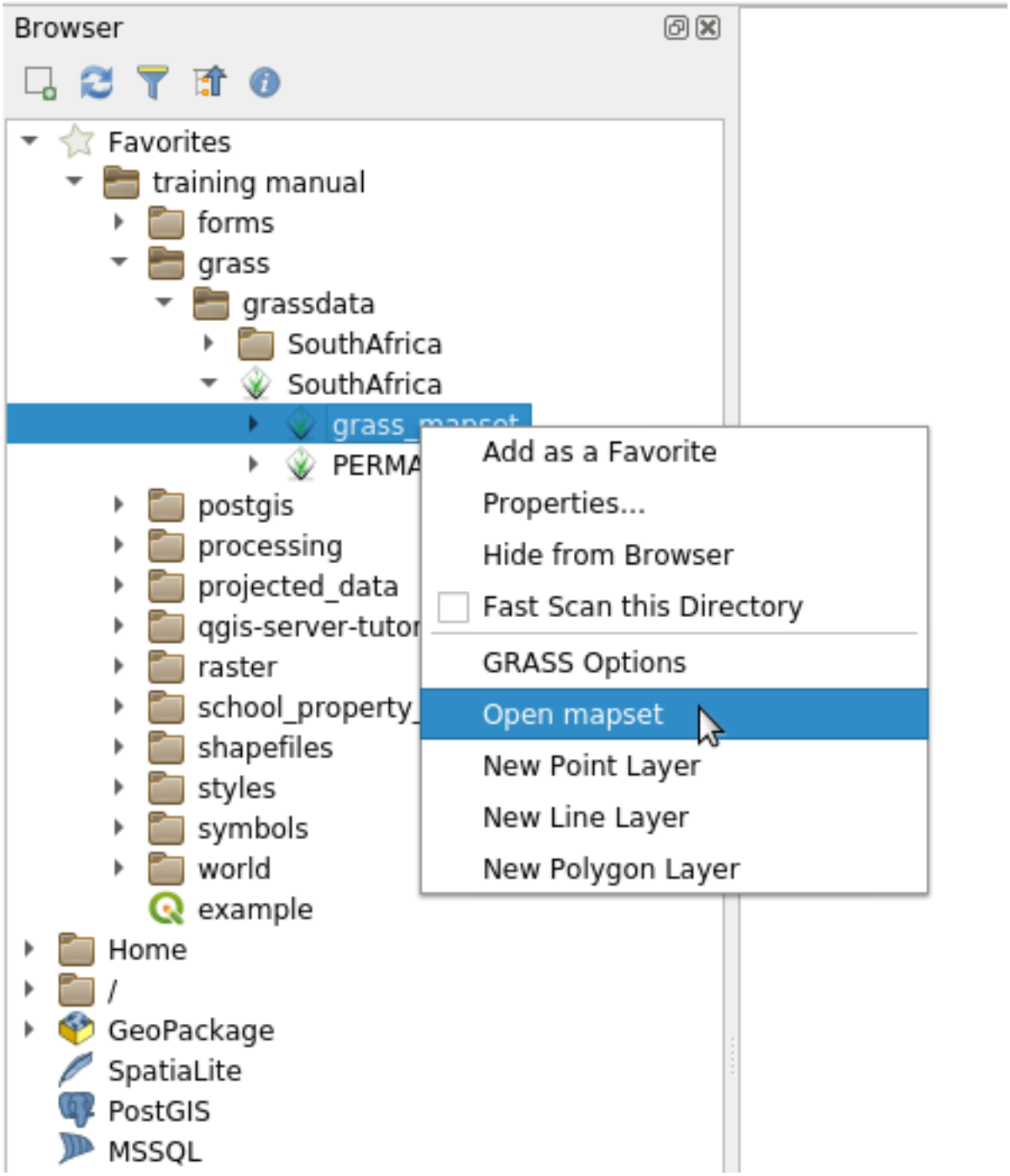
3. Choose the `Location` *SouthAfrica* and the `Mapset` *grass_mapset* that we have created before.

That's it! The GRASS Panel will become active meaning that the Mapset has been correctly opened.

Follow Along: Using the QGIS Browser

Even faster and easier is opening a `Mapset` using the QGIS Browser:

1. Close the Mapset (if it is open) by clicking on the *Close Mapset* button of the *GRASS Tools* window.
2. In the QGIS Browser, browse to the folder of the GRASS database.
3. Right click on the Mapset (remember, the Mapset has the GRASS icon next to it). You will see some options.
4. Click on *Open mapset*:



The Mapset is now open and ready to use!

Совет: Right click on a GRASS Mapset offers you a lot of different settings. Try to explore them and see all the useful options.

12.1.7 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.

12.1.8 What's Next?

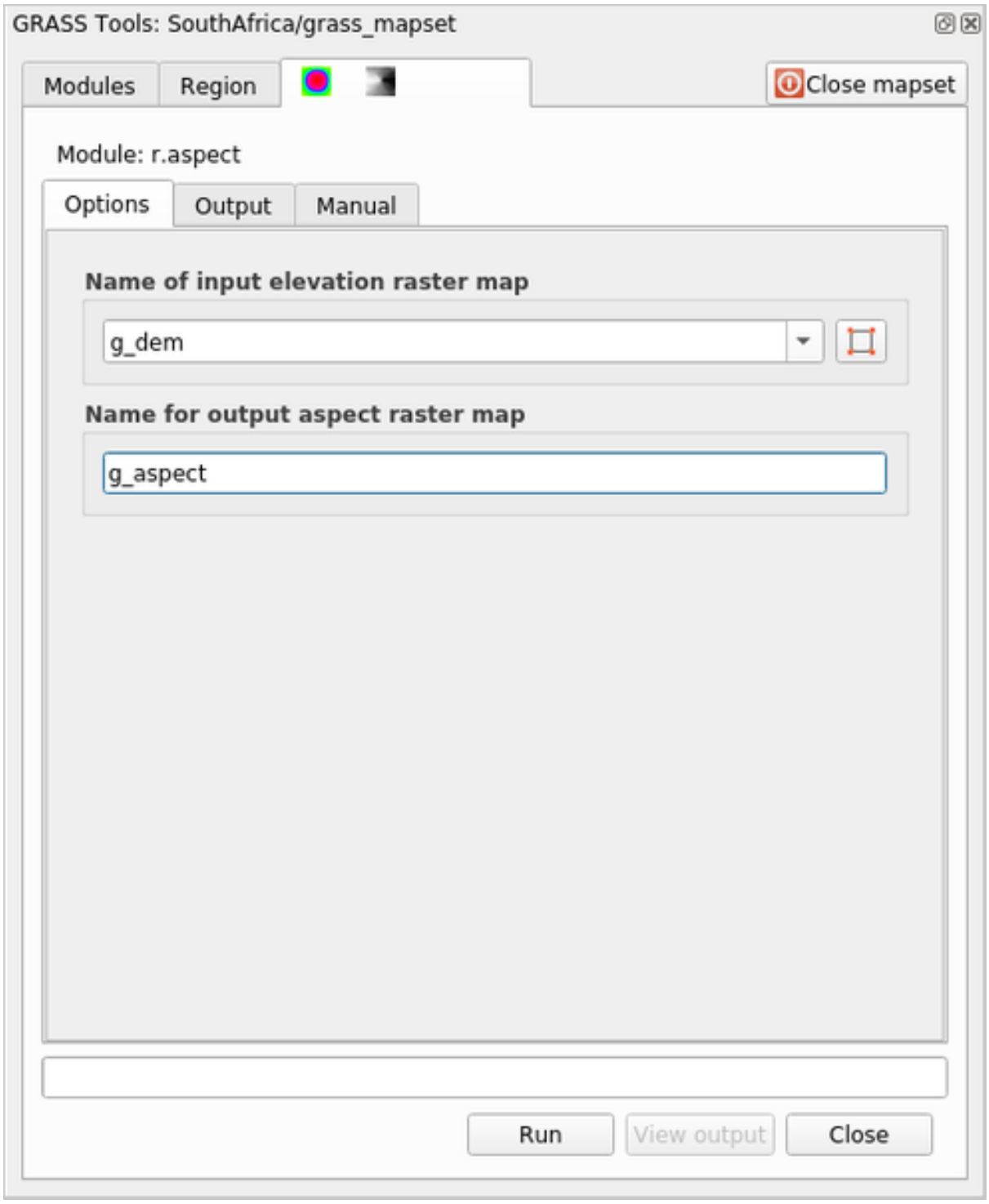
Now that the data is imported into GRASS, we can look at the advanced analysis operations that GRASS offers.

12.2 Lesson: GRASS Tools

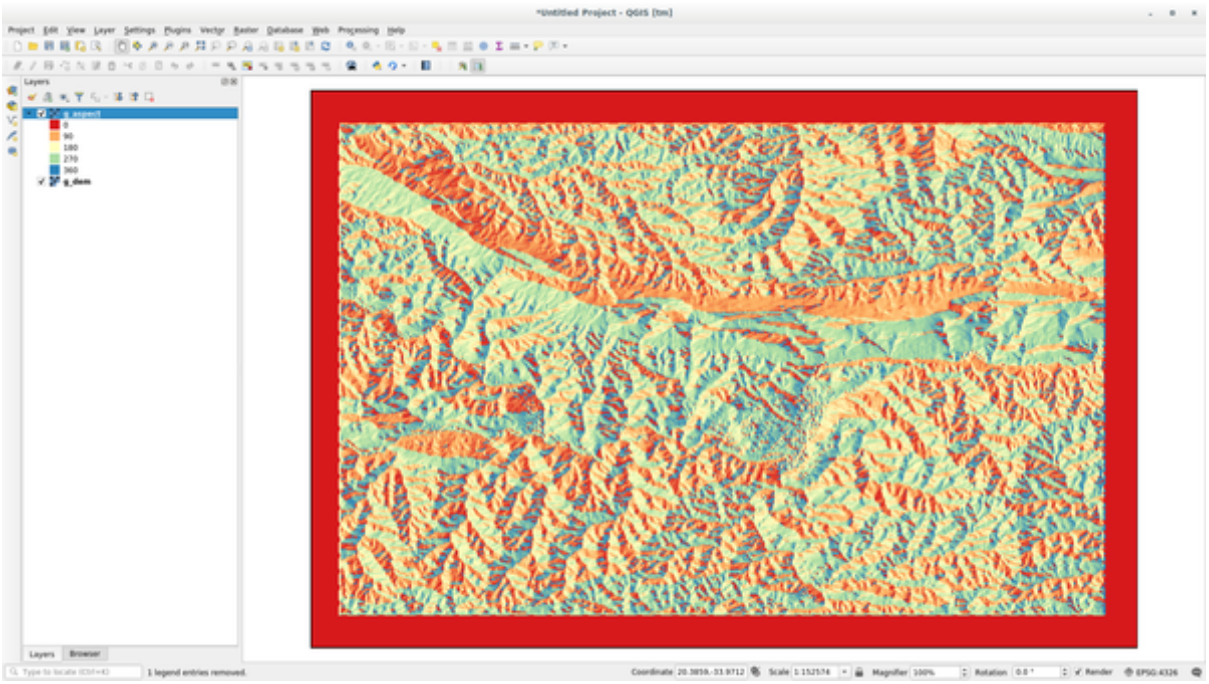
In this lesson we will present a selection of tools to give you an idea of the capabilities of GRASS.

12.2.1  Follow Along: Create an aspect map

1. Open the *GRASS Tools* tab
2. Load the `g_dem` raster layer from the *grass_mapset* Mapset
3. Look for the *r.aspect* module by searching for it in the *Filter* field of the *Modules List* tab
4. Open the tool and set it up like this and click on the *Run* button:



5. When the process is finished click on *View Output* to load the resulting layer in the canvas:

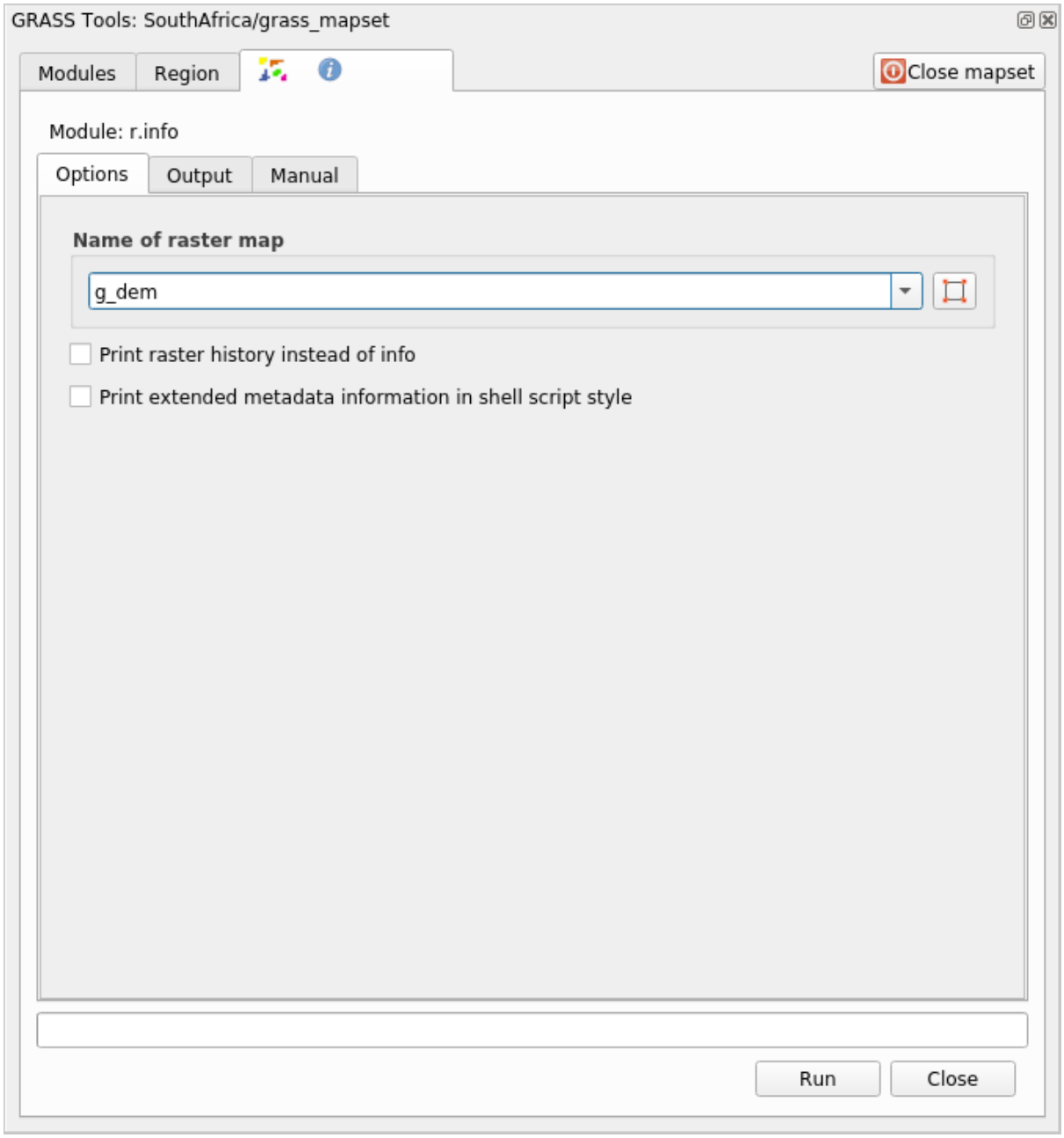


The `g_aspect` layer is stored within the *grass_mapset* Mapset so you can remove the layer from the canvas and reload it whenever you want.

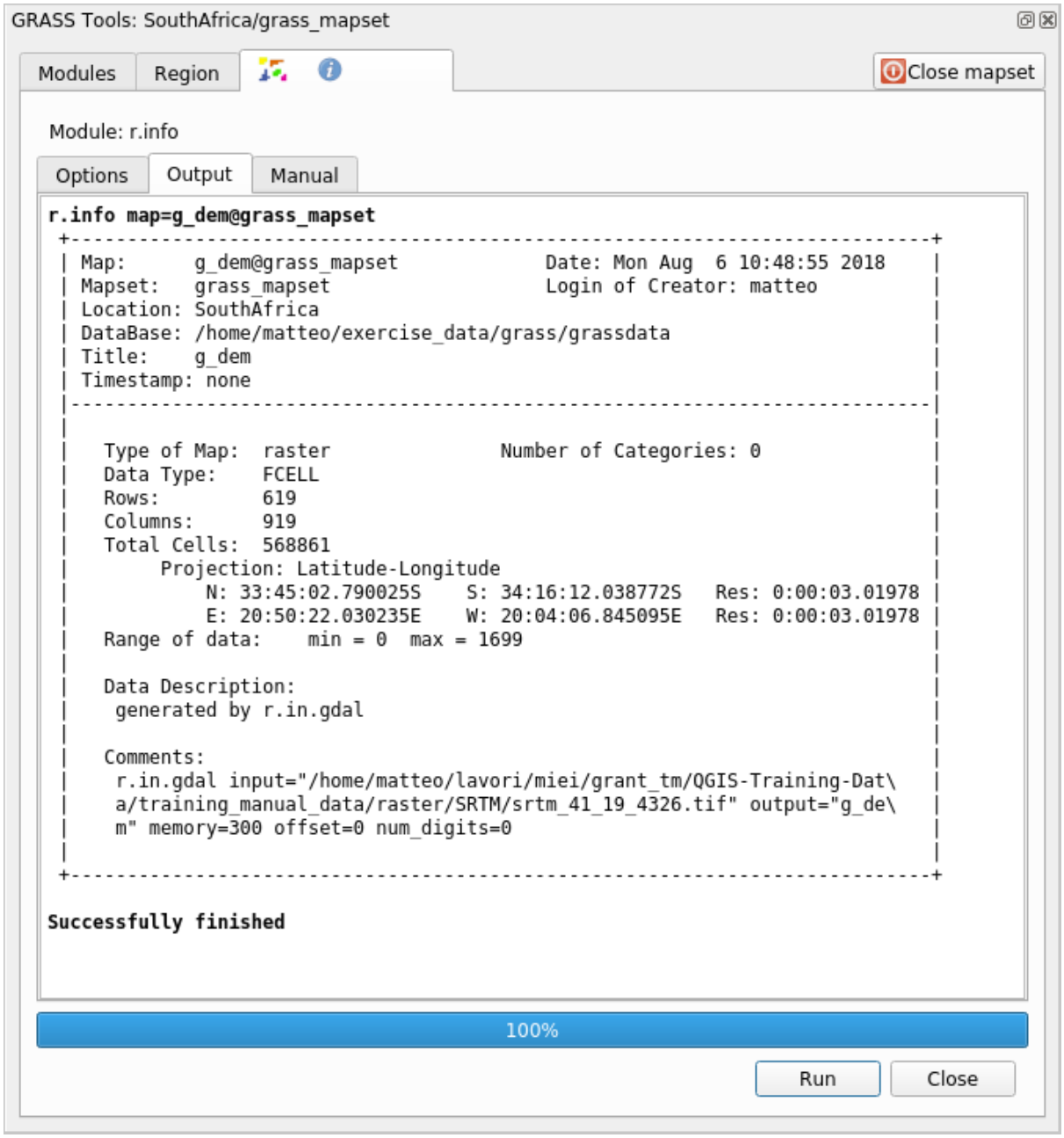
12.2.2  Follow Along: Get basic statistic of raster layer

We want to know some basic statistics of the `g_dem` raster layer.

1. Open the *GRASS Tools* tab
2. Load the `g_dem` raster layer from the *grass_mapset* Mapset
3. Look for the *r.info* module by searching for it in the *Filter* field of the *Modules List* tab
4. Set up the tool like this and click on *Run*:



5. Within the Output tab you will see some raster information printed, like the path of the file, the number of rows and columns and other useful information:



12.2.3  Follow Along: The Reclass Tool

Reclassifying a raster layer is a very useful task. We just created the `g_aspect` layer from the `g_dem` one. The value range gets from 0 (North) passing through 90 (East), 180 (South), 270 (West) and finally to 360 (North again). We can reclassify the `g_aspect` layer to have just 4 **categories** following specific *rules* (North = 1, East = 2, South = 3 and West = 4).

Grass reclassify tool accepts a `txt` file with the defined rules. Writing the rules is very simple and the GRASS Manual contains very good description.

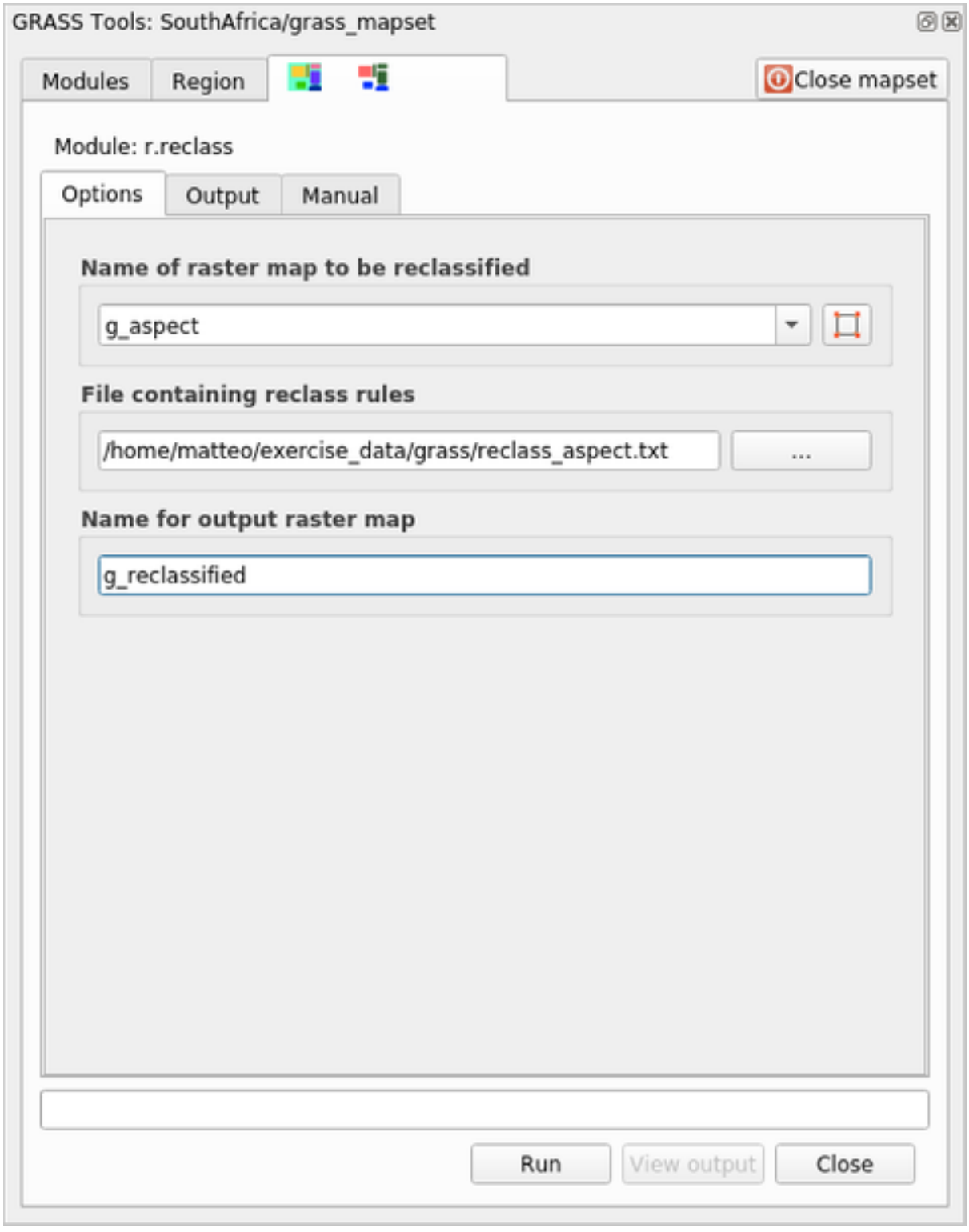
Совет: Each GRASS tool has its own Manual tab. Take the time to read the description of the tool you are using to don't miss some useful parameters

---

1. Load the `g_aspect` layer or, if you don't have create it, go back to the *Follow Along: Create an aspect map*

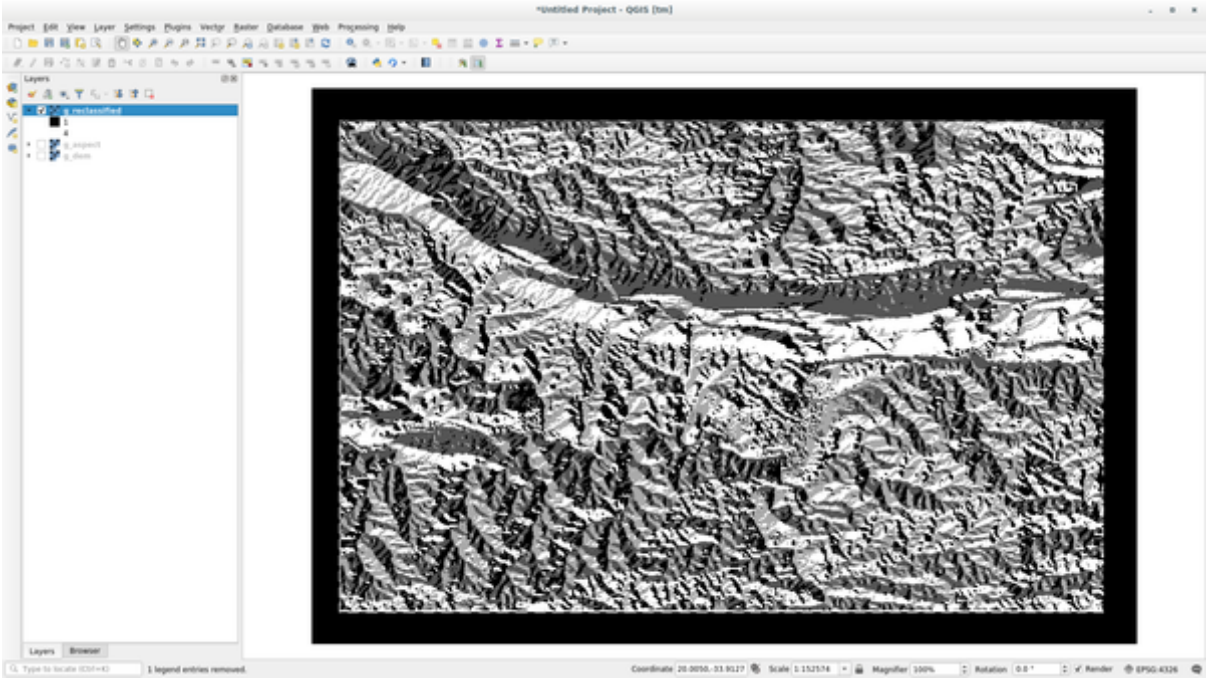
section.

2. Look for the *r.reclass* module by searching for it in the *Filter* field of the *Modules List* tab
3. Open the tool and set it up like the following picture. The file containing the rules is in the `exercise_data/grass/` folder, named `reclass_aspect.txt`.
4. Click on *Run* and wait until the process is finished:



5. Click on *View Output* to load the reclassified raster in the canvas

The new layer is made up by just 4 values (1, 2, 3, and 4) and it is easier to manage and to process.



Совет: Open the `reclass_aspect.txt` with a text editor to see the rules and to start becoming used to them. Moreover, take a deep look at the GRASS manual: a lot of different examples are pointed out.

12.2.4 Try Yourself Reclassify with your rules

Try to reclassify the `g_dem` layer into 3 new categories:

- from 0 to 1000, new value = 1
- from 1000 to 1400, new value = 2
- from 1400 to the maximum raster value, new value = 3

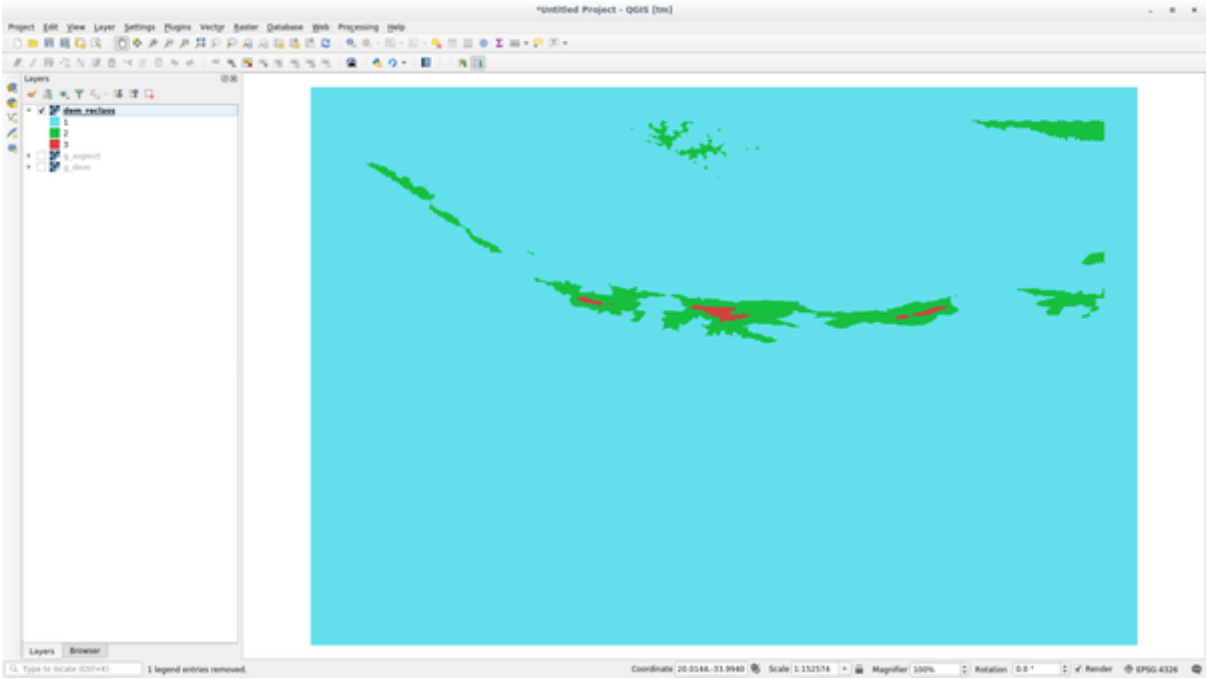
Answer

To discover the maximum value of the raster run the `r.info` tool: in the console you will see that the maximum value is 1699. You are now ready to write the rules.

1. Open a text editor and add the following rules:

```
0 thru 1000 = 1
1000 thru 1400 = 2
1400 thru 1699 = 3
```

2. Save the file as `my_rules.txt` file and close the text editor.
3. Run the `r.reclass` tool: choose the `g_dem` layer and load the file containing the rules you just have saved.
4. Click on *Run* and then on *View Output*. You can change the colors and the final result should look like the following picture:



12.2.5 Follow Along: The Mapcalc Tool

The Mapcalc tools is similar to the Raster Calculator of QGIS. You can perform mathematical operation on one or more raster layers and the final result will be a new layer with the calculated values.

The aim of the next exercise is to extract the values greater than 1000 from the `g_dem` raster layer.

1. Look for the *r.mapcalc* module by searching for it in the *Filter* field of the *Modules List* tab.
2. Start the tool.

   The *Mapcalc* dialog allows you to construct a sequence of analyses to be performed on a raster, or collection of rasters. You will use these tools to do so:

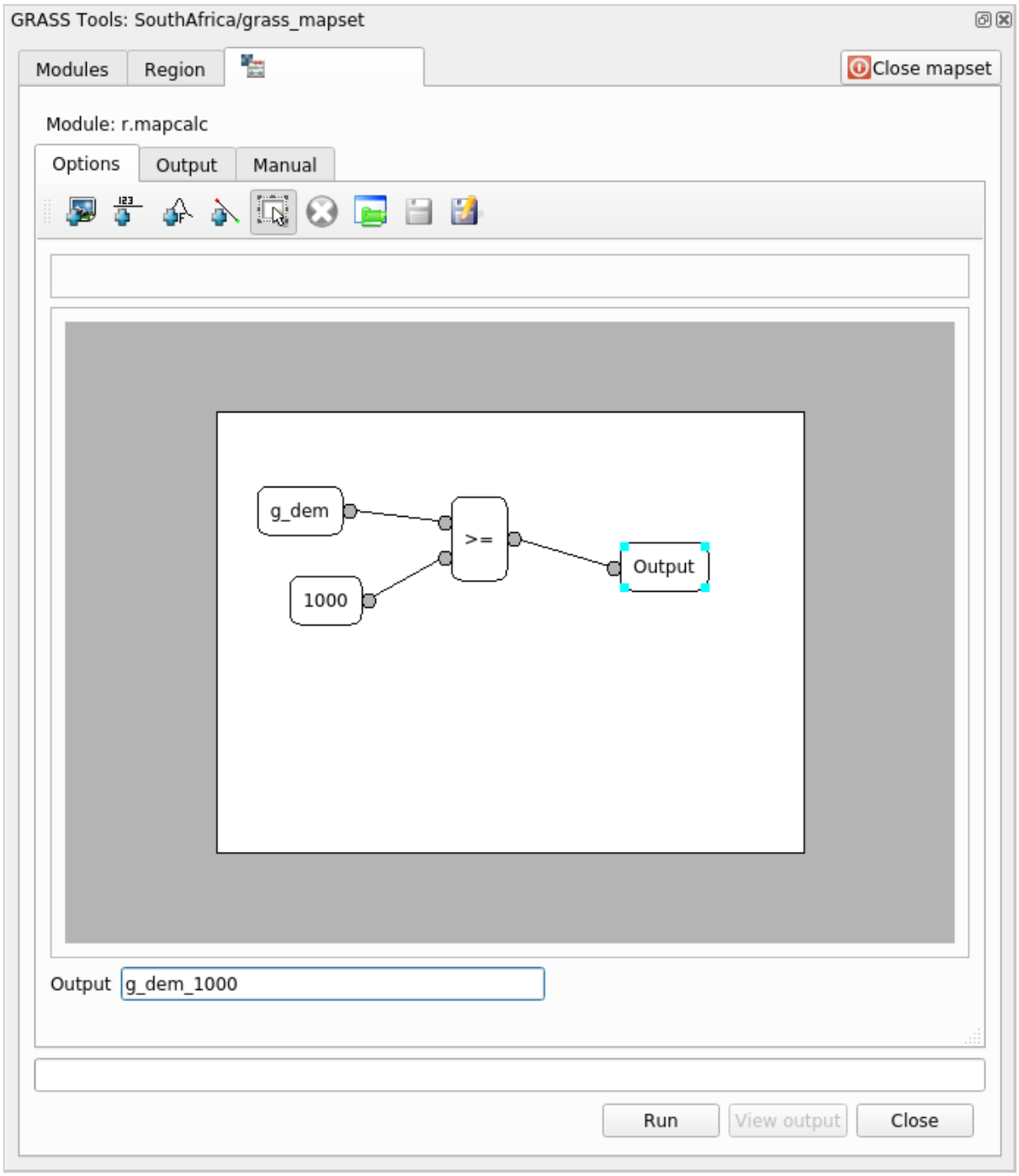


   In order, they are:

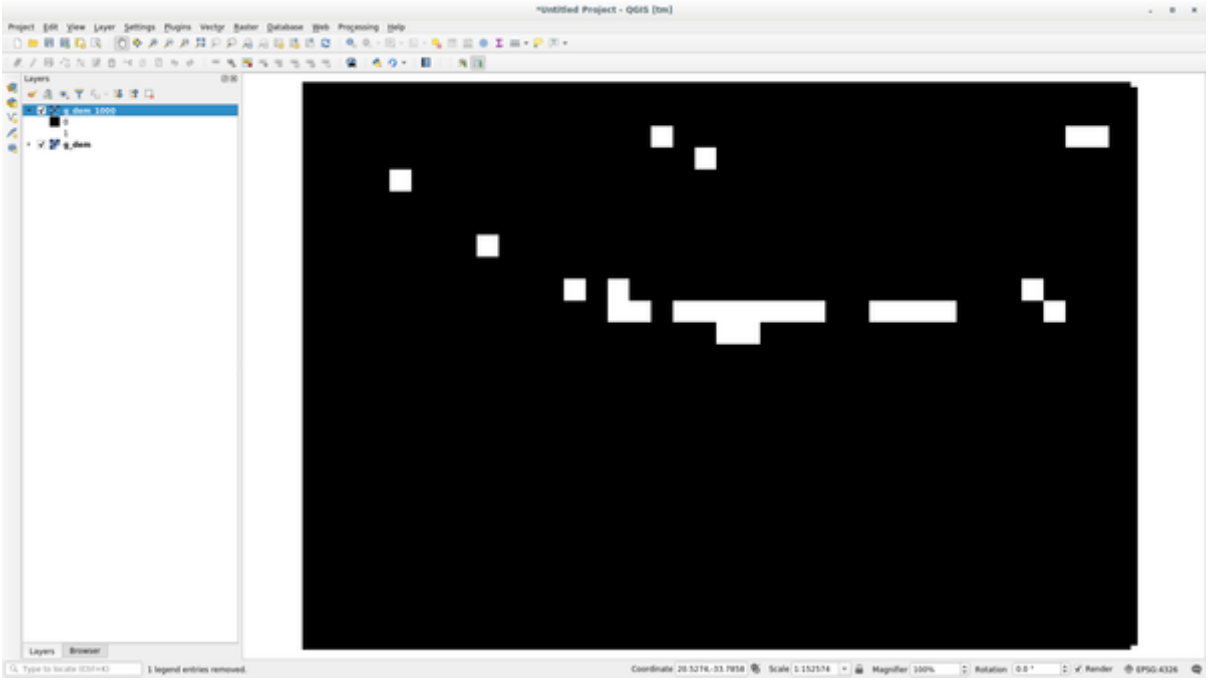
   - *Add map*: Add a raster file from your current GRASS mapset.
   - *Add constant value*: Add a constant value to be used in functions, 1000 in this case
   - *Add operator or function*: Add an operator or function to be connected to inputs and outputs, we will use the operator `greater equals than`
   - *Add connection*: Connect elements. Using this tool, click and drag from the red dot on one item to the red dot on another item. Dots that are correctly connected to a connector line will turn gray. If the line or dot is red, it is not properly connected!
   - *Select item*: Select an item and move selected items.
   - *Delete selected item*: Removes the selected item from the current mapcalc sheet, but not from the mapset (if it is an existing raster)

- *Open*: Open an existing file with the operation defined
- *Save*: Save all the operation in a file
- *Save as*: Save all the operations as a new file on the disk.

3. Using these tools, construct the following algorithm:



4. Click on *Run* and then on *View output* to see the output displayed in your map:



This shows all the areas where the terrain is higher than 1000 meters.

Совет: You can also save the formula you have created and load it in another QGIS project by clicking on the last button on the GRASS Mapcalc toolbar.

12.2.6 In Conclusion

In this lesson, we have covered only a few of the many tools GRASS offers. To explore the capabilities of GRASS for yourself, open the *GRASS Tools* dialog and scroll down the *Modules List*. Or for a more structured approach, look under the *Modules Tree* tab, which organizes tools by type.

ГЛАВА 13

Module: Assessment

Use your own data for this section. You will need:

- a point vector dataset of points of interest, with point names and multiple categories
- a line vector dataset of roads
- a polygon vector dataset of land use (using property boundaries)
- a visual-spectrum image (such as an aerial photograph)
- a DEM (downloadable from the CGIAR-CSI if you don't have your own)

13.1 Create a base map

Before doing any data analysis, you will need a base map, which will provide your analysis result with context.

13.1.1 Add the point layer

- Add in the point layer. Based on the level that you're doing the course at, do only what is listed in the appropriate section below:



- Label the points according to a unique attribute, such as place names. Use a small font and keep the labels inconspicuous. The information should be available, but shouldn't be a main feature of the map.
- Classify the points themselves into different colors based on a category. For example, categories could include «tourist destination», «police station», and «town center».



- Do the same as the  section.
- Classify the point size by importance: the more significant a feature, the larger its point. However, don't exceed the point size of `2.00`.
- For features that aren't located at a single point (for example, provincial/regional names, or town names at a large scale), don't assign any point at all.



- Don't use point symbols to symbolize the layer at all. Instead, use labels centered over the points; the point symbols themselves should have no size.
- Use *Data defined settings* to style the labels into meaningful categories.
- Add appropriate columns to the attribute data if necessary. When doing so, don't create fictional data - rather, use the *Field Calculator* to populate the new columns, based on appropriate existing values in the dataset.

13.1.2 Add the line layer

- Add the road layer and then change its symbology. Don't label the roads.



- Change the road symbology to a light color with a broad line. Make it somewhat transparent as well.



- Create a symbol with multiple symbol layers. The resulting symbol should look like a real road. You can use a simple symbol for this; for example, a black line with a thin white solid line running down the center. It can be more elaborate as well, but the resulting map should not look too busy.
- If your dataset has a high density of roads at the scale you want to show the map at, you should have two road layers: the elaborate road-like symbol, and a simpler symbol at smaller scales. (Use scale-based visibility to make them switch out at appropriate scales.)
- All symbols should have multiple symbol layers. Use symbols to make them display correctly.



- Do the same as in the  section above.
- In addition, roads should be classified. When using realistic road-like symbols, each type of road should have an appropriate symbol; for example, a highway should appear to have two lanes in either direction.

13.1.3 Add the polygon layer

- Add the land use layer and change its symbology.



- Classify the layer according to land use. Use solid colors.



- Classify the layer according to land use. Where appropriate, incorporate symbol layers, different symbol types, etc. Keep the results looking subdued and uniform, however. Keep in mind that this will be part of a backdrop!



- Use rule-based classification to classify the land use into general categories, such as «urban», «rural», «nature reserve», etc.

13.1.4 Create the raster backdrop

- Create a hillshade from the DEM, and use it as an overlay for a classified version of the DEM itself. You could also use the *Relief* plugin (as shown in the lesson on plugins).

13.1.5 Finalize the base map

- Using the resources above, create a base map using some or all of the layers. This map should include all the basic information needed to orient the user, as well as being visually unified / «simple».

13.2 Analyze the data

- You are looking for a property that satisfies certain criteria.
- You can decide on your own criteria, which you must document.
- There are some guidelines for these criteria:
  - the target property should be of (a) certain type(s) of land use
  - it should be within a given distance from roads, or be crossed by a road
  - it should be within a given distance from some category of points, like a hospital for example

13.2.1  / 

- Include raster analysis in your results. Consider at least one derived property of the raster, such as its aspect or slope.

13.3 Final Map

- Use the *Print Layout* to create a final map, which incorporates your analysis results.
- Include this map in a document along with your documented criteria. If the map has become too visually busy due to the added layer(s), deselect the layers which you feel are the least necessary.
- Your map must include a title and a legend.

ГЛАВА 14

Module: Forestry Application

In modules 1 through 13, you have already learned quite a lot about QGIS and how to work with it. If you are interested in learning about some basic forestry applications of GIS, following this module will give you the ability to apply what you have learned and will show you some new useful tools.



The development of this module has been sponsored by the European Union.

14.1 Lesson: Forestry Module Presentation

Following this module about a forestry application requires the knowledge you have learned through the modules 1 to 11 of this training manual. The exercises in the following lessons assume you are already capable of doing many of the basic operations in QGIS and only tools that have not been used before are presented in more detail.

Nevertheless, the module follows a basic level throughout the lessons so that if you have previous experience with QGIS, you can probably follow the instructions without problems.

Note that you need to download an additional data package for this module.

14.1.1 Forestry Sample Data

Примечание: The sample data used in this module is part of the training manual data set and is available in the `exercise_data\forestry\` folder.

The forestry related sample data (forestry map, forest data), has been provided by the EVO-HAMK forestry school. The datasets have been modified to adapt to the lessons needs.

The general sample data (aerial images, LiDAR data, basic maps) has been obtained from the National Land Survey of Finland open data service, and adapted for the purposes of the exercises. The open data file download service can be accessed in English here.

Предупреждение: As for the rest of the training manual, this module includes instructions on adding, deleting and altering GIS datasets. We have provided training datasets for this purpose. Before using the techniques described here on your own data, always ensure you have proper backups!

14.2 Lesson: Georeferencing a Map

A common forestry task would be the update of the information for a forestry area. It is possible that the previous information for that area dates several years back and was collected analogically (that is, in paper) or perhaps it was digitized but all you have left is the paper version of that inventory data.

Most likely you would like to use that information in your GIS to, for example, compare later with later inventories. This means that you will need to digitize the information at hand using your GIS software. But before you can start the digitizing, there is an important first step to be done, scanning and georeferencing your paper map.

The goal for this lesson: To learn to use the Georeferencer tool in QGIS.

14.2.1  Scan the map

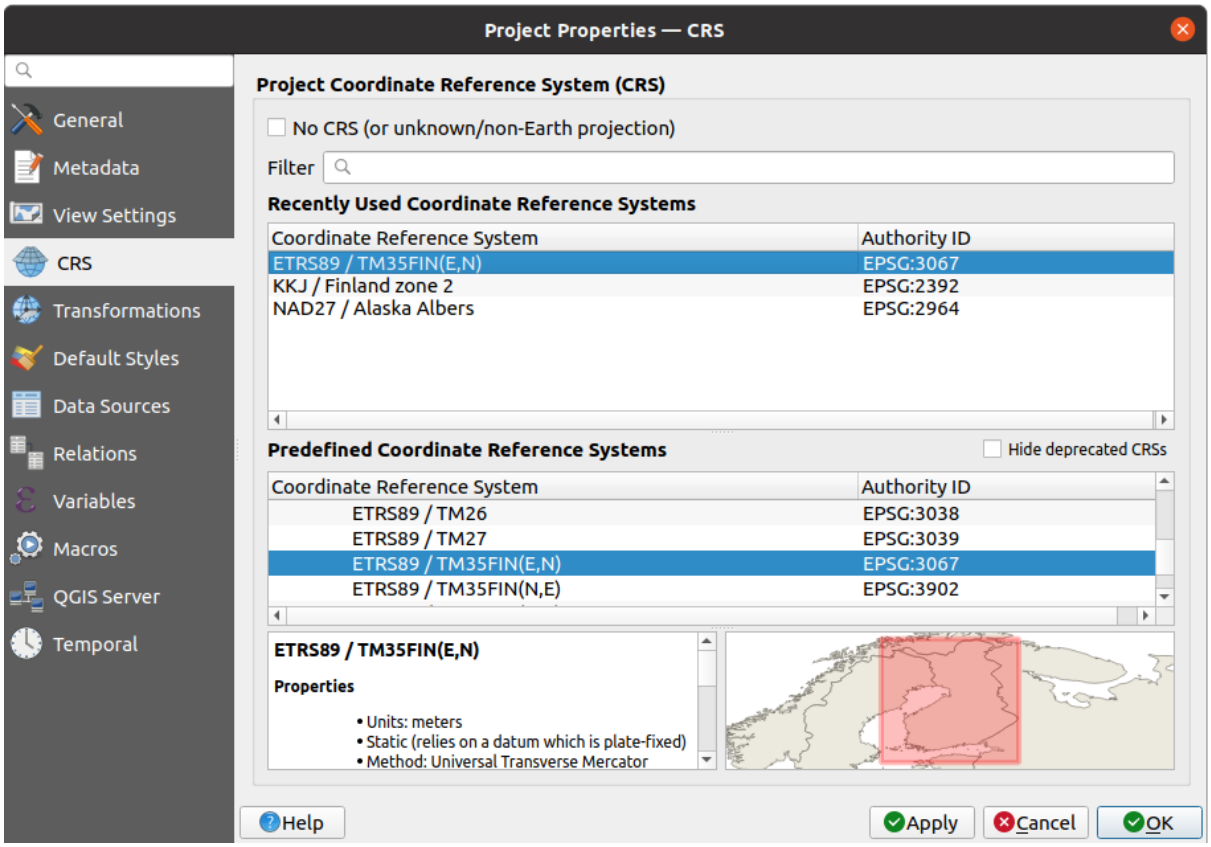
The first task you will have to do is to scan your map. If your map is too big, then you can scan it in different parts but keep in mind that you will have to repeat preprocessing and georeferencing tasks for each part. So if possible, scan the map in as few parts as possible.

If you are going to use a different map that the one provided with this manual, use your own scanner to scan the map as an image file, a resolution of 300 DPI will do. If your map has colors, scan the image in color so that you can later use those colors to separate information from your map into different layers (for ex., forest stands, contour lines, roads…).

For this exercise you will use a previously scanned map, you can find it as `rautjarvi_map.tif` in the data folder `exercise_data/forestry`

14.2.2  Follow Along: Georeferencing the scanned map

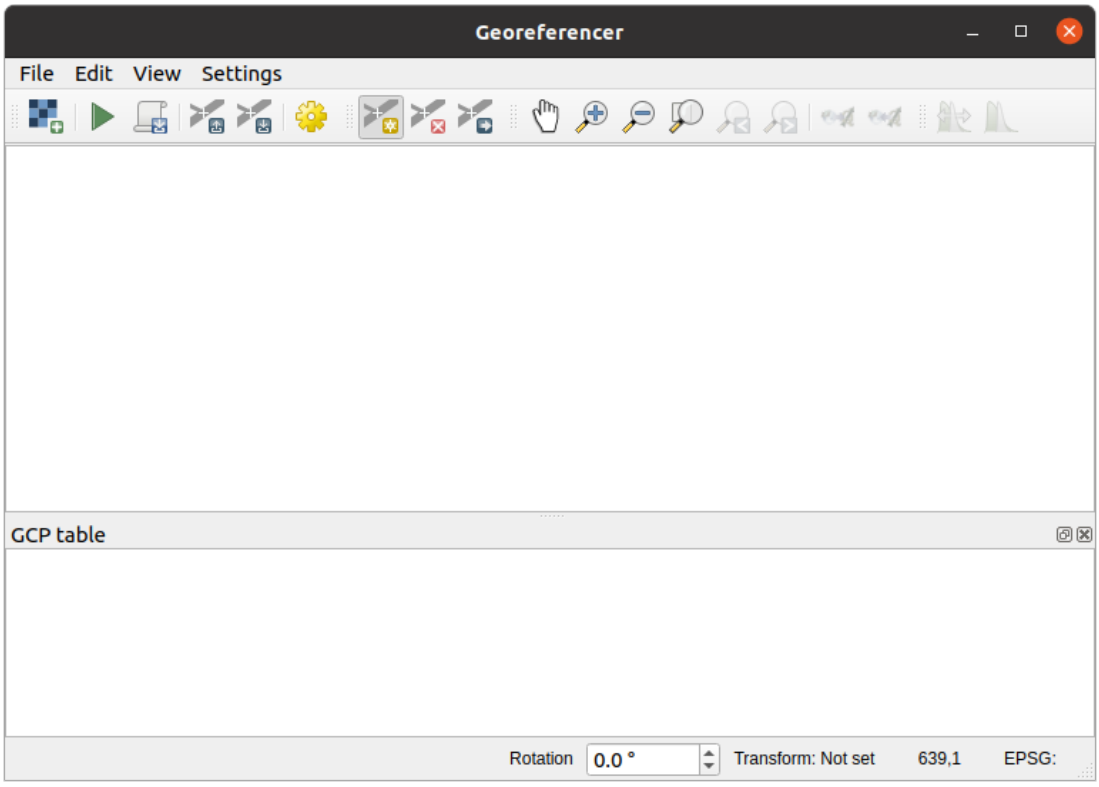
1. Open QGIS and set the project's CRS to `EPSG:3067 - ETRS89 / TM35FIN(E,N)` in *Project* ► *Properties* ► *CRS*, which is the currently used CRS in Finland.



2. Save the QGIS project as `map_digitizing.qgs`.

You will use the georeferencing tool from QGIS named *Georeferencer*. To georeference the map:

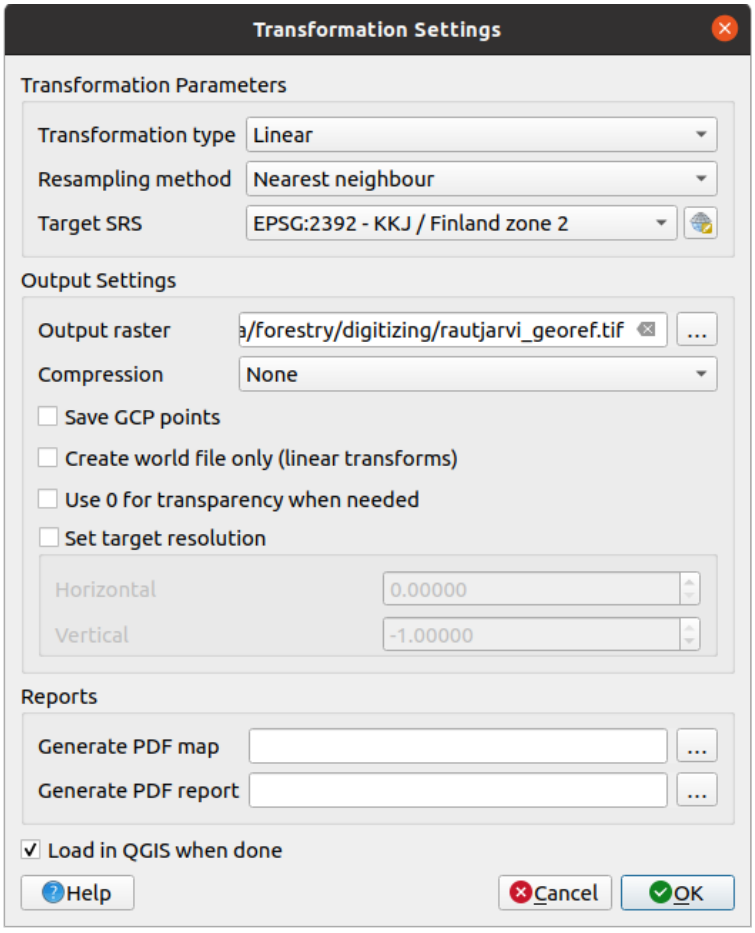
1. Open the georeference tool, *Raster* ► *Georeferencer*....



2. Add the map image file, `rautjarvi_map.tif`, as the image to georeference, *File* ► *Open raster*.
3. Click *OK*.

Next you should define the transformation settings for georeferencing the map:

1. Open *Settings* ► *Transformation settings*.
2. Set the *Transformation type* to `Linear` and the *Resampling method* to `Nearest neighbour`.
3. Press the Select CRS button next to the *Target SRS* option and select the `EPSG:2392 - KKJ / Finland zone 2` CRS; it is the CRS that was used in Finland back in 1994 when this map was created.
4. Click the icon next to the *Output raster* box, go to the folder and create the folder `exercise_data\forestry\digitizing` and name the file as `rautjarvi_georef.tif`.
5. Check *Load in QGIS when done*
6. Leave the rest of parameters as default.

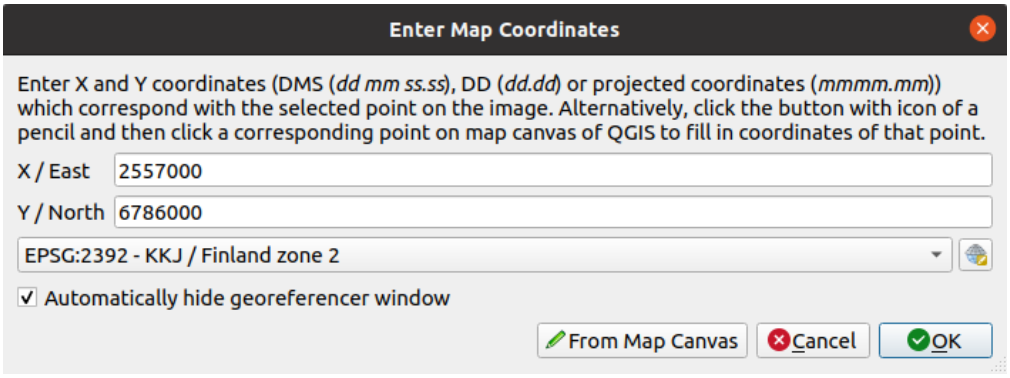


7. Click *OK*.

The map contains several cross-hairs marking the coordinates in the map, we will use those to georeference this image. You can use the zooming and panning tools as you usually do in QGIS to inspect the image in the Georeferencer's window.

1. Zoom in to the left lower corner of the map and note that there is a cross-hair with a coordinate pair, X and Y, that as mentioned before are in `KKJ / Finland zone 2` CRS. You will use this point as the first ground control point for the georeferencing your map.
2. Select the *Add point* tool and click in the intersection of the cross-hairs (pan and zoom as needed).

3. In the *Enter map coordinates* dialogue write the coordinates that appear in the map (X: 2557000 and Y: 6786000) and their CRS (`EPSG:2392 – KKJ / Finland zone 2`)



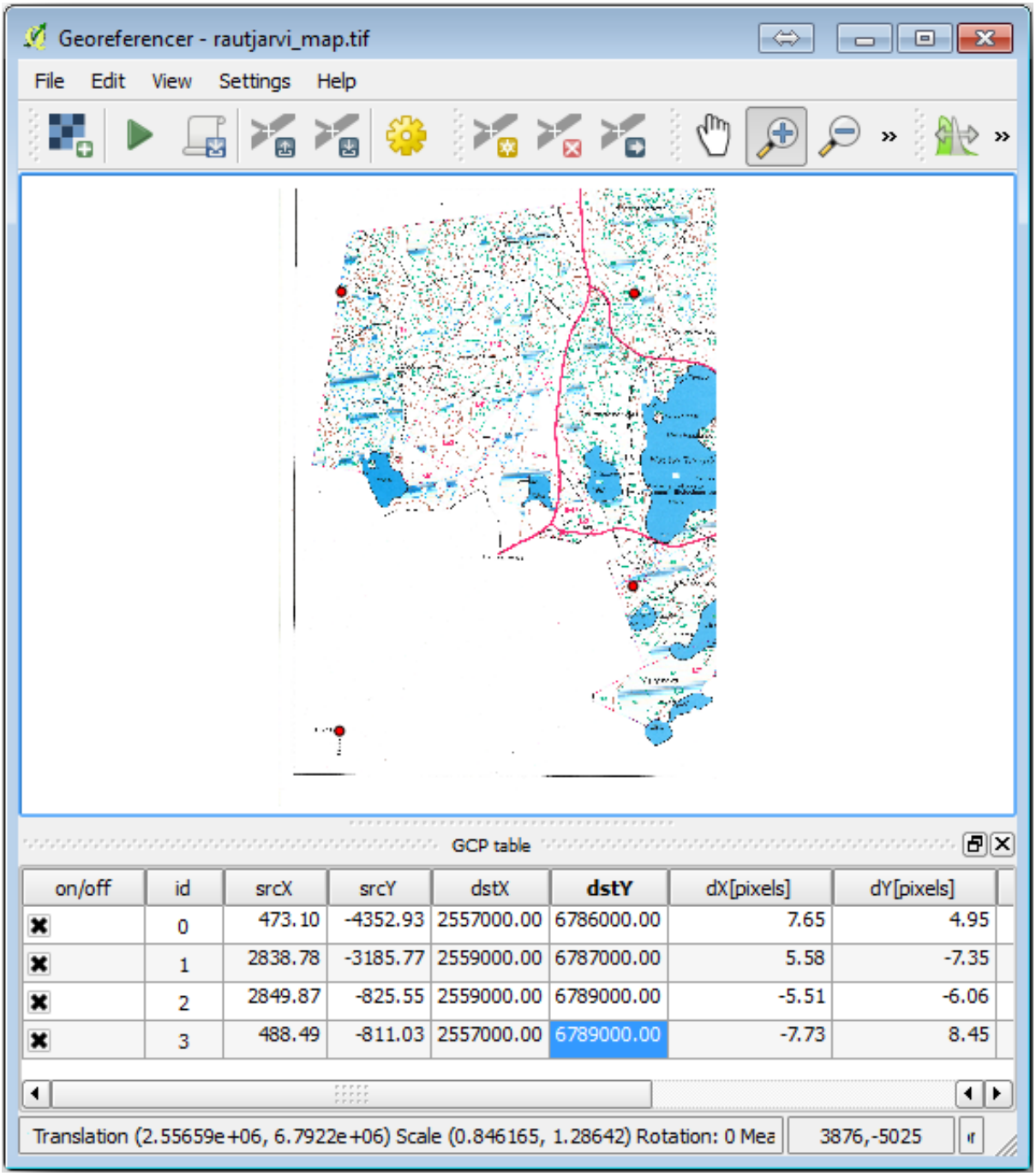
4. Click *OK*.

   The first coordinate for the georeferencing is now ready.

5. Look for other black cross-hairs in the image, they are separated 1000 meters from each other both in North and East direction. You should be able to calculate the coordinates of those points in relation to the first one.

6. Zoom out in the image and move to the right or the top until you find other cross-hair, and estimate how many kilometres you have moved. Try to get ground control points as far from each other as possible.

7. Digitize at least three more ground control points in the same way you did the first one. You should end up with something similar to this:



With already three digitized ground control points you will be able to see the georeferencing error as a red line coming out of the points. The error in pixels can be seen also in the *GCP table* in the *dX[pixels]* and *dY[pixels]* columns. The error in pixels should not be higher than 10 pixels. If it is, you should review the points you have digitized and the coordinates you have entered to find what the problem is. You can use the image above as a guide.

Once you are happy with your control points, you can save them for later use:

1. Go to *File* ► *Save GCP points as...*.
2. In the folder `exercise_data\forestry\digitizing`, name the file `rautjarvi_map.tif.points`.

Finally, georeference your map:

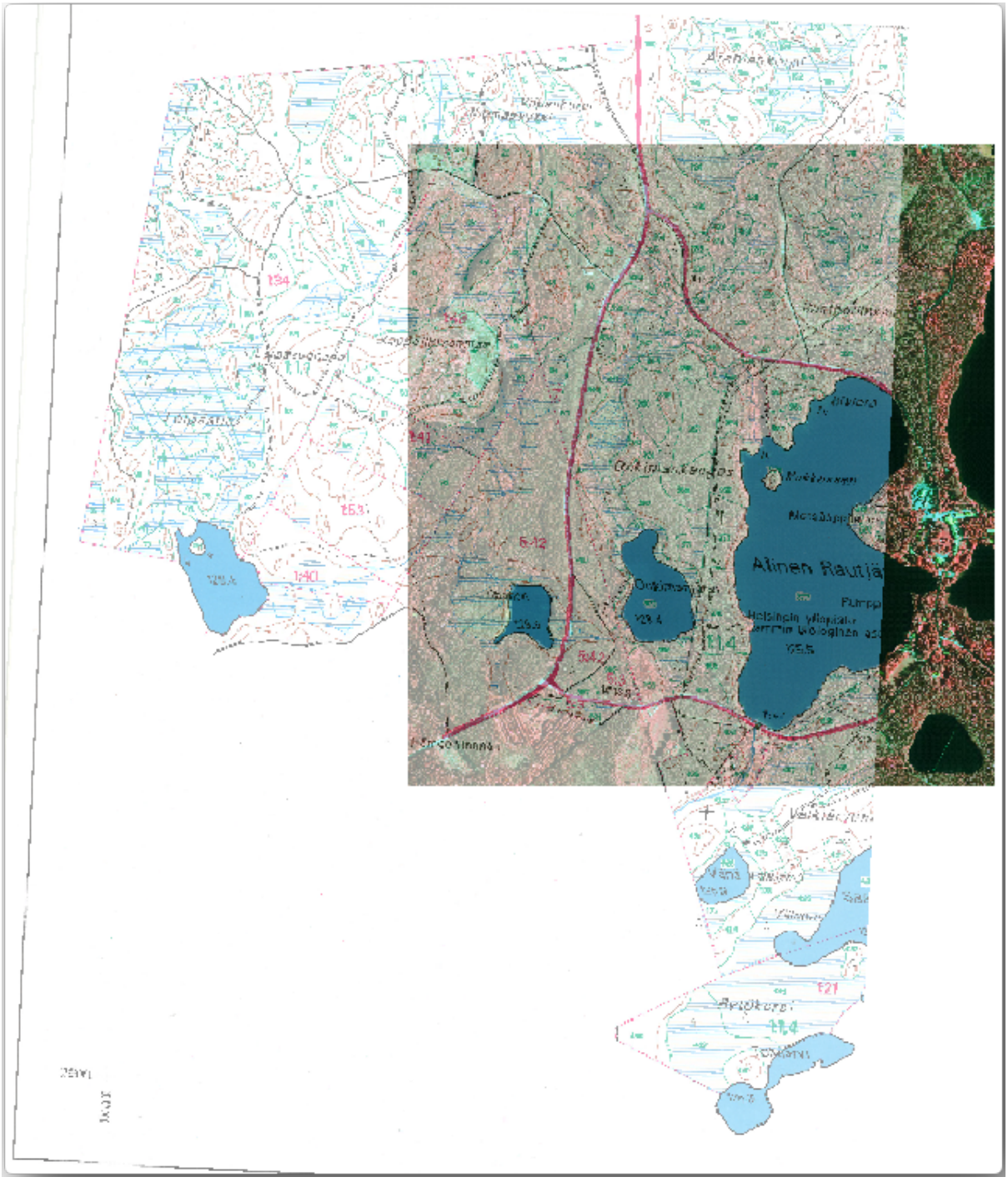
1. Go to *File* ► *Start georeferencing*.

2. Note that you named the file already as `rautjarvi_georef.tif` when you edited the Georeferencer settings.

   Now you can see the map in QGIS project as a georeferenced raster. Note that the raster seems to be slightly rotated, but that is simply because the data is in `KKJ / Finland zone 2` and your project is in `ETRS89 / TM35FIN(E,N)`.

3. To check that your data is properly georeferenced, you can

   1. Open the aerial image in the `exercise_data\forestry` folder, named `rautjarvi_aerial.tif`.
   2. Your map and this image should match quite well: set the map transparency to 50% and compare it to the aerial image.



4. Save the changes to your QGIS project, you will continue from this point for the next lesson.

14.2.3 In Conclusion

You have now georeferenced a paper map, making it possible to use it as a map layer in QGIS.

14.2.4 What's Next?

In the next lesson, you will digitize the forest stands in your map as polygons and add the inventory data to them.

14.3 Lesson: Digitizing Forest Stands

Unless you are going to use your georeferenced map as a simple background image, the next natural step is to digitize elements from it. You have already done so in the exercises about creating vector data in *Занятие: Создание нового набора векторных данных*, when you digitized the school fields. In this lesson, you are going to digitize the forest stands" borders that appear in the map as green lines but instead of doing it using an aerial image, you will use your georeferenced map.

The goal for this lesson: Learn a technique to help the digitizing task, digitizing forest stands and finally adding the inventory data to them.

14.3.1  Follow Along: Extracting the Forest Stands Borders

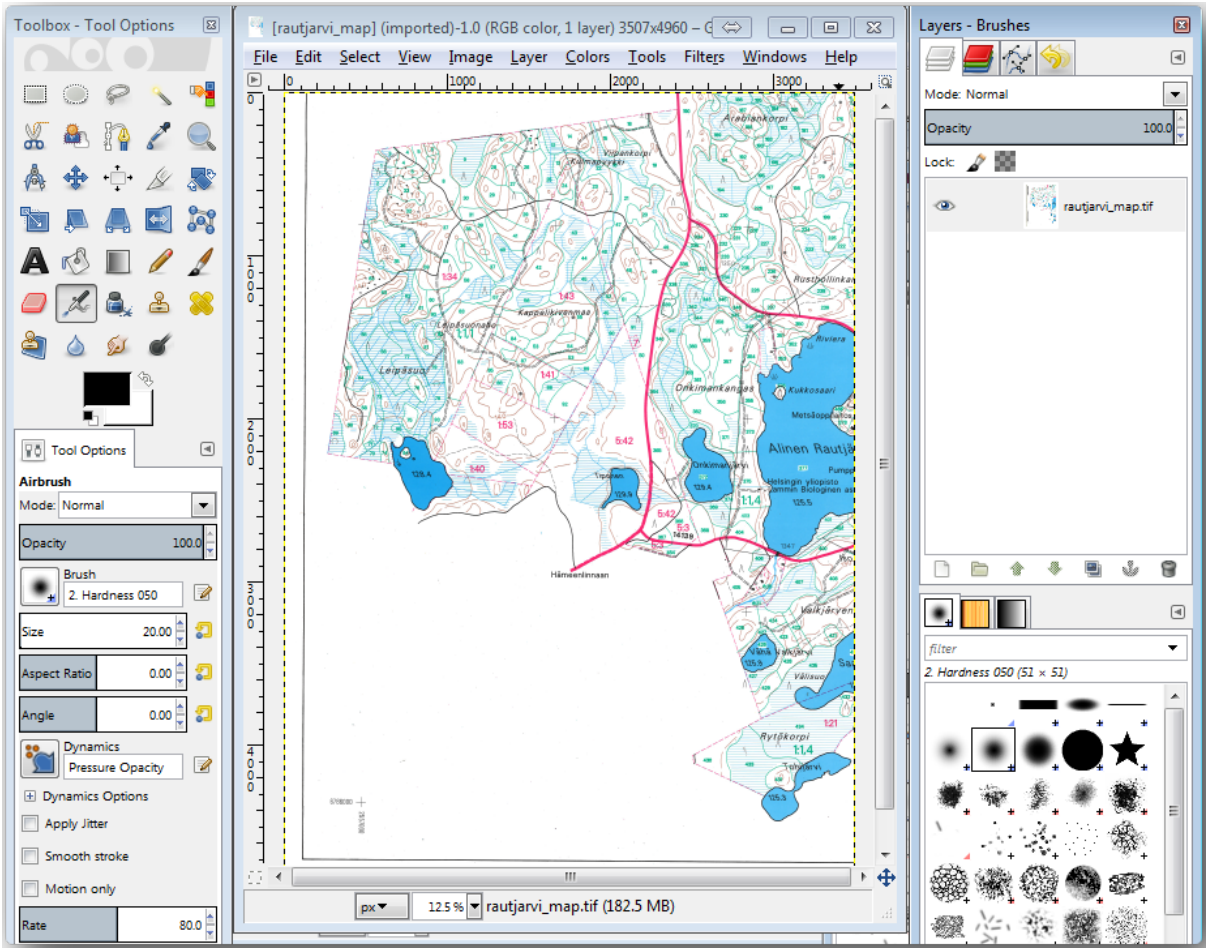
Open your `map_digitizing.qgs` project in QGIS, that you saved from the previous lesson.

Once you have scanned and georeferenced your map you could start to digitize directly by looking at the image as a guide. That would most likely be the way to go if the image you are going to digitize from is, for example, an aerial photograph.

If what you are using to digitize is a good map, as it is in our case, it is likely that the information is clearly displayed as lines with different colors for each type of element. Those colors can be relatively easy extracted as individual images using an image processing software like GIMP. Such separate images can be used to assist the digitizing, as you will see below.

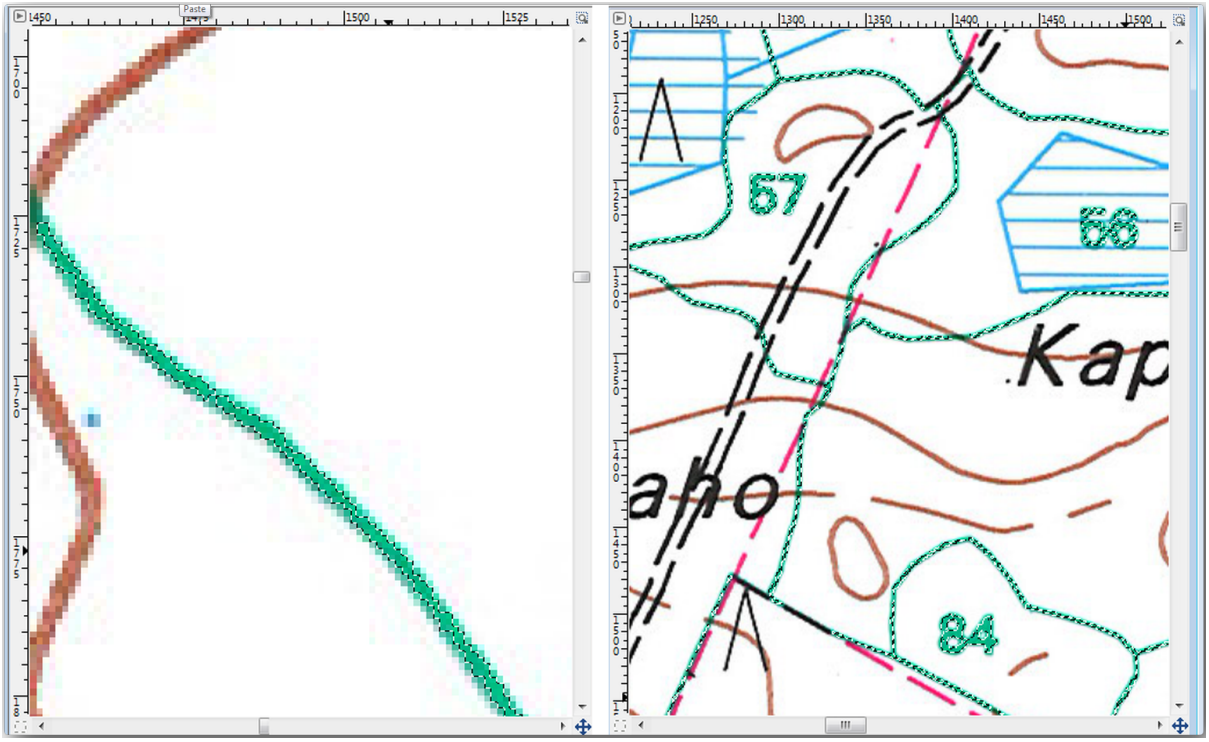
The first step will be to use GIMP to obtain an image that contains only the forest stands, that is, all those greenish lines that you could see in the original scanned map:

- Open GIMP (if you don't have it installed yet, download it from the internet or ask your teacher).
- Open the original map image, *File* ► *Open*, `rautjarvi_map.tif` in the `exercise_data/forestry` folder. Note that the forest stands are represented as green lines (with the number of the stand also in green inside each polygon).



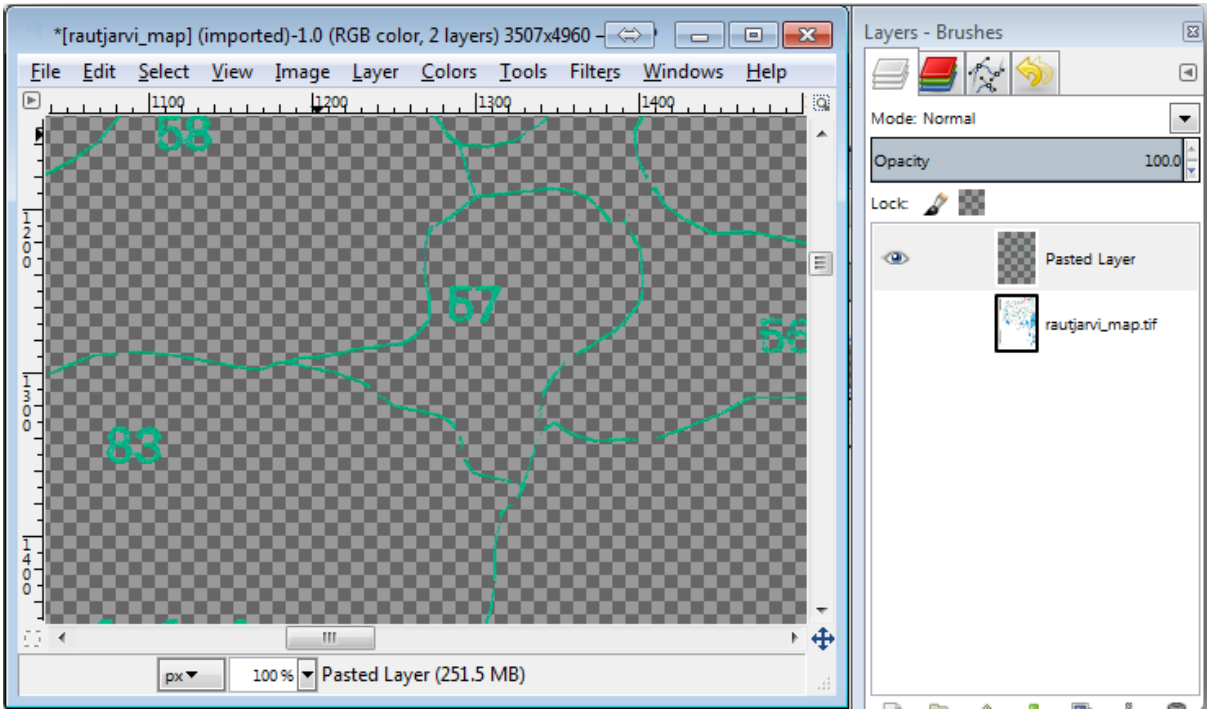
Now you can select the pixels in the image that are making up the forest stands" borders (the greenish pixels):

- Open the tool *Select* ► *By color*.
- With the tool active, zoom into the image (*Ctrl* + *mouse wheel*) so that a forest stand line is close enough to differentiate the pixels forming the line. See the left image below.
- Click and drag the mouse cursor in the middle of the line so that the tool will collect several pixel color values.
- Release the mouse click and wait a few seconds. The pixels matching the colors collected by the tool will be selected through the whole image.
- Zoom out to see how the greenish pixels have been selected throughout the image.
- If you are not happy with the result, repeat the click and drag operation.
- Your pixel selection should look something like the right image below.



Once you are done with the selection you need to copy this selection as a new layer and then save it as separate image file:

- Copy (*Ctr+C*) the selected pixels.
- And paste the pixels directly (*Ctr+V*), GIMP will display the pasted pixels as a new temporary layer in the *Layers - Brushes* panel as a *Floating Selection (Pasted Layer)*.
- Right click that temporary layer and select *To New Layer*.
- Click the «eye» icon next to the original image layer to switch it off, so that only the *Pasted Layer* is visible:



- Finally, select *File ► Export…*, set *Select File Type (By Extension)* as a *TIFF image*, select the `digitizing` folder and name it `rautjarvi_map_green.tif`. Select no compression when asked.

You could do the same process with other elements in the image, for example extracting the black lines that represent roads or the brown ones that represent the terrain“ contour lines. But for us, the forest stands is enough.

14.3.2  Try Yourself Georeference the Green Pixels Image

As you did in the previous lesson, you need to georeference this new image to be able to use it with the rest of your data.

Note that you don’t need to digitize the ground control points any more because this image is basically the same image as the original map image, as far as the Georeferencer tool is concerned. Here are some things you should remember:

- This image is also, of course, in `KKJ / Finland zone 2` CRS.
- You should use the ground control points you saved, *File ► Load GCP points*.
- Remember to review the *Transformation settings*.
- Name the output raster as `rautjarvi_green_georef.tif` in the `digitizing` folder.

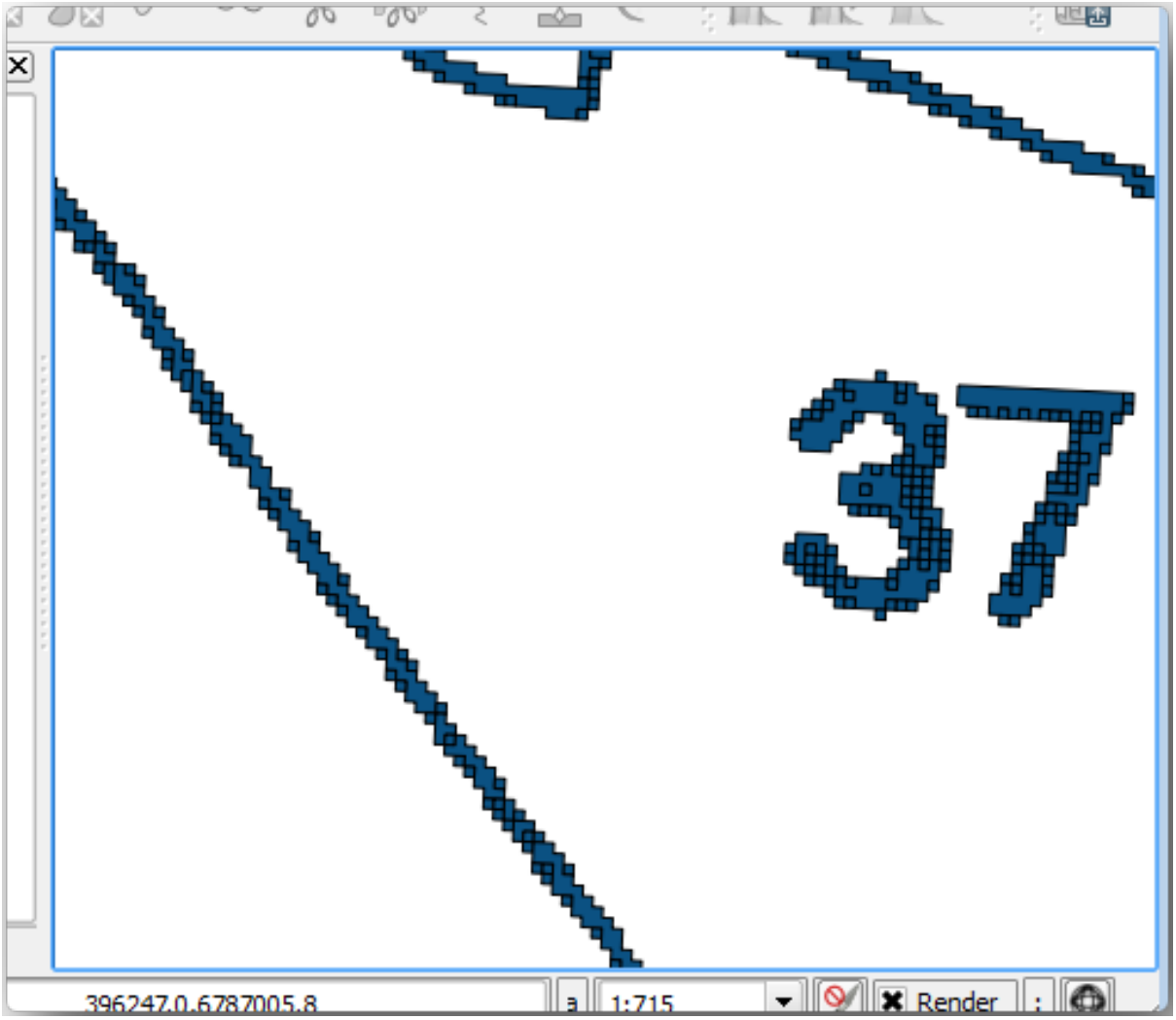
Check that the new raster is fitting nicely with the original map.

14.3.3  Follow Along: Creating Supporting Points for Digitizing

Having in mind the digitizing tools in QGIS, you might already be thinking that it would be helpful to snap to those green pixels while digitizing. That is precisely what you are going to do next create points from those pixels to use them later to help you follow the forest stands“ borders when digitizing, by using the snapping tools available in QGIS.

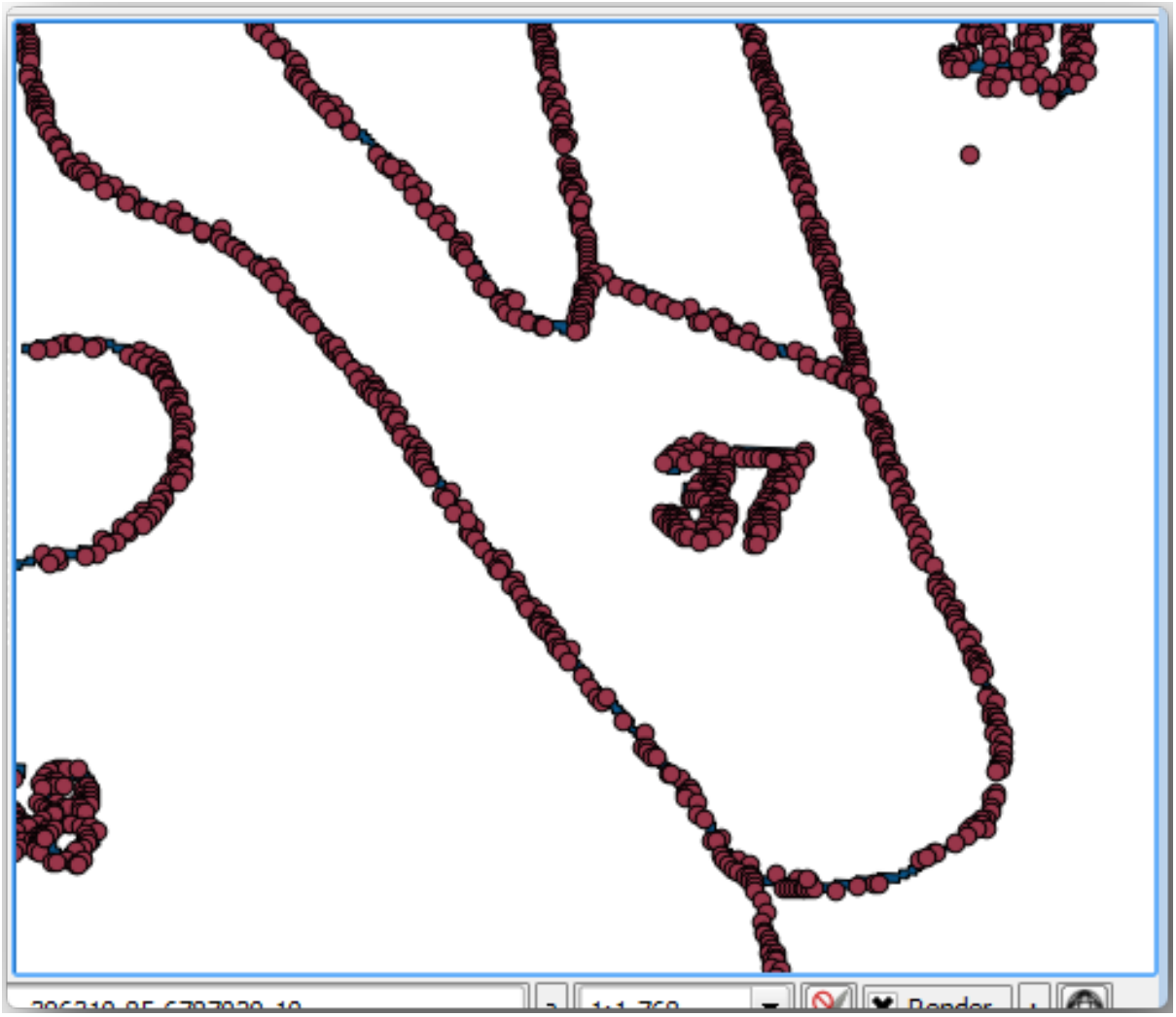
- Use the *Raster ► Conversion ► Polygonize (Raster to Vector)* tool to vectorize your green lines to polygons. If you don’t remember how, you can review it in *Занятие: Преобразование растрового в векторное*.
- Save as `rautjarvi_green_polygon.shp` inside the `digitizing` folder.

Zoom in and see what the polygons look like. You will get something like this:



Next one option to get points out of those polygons is to get their centroids:

- Open *Vector* ► *Geometry tools* ► *Polygon centroids*.
- Set the polygon layer you just got as the input file for the tool.
- Name the output as `green_centroids.shp` inside the `digitizing` folder.
- Check *Add result to canvas*.
- Run the tool to calculate the centroids for the polygons.



Now you can remove the *rautjarvi_green_polygon* layer from the TOC.

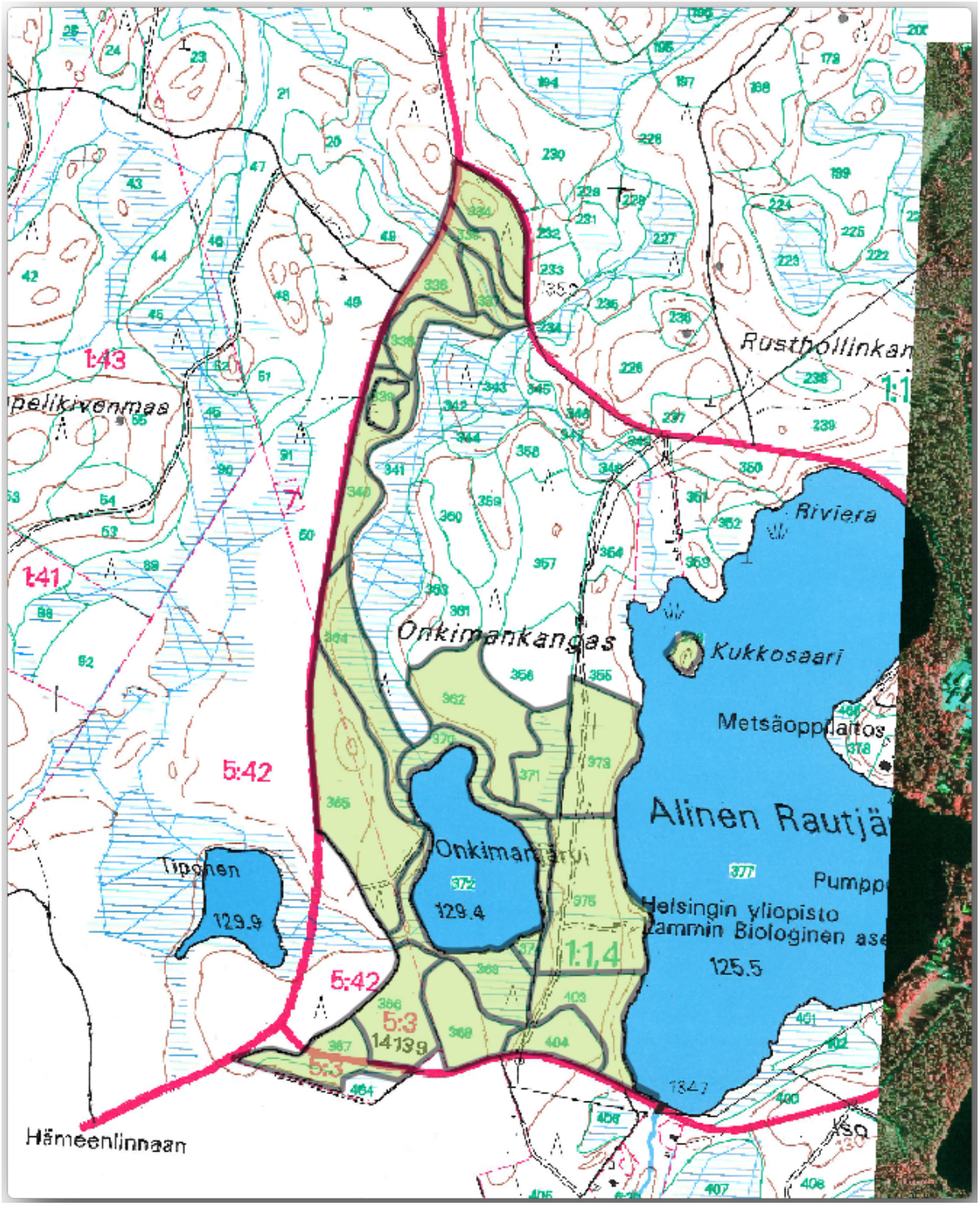
Change the symbology of the centroids layer as:

- Open the *Layer Properties* for *green_centroids*.
- Go to the *Symbology* tab.
- Set the *Unit* to Map unit.
- Set the *Size* to 1.

It is not necessary to differentiate points from each other, you just need them to be there for the snapping tools to use them. You can use those points now to follow the original lines much easily than without them.

14.3.4 Follow Along: Digitize the Forest Stands

Now you are ready to start with the actual digitizing work. You would start by creating a vector file of *polygon type*, but for this exercise, there is a shapefile with part of the area of interest already digitized. You will just finish digitizing the half of the forest stands that are left between the main roads (wide pink lines) and the lake:



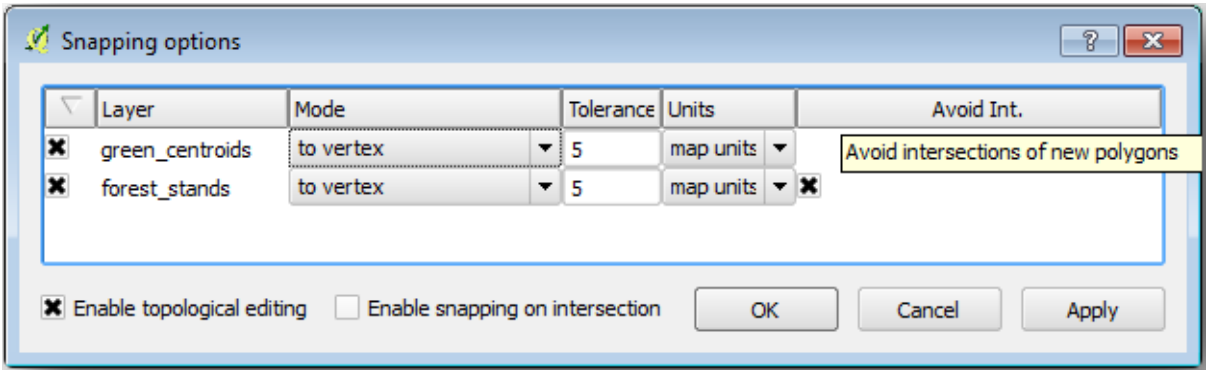
- Go to the `digitizing` folder using your file manager browser.
- Drag and drop the `forest_stands.shp` vector file to your map.

Change the new layer's symbology so that it will be easier to see what polygons have already been digitized:

- The filling of the polygon to green.
- The polygons" borders to 1 mm.
- and set the transparency to 50%.

Now, if you remember past modules, we have to set up and activate the snapping options:

- Go to *Project* ► *Snapping options…*.
- Activate the snapping for the `green_centroids` and the `forest_stands` layers.
- Set their *Tolerance* to 5 map units.
- Check the *Avoid Int.* box for the `forest_stands` layer.
- Check *Enable topological editing*.
- Click *Apply*.



With these snapping settings, whenever you are digitizing and get close enough to one of the points in the centroids layer or any vertex of your digitized polygons, a pink cross will appear on the point that will be snapped to.

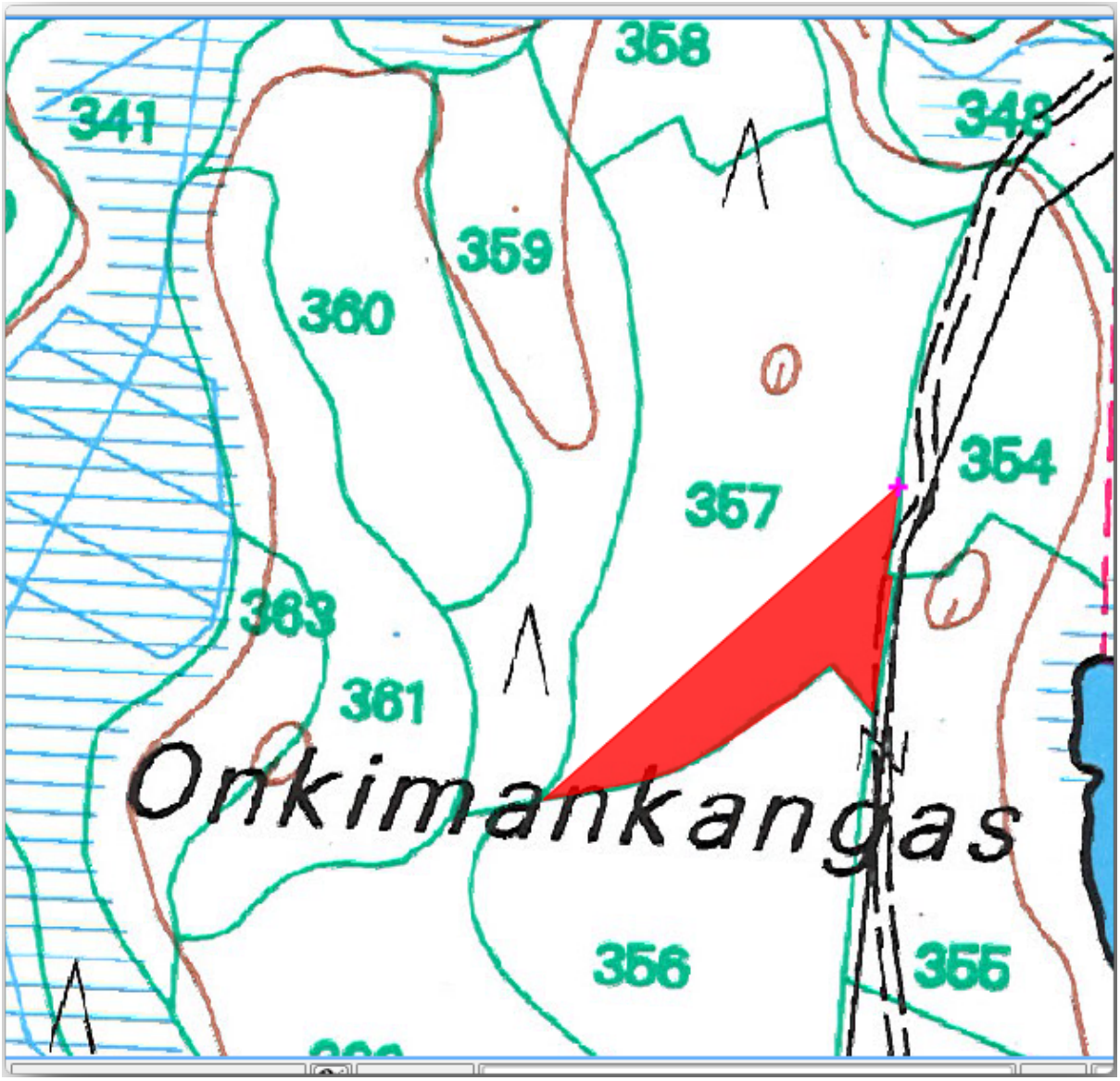
Finally, turn off the visibility of all the layers except `forest_stands` and `rautjarvi_georef`. Make sure that the map image has not transparency any more.

A couple of important things to note before you start digitizing:

- Don't try to be too accurate with the digitizing of the borders.
- If a border is a straight line, digitize it with just two nodes. In general, digitize using as few nodes as possible.
- Zoom in to close ranges only if you feel that you need to be accurate, for example, at some corners or when you want a polygon to connect with another polygon at a certain node.
- Use the mouse's middle button to zoom in/out and to pan as you digitize.
- Digitize only one polygon at a time.
- After digitizing one polygon, write the forest stand id that you can see from the map.

Now you can start digitizing:

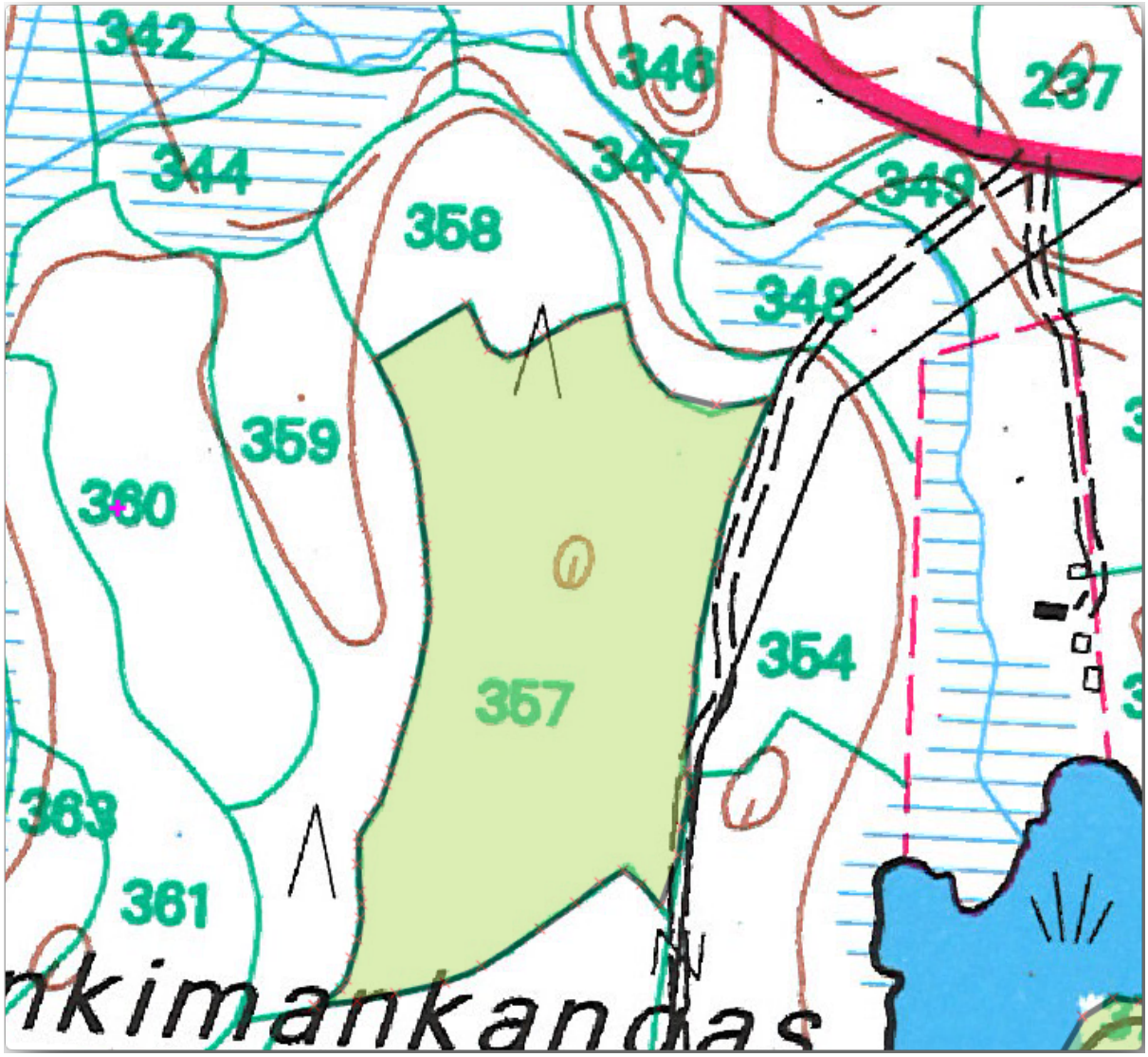
- Locate the forest stand number 357 in the map window.
- Enable editing for the `forest_stands.shp` layer.
- Select the *Add feature* tool.
- Start digitizing the stand `357` by connecting some of the dots.
- Note the pink crosses indicating the snapping.



- When you are done, right click to end digitizing that polygon.
- Enter the forest stand `id` (in this case 357).
- Click *OK*.

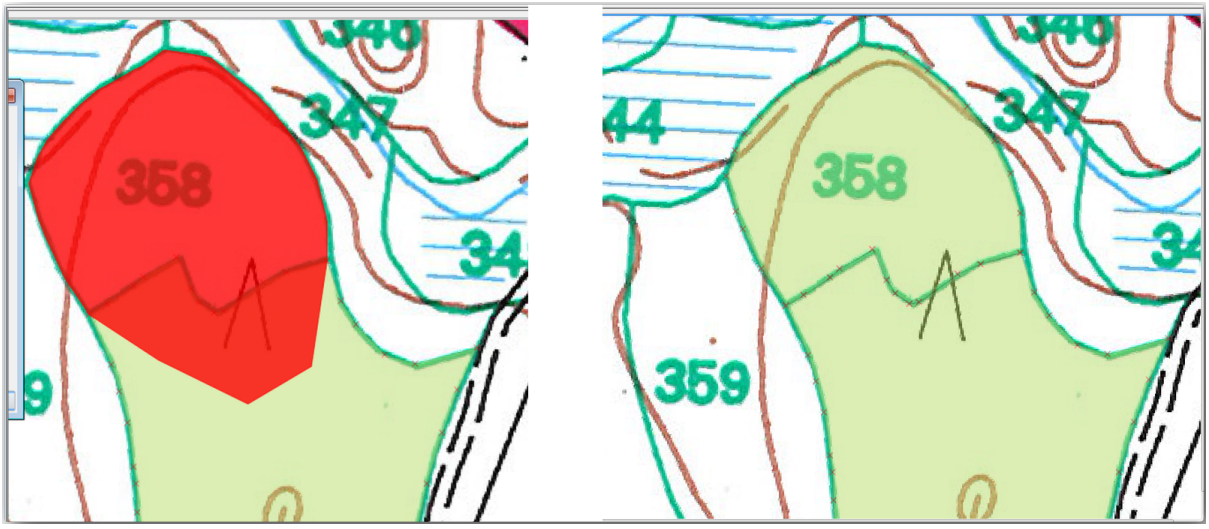
If you were not prompted for the polygon `id` when you finished digitizing it, go to *Settings* ► *Options* ► *Digitizing* and make sure that the *Suppress attribute form pop-up after feature creation* is not checked.

Your digitized polygon will look like this:



Now for the second polygon, pick up the stand number 358. Make sure that the *Avoid int.* is checked for the `forest_stands` layer. This option does not allow intersecting polygons at digitizing, so that if you digitize over an existing polygon, the new polygon will be trimmed to meet the border of the already existing polygons. You can use this characteristic to automatically obtain a common border.

- Begin digitizing the stand 358 at one of the common corners with the stand 357.
- Then continue normally until you get to the other common corner for both stands.
- Finally, digitize a few points inside polygon 357 making sure that the common border is not intersected. See left image below.
- Right click to finish editing the forest stand 358.
- Enter the `id` as 358.
- Click *OK*, your new polygon should show a common border with the stand 357 as you can see in the image on the right.



The part of the polygon that was overlapping the existing polygon has been automatically trimmed out and you are left with a common border, as you intended it to be.

14.3.5 Try Yourself Finish Digitizing the Forest Stands

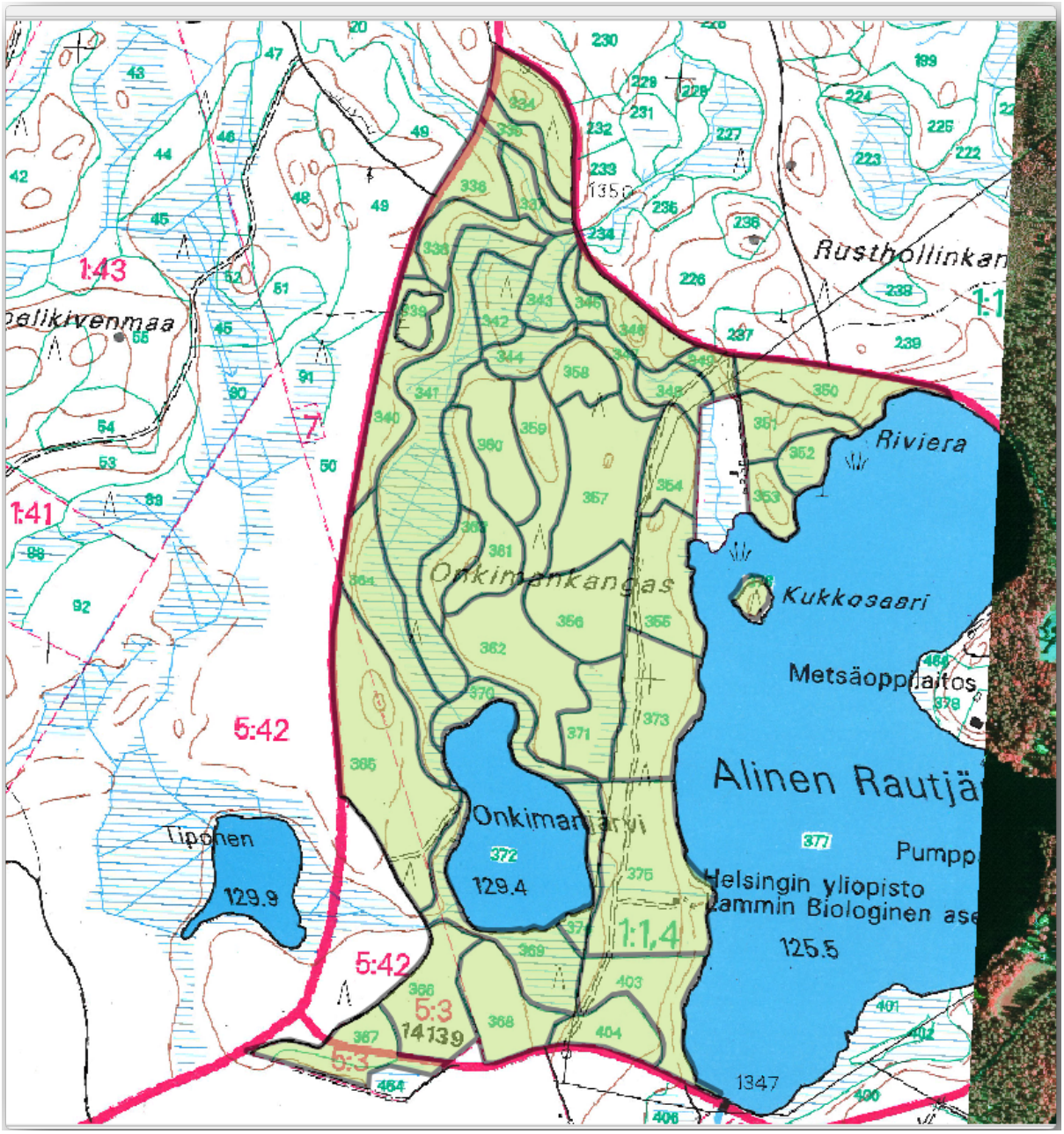
Now you have two forest stands ready. And a good idea on how to proceed. Continue digitizing on your own until you have digitized all the forest stands that are limited by the main road and the lake.

It might look like a lot of work, but you will soon get used to digitizing the forest stands. It should take you about 15 minutes.

During the digitizing you might need to edit or delete nodes, split or merge polygons. You learned about the necessary tools in *Занятие: Топология функций*, now is probably a good moment to go read about them again.

Remember that having *Enable topological editing* activated, allows you to move nodes common to two polygons so that the common border is edited at the same time for both polygons.

Your result will look like this:

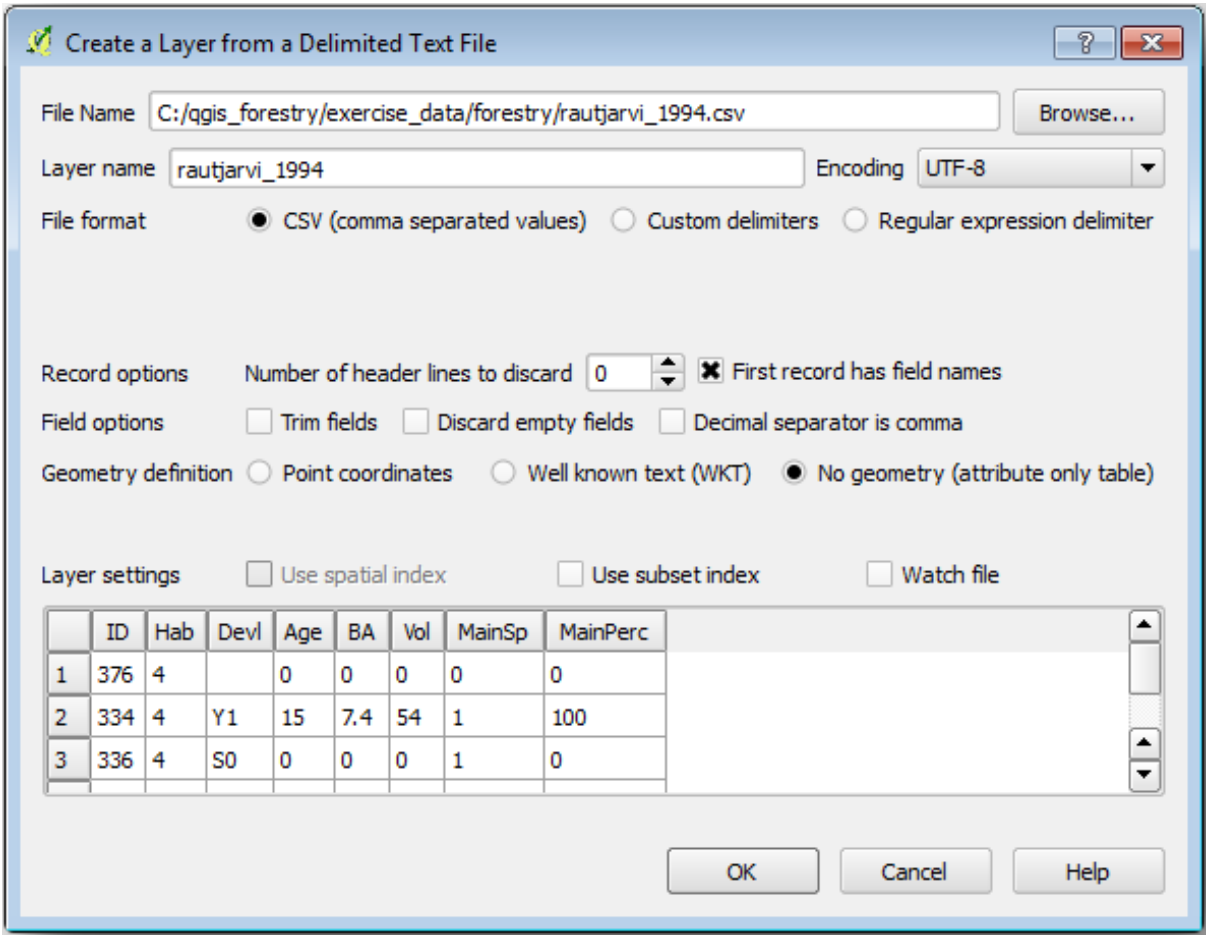


14.3.6 Follow Along: Joining the Forest Stand Data

It is possible that the forest inventory data you have for you map is also written in paper. In that case, you would have to first write that data to a text file or a spreadsheet. For this exercise, the information from the inventory for 1994 (the same inventory as the map) is ready as a comma separated text (csv) file.

Open the `rautjarvi_1994.csv` file from the `exercise_data\forestry` directory in a text editor and note that the inventory data file has an attribute called `ID` that has the numbers of the forest stands. Those numbers are the same as the forest stands ids you have entered for your polygons and can be used to link the data from the text file to your vector file. You can see the metadata for this inventory data in the file `rautjarvi_1994_legend.txt` in the same folder.

- Open the `.csv` in QGIS with the *Layer* ► *Add Delimited Text Layer…* tool. In the dialog, set it as follows:



To add the data from the `.csv` file:

- Open the Layer Properties for the `forest_stands` layer.
- Go to the *Joins* tab.
- Click the plus sign on the bottom of the dialog box.
- Select `rautjarvi_1994.csv` as the *Join layer* and `ID` as the *Join* field.
- Make sure that the *Target* field is also set to `id`.
- Click *OK* two times.

The data from the text file should be now linked to your vector file. To see what has happened, open the attribute table for the `forest_stands` layer. You can see that all the attributes from the inventory data file are now linked to your digitized vector layer.

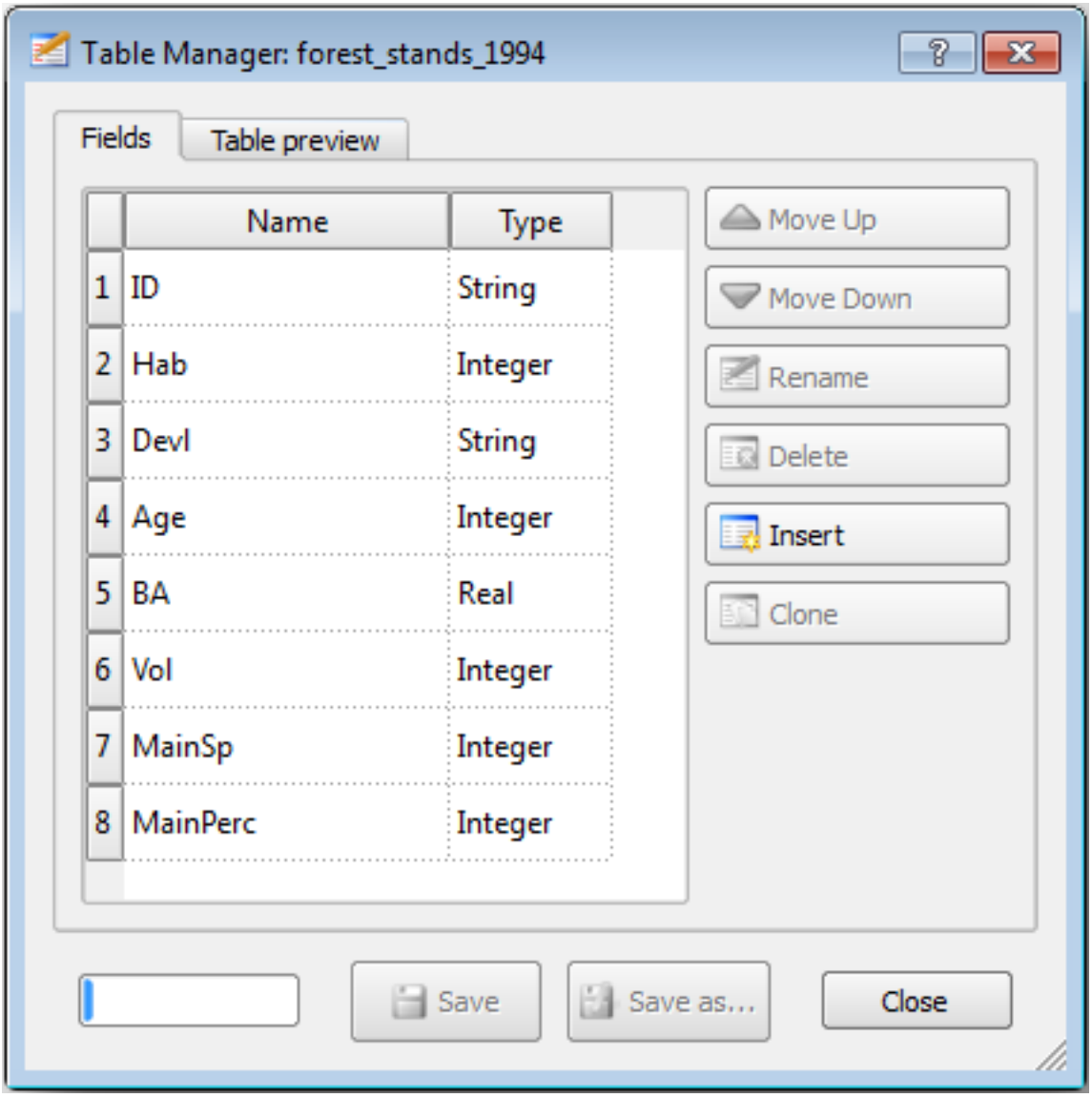
14.3.7  Try Yourself Renaming Attribute Names and Adding Area and Perimeter

The data from the `.csv` file is just linked to your vector file. To make this link permanent, so that the data is actually recorded to the vector file you need to save the `forest_stands` layer as a new vector file. Close the attribute table and right click the forest_stands layer to save it as `forest_stands_1994.shp`.

Open your new `forest_stands_1994.shp` in your map if you did not added yet. Then open the attribute table. You notice that the names of the columns that you just added are no very useful. To solve this:

- Add the plugin *Table Manager* as you have done with other plugins before.

- Make sure the plugin is activated.
- In the TOC select the layer `forest_stands_1994.shp`.
- Then, go to *Vector* ► *Table Manager* ► *Table manager*.
- Use the dialogue box to edit the names of the columns to match the ones in the `.csv` file.



- Click on *Save*.
- Select *Yes* to keep the layer style.
- Close the *Table Manager* dialogue.

To finish gathering the information related to these forest stands, you might calculate the area and the perimeter of the stands. You calculated areas for polygons in *Занятие: Дополнительное упражнение*. Go back to that lesson if you need to and calculate the areas for the forest stands, name the new attribute `Area` and make sure that the values calculated are in hectares.

Now your `forest_stands_1994.shp` layer is ready and packed with all the available information.

Save your project to keep the current map presentation in case you need to come back later to it.

14.3.8 In Conclusion

It has taken a few clicks of the mouse but you now have your old inventory data in digital format and ready for use in QGIS.

14.3.9 What's Next?

You could start doing different analysis with your brand new dataset, but you might be more interested in performing analysis in a dataset more up to date. The topic of the next lesson will be the creation of forest stands using current aerial photos and the addition of some relevant information to your dataset.

14.4 Lesson: Updating Forest Stands

Now that you have digitized the information from the old inventory maps and added the corresponding information to the forest stands, the next step would be to create the inventory of the current state of the forest.

You will digitize new forest stands from scratch following an aerial photo from that forest area. The forestry map you digitized in the previous lesson was created from an aerial Color Infrared (CIR) photograph. This type of imagery, where the infrared light is recorded instead of the blue light, are widely used to study vegetated areas. You will also use a CIR photograph in this lesson.

After digitizing the forest stands, you will add information such as new constraints given by conservation regulations.

The goal for this lesson: To digitize a new set of forest stands from CIR aerial photographs and add information from other data-sets.

14.4.1  Comparing the Old Forest Stands to Current Aerial Photographs

The National Land Survey of Finland has an open data policy that allows you downloading a variety of geographical data like aerial imagery, traditional topographic maps, DEM, LiDAR data, etc. The service can be accessed also in English here. The aerial image used in this exercise has been created from two orthorectified CIR images downloaded from that service (M4134F_21062012 and M4143E_21062012).

- Open QGIS and set the project's CRS to *ETRS89 / ETRS-TM35FIN* in *Project* ► *Properties…* ► *CRS*.
- From the `exercise_data\forestry\` folder, add the CIR image `rautjarvi_aerial.tif` that is containing the digitized lakes.
- Then save the QGIS project as `digitizing_2012.qgs`.

The CIR images are from 2012. You can compare the stands that were created in 1994 with the situation almost 20 years later.

- Add your `forest_stands_1994.shp` layer.
- Set its styling so that you can see through your polygons.
- Review how the old forest stands follow (or not) what you might visually interpret as an homogeneous forest.

Zoom and pan around the area. You probably will notice that some of the old forest stands might be still corresponding with the image but others are not.

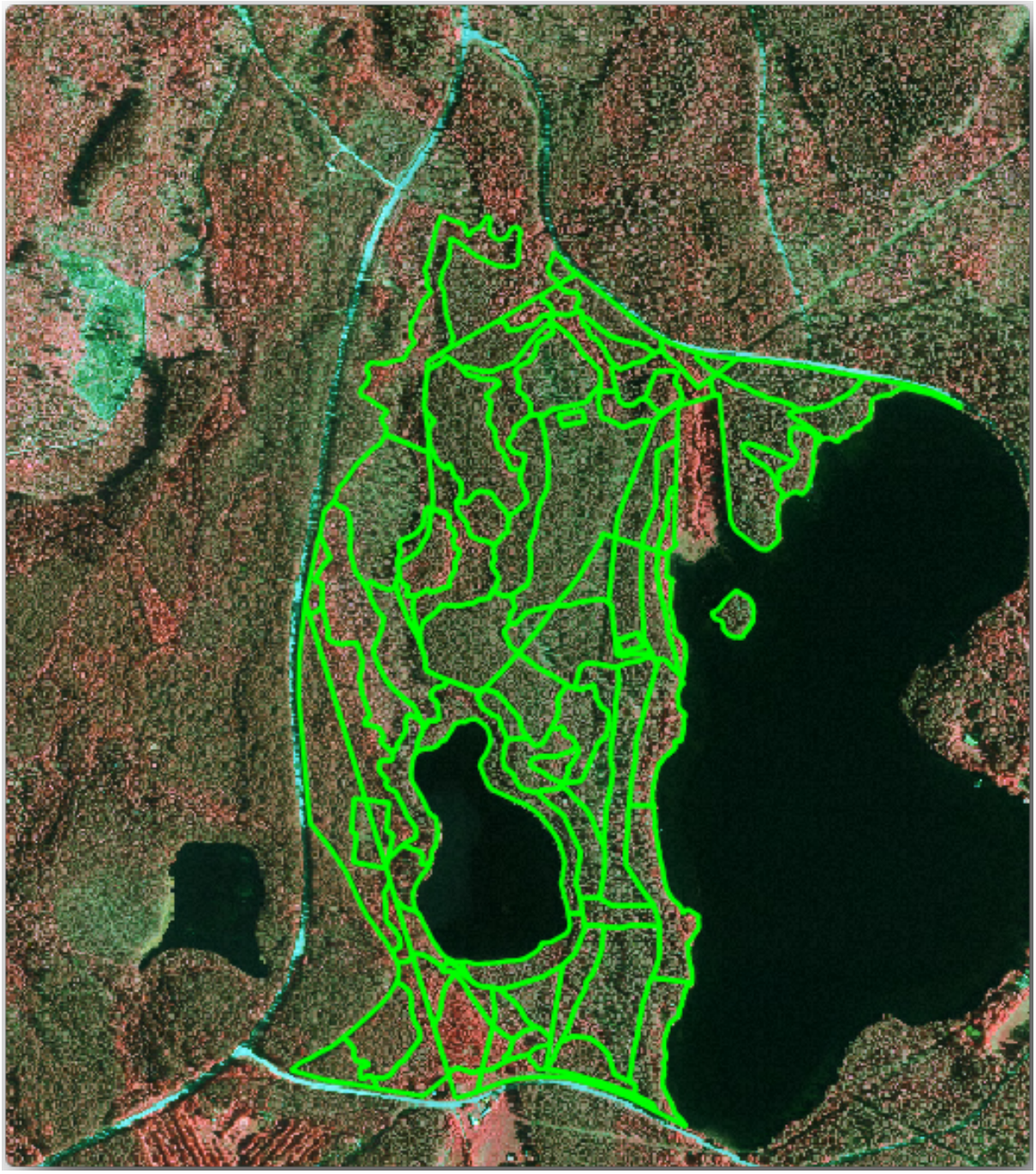
This is a normal situation, as some 20 years have passed by and different forest operations have been done (harvesting, thinning…). It is also possible that the forest stands looked homogeneous back in 1992 to the person who digitized them but as time has passed some forest has developed in different ways. Or simply the priorities for the forest inventory were different that they are today.

Next, you will create new forest stands for this image without using the old ones. Later you can compare them to see the differences.

14.4.2 Interpreting the CIR Image

Let's digitize the same area that was covered by the old inventory, limited by the roads and the lake. You don't have to digitize the whole area, as in the previous exercise you can start with a vector file that already contains most of the forest stands.

- Remove the `forest_stands_1994.shp` layer.
- Add the `forest_stands_2012.shp` layer, located in the `exercise_data\forestry\` folder.
- Set the styling of this layer so that the polygons have no fill and the borders are visible.

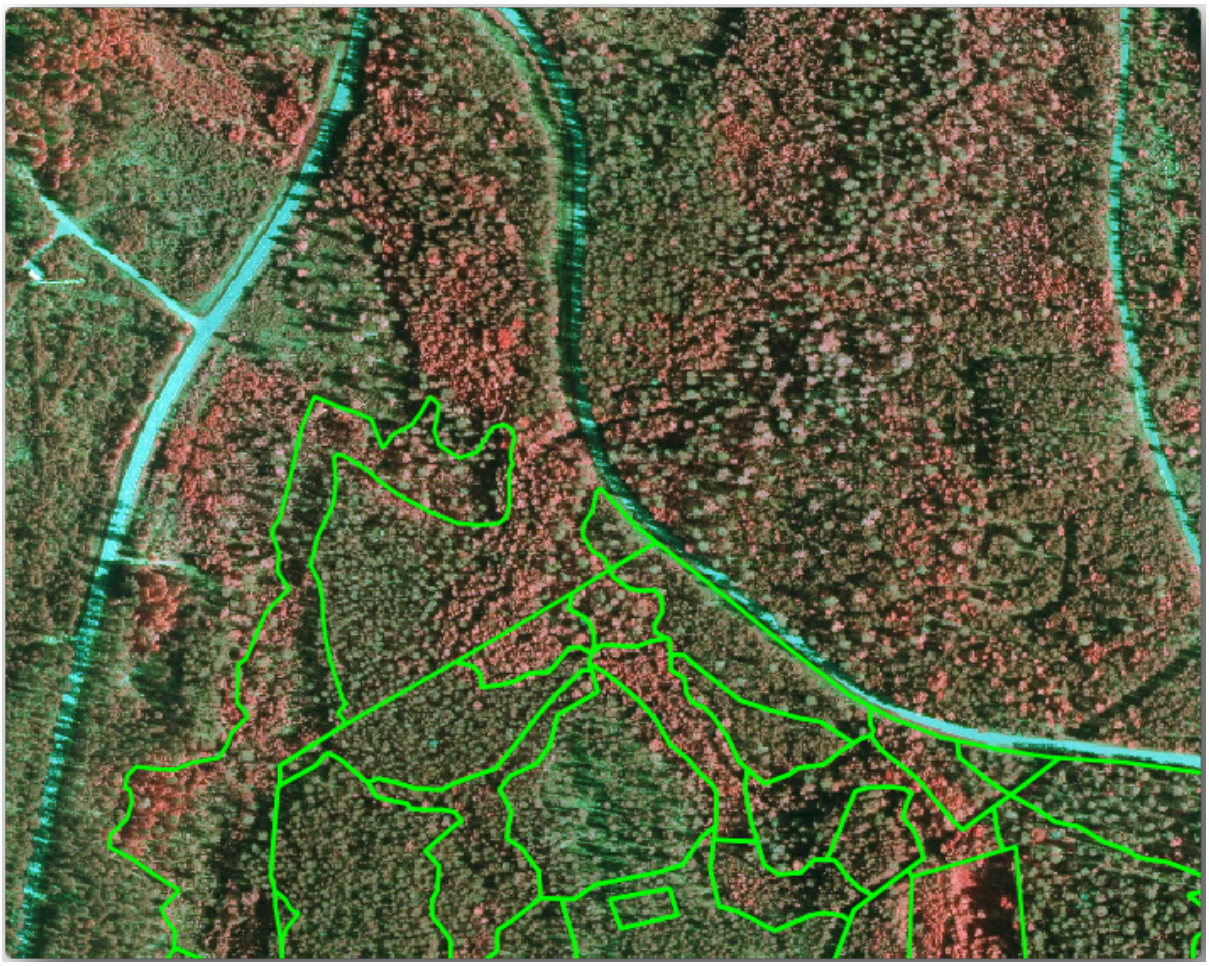


You can see that a region to the North of the inventory area is still missing. That will be your task, digitizing the missing forest stands.

But before you start, spend some time reviewing the forest stands already digitized and the corresponding forest in the image. Try to get an idea about how the stands borders are decided, it helps if you have some forestry knowledge.

Some ideas about what you could identify from the images:

- What forests are deciduous species (in Finland mostly birch forests) and which ones are conifers (in this region pine or spruce). In CIR images, deciduous species will often come as bright red color whereas conifers present dark green colors.
- When a forest stand age changes, by looking at the sizes of the tree crowns that can be identified in the imagery.
- The different forest stands“ densities, for example forest stand were a thinning operation has recently been done would clearly show spaces between the tree crowns and should be easy to differentiate from other forest stands around it.
- Blueish areas indicate barren terrain, roads and urban areas, crops that have not started to grow etc.
- Don’t use zooms too close to the image when trying to identify forest stands. A scale between 1:3 000 and 1: 5 000 should be enough for this imagery. See the image below (1:4000 scale):

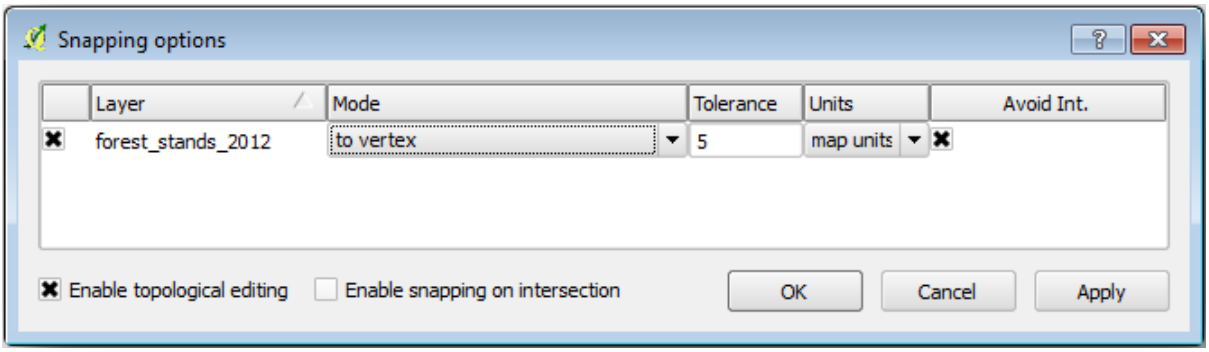


14.4.3 Try Yourself Digitizing Forest Stands from CIR Imagery

When digitizing the forest stands, you should try to get forest areas that are as homogeneous as possible in terms of tree species, forest age, stand density… Don't be too detailed though, or you will end up making hundreds of small forest stands that would not be useful at all. You should try to get stands that are meaningful in the context of forestry, not too small (at least 0.5 ha) but not too big either (no more than 3 ha).

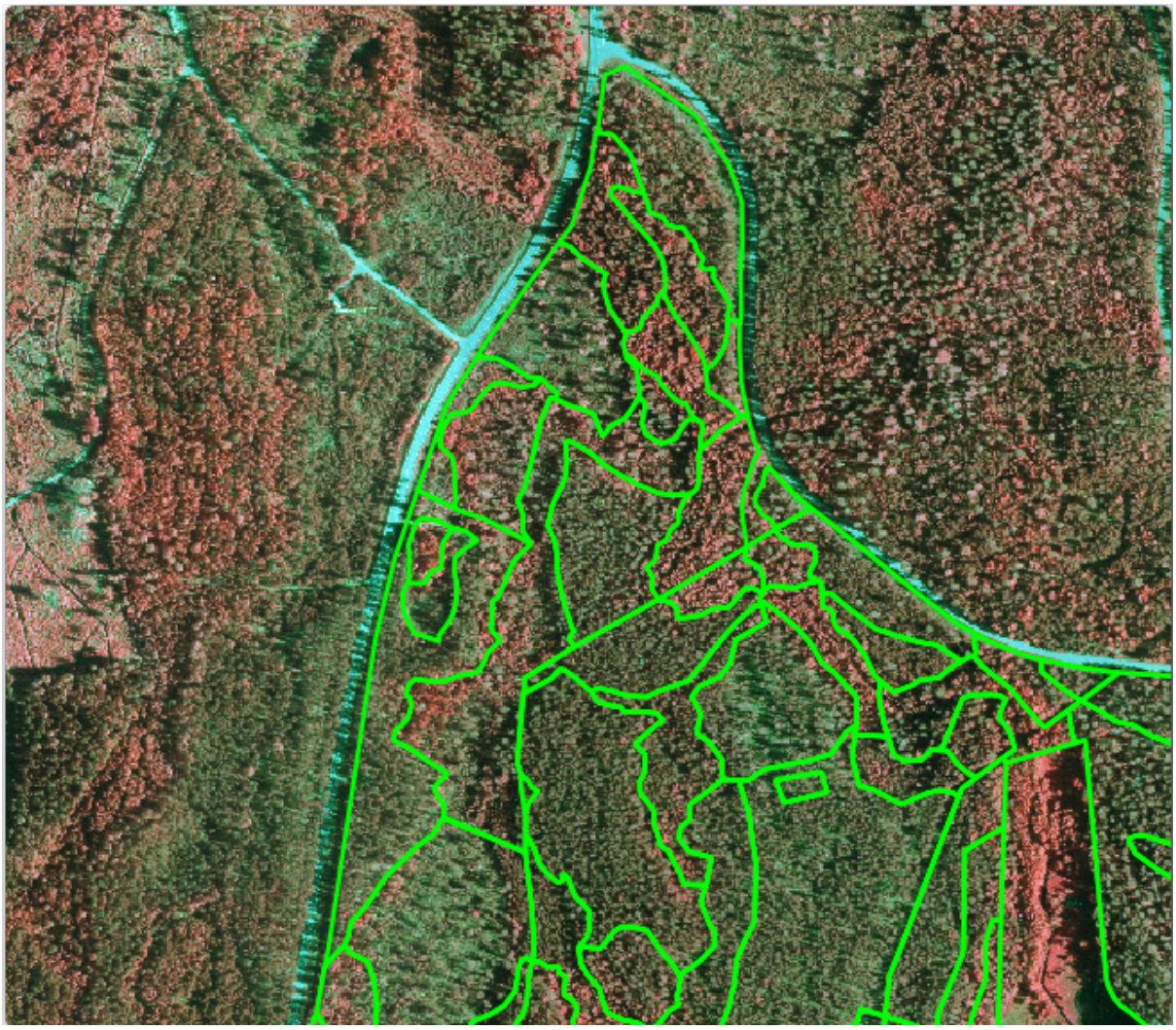
With this indications in mind, you can now digitize the missing forest stands.

- Enable editing for `forest_stands_2012.shp`.
- Set up the snapping and topology options as in the image.
- Remember to click *Apply* or *OK*.



Start digitizing as you did in the previous lesson, with the only difference that you don't have any point layer that you are snapping to. For this area you should get around 14 new forest stands. While digitizing, fill in the `Stand_id` field with numbers starting at `901`.

When you are finished your layer should look something like:



Now you have a new set of polygons defining the different forest stands for the current situation as can interpreted from the CIR images. But you are obviously still missing the forest inventory data, right? For that you will still need to visit the forest and get some sample data that you will use to estimate the forest attributes for each of the forest stands. You will see how to do that in the next lesson.

For the moment, you still can improve your vector layer with some extra information that you have about conservation regulation that should be taken into account for this area.

14.4.4 Follow Along: Updating Forest Stands with Conservation Information

For the area you are working with, it has been researched that the following conservation regulations must be taken into account while doing the forest planning:

- Two locations of a protected species of Siberian flying squirrel (Pteromys volans) have been identified. According to the regulation, an area of 15 meters around the spots must be left untouched.
- A riparian forest of special interest growing along a stream in the area must be protected. In a visit to the field, it was found that 20 meters to both sides of the stream must be protected.

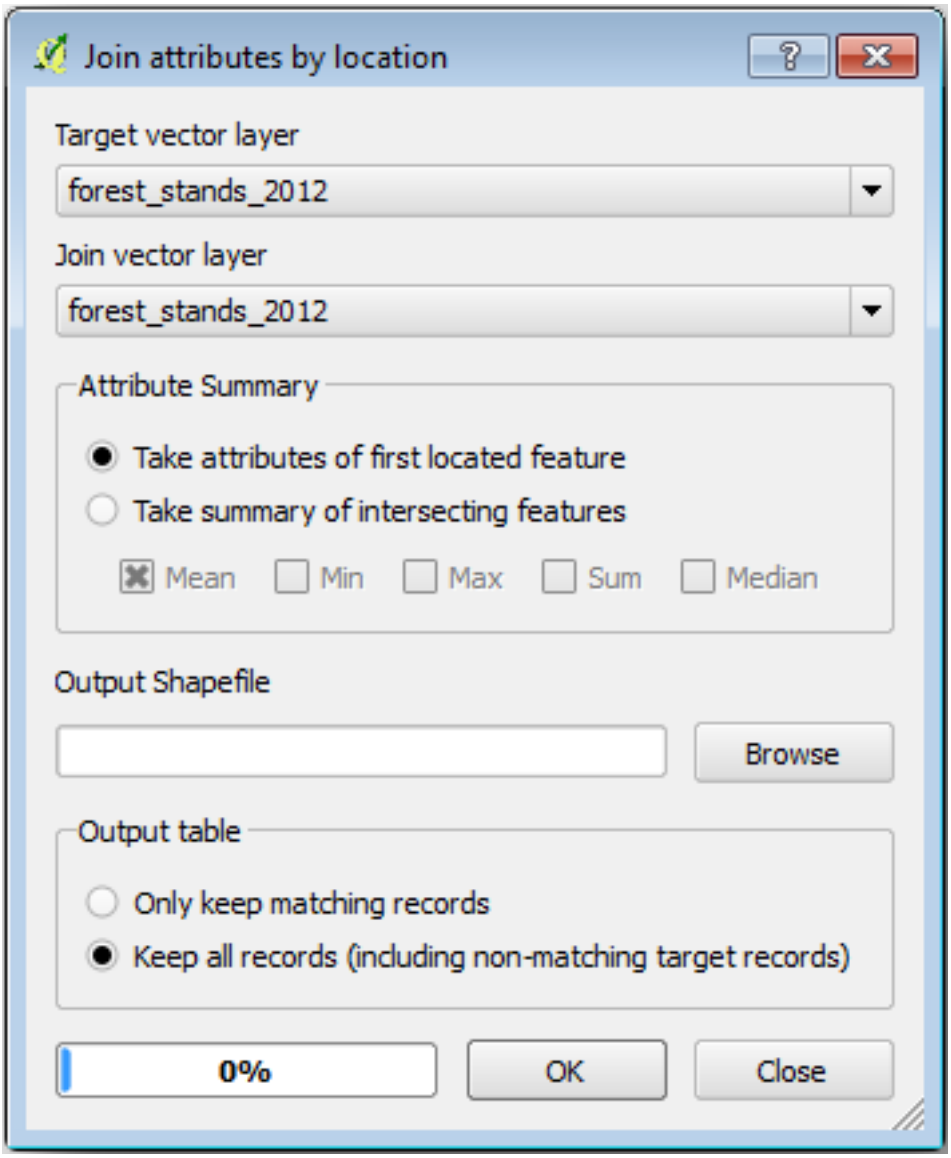
You have one vector file containing the information about the squirrel locations and another containing the digitized stream running in the North area towards the lake. From the `exercise_data\forestry\` folder, add the vector files `squirrel.shp` and `stream.shp`.

For the protection of the squirrels locations, you are going to add a new attribute (column) to your new forest stands that will contain information about point locations that have to be protected. That information will later be available whenever a forest operation is planned, and the field team will be able to mark the area that has to be left untouched before the work starts.

- Open the attribute table for the `squirrel` layer.
- You can see that there are two locations that are defined as Siberian flying squirrel, and that the area to be protected is indicated by a distance of 15 meters from the locations.

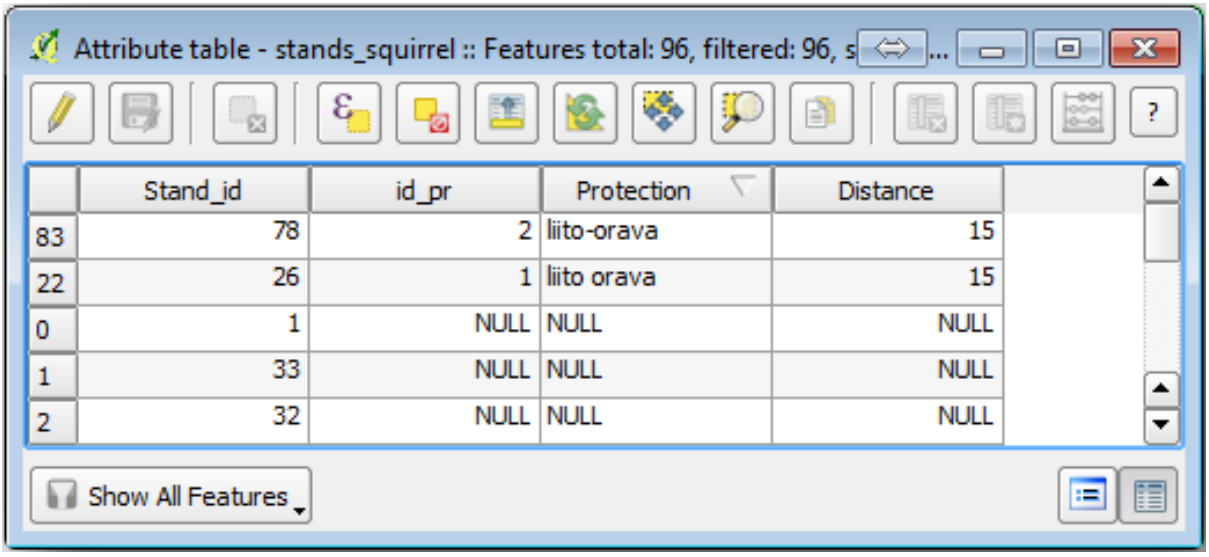
To join the information about the squirrels to your forest stands, you can use the *Join attributes by location*:

- Open *Vector* ► *Data Management Tools* ► *Join attributes by location*.
- Set the `forest_stands_2012.shp` layer as the *Target vector layer*.
- As *Join vector layer* select the `squirrel.shp` point layer.
- Name the output file as `stands_squirrel.shp`.
- In *Output table* select *Keep all records (including non-matching target records)*. So that you keep all the forest stands in the layer instead of only keeping those that are spatially related to the squirrel locations.
- Click *OK*.
- Select *Yes* when prompted to add the layer to the TOC.
- Close the dialogue box.



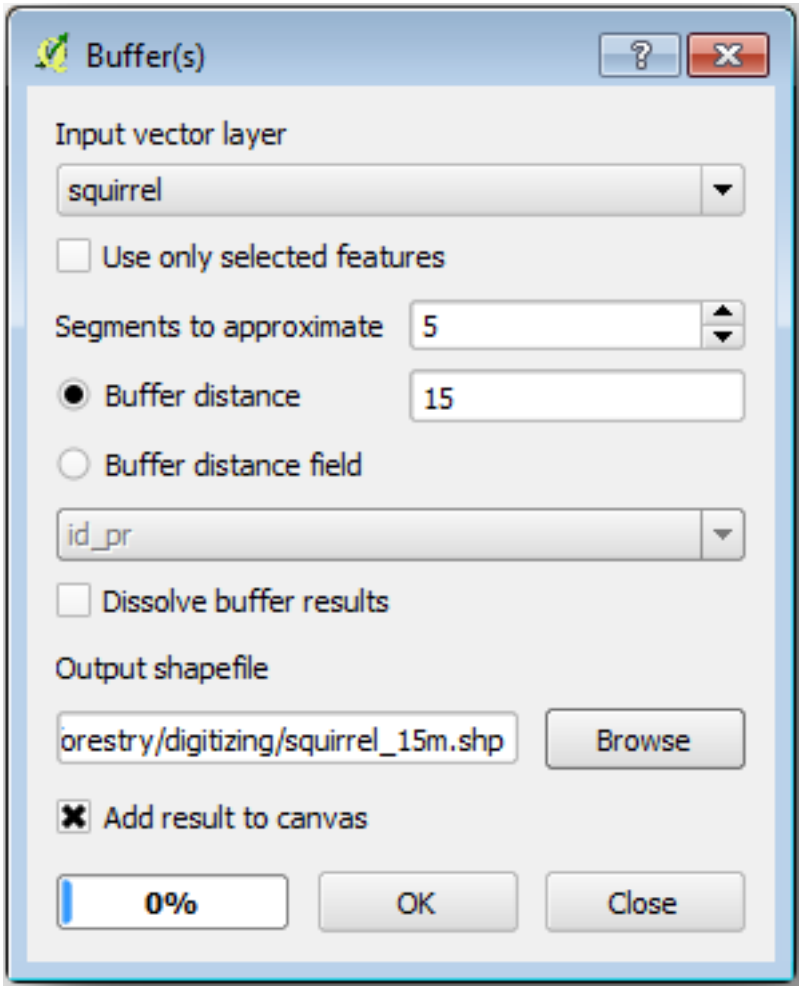
Now you have a new forest stands layer, `stands_squirrel` where there are new attributes corresponding to the protection information related to the Siberian flying squirrel.

Open the table of the new layer and order it so that the forest stands with information for the *Protection* attribute are on top. You should have now two forest stands where the squirrel has been located:

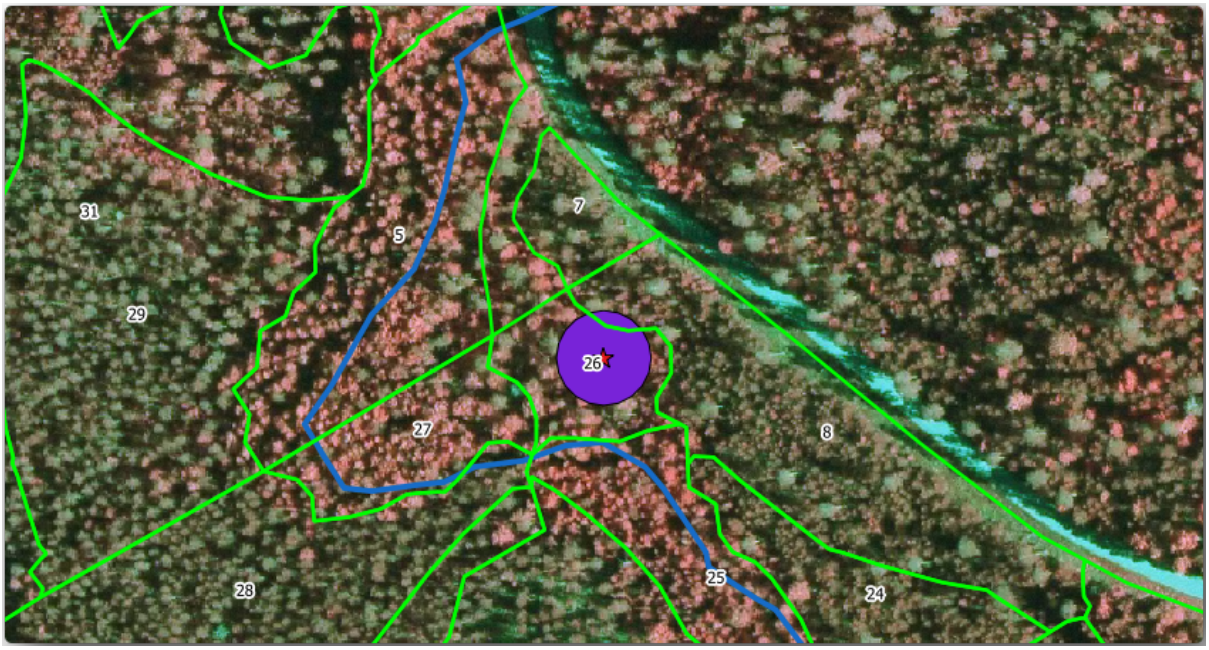


Although this information might be enough, look at what areas related to the squirrels should be protected. You know that you have to leave a buffer of 15 meters around the squirrels location:

- Open *Vector* ► *Geoprocessing Tools* ► *Buffer*.
- Make a buffer of 15 meters for the `squirrel` layer.
- Name the result `squirrel_15m.shp`.

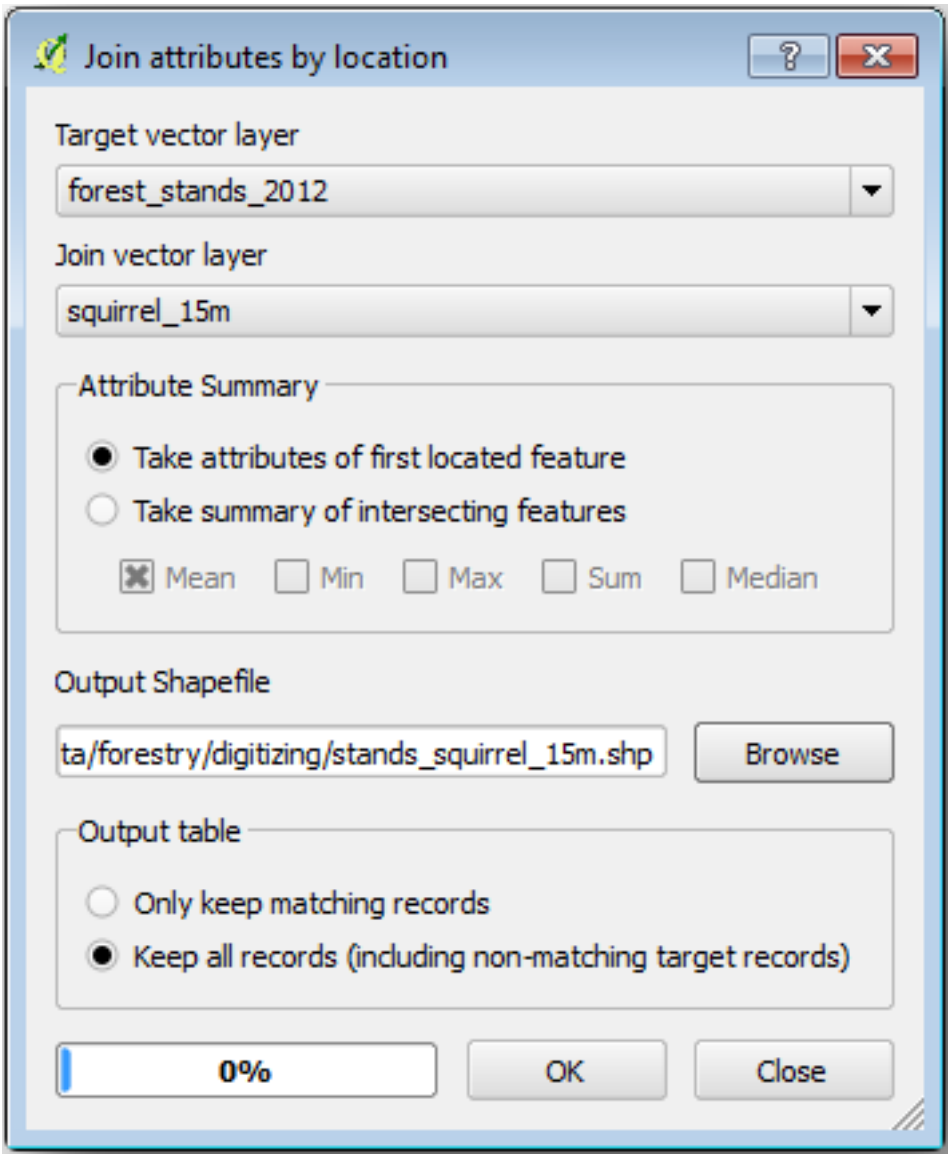


You will notice that if you zoom in to the location in the Northern part of the area, the buffer area extends to the neighbouring stand as well. This means that whenever a forest operation would take place in that stand, the protected location should also be taken into account.



From your previous analysis, you did not get that stand to register information about the protection status. To solve this problem:

- Run the *Join attributes by location* tool again.
- But this time use the `squirrel_15m` layer as join layer.
- Name the output file as `stands_squirrel_15m.shp`.



Open the attribute table for the this new layer and note that now you have three forest stands that have the information about the protection locations. The information in the forest stands data will indicate to the forest manager that there are protection considerations to be taken into account. Then he or she can get the location from the `squirrel` dataset, and visit the area to mark the corresponding buffer around the location so that the operators in the field can avoid disturbing the squirrels environment.

14.4.5 Try Yourself Updating Forest Stands with Distance to the Stream

Following the same approach as indicated for the protected squirrel locations you can now update your forest stands with protection information related to the stream identified in the field:

- Remember that the buffer in this case is 20 meters around it.
- You want to have all the protection information in the same vector file, so use the `stands_squirrel_15m` layer as the target.
- Name your output as `forest_stands_2012_protect.shp`.

Open the attributes table for the new vector layer and confirm that you now have all the protection information for the stands that are affected by the protection measures to protect the riparian forest associated with the stream.

Save your QGIS project.

14.4.6 In Conclusion

You have seen how to interpret CIR images to digitize forest stands. Of course it would take some practice to make more accurate stands and usually using other information like soil maps would give better results, but you know now the basis for this type of task. And adding information from other datasets resulted to be quite a trivial task.

14.4.7 What's Next?

The forest stands you digitized will be used for planning forestry operations in the future, but you still need to get more information about the forest. In the next lesson, you will see how to plan a set of sampling plots to inventory the forest area you just digitized, and get the overall estimate of forest parameters.

14.5 Lesson: Systematic Sampling Design

You have already digitized a set of polygons that represent the forest stands, but you don't have information about the forest just yet. For that purpose you can design a survey to inventory the whole forest area and then estimate its parameters. In this lesson you will create a systematic set of sampling plots.

When you start planning your forest inventory it is important to clearly define the objectives, the types of sample plots that will be used, and the data that will be collected to achieve the objectives. For each individual case, those will depend on the type of forest and the management purpose; and should be carefully planned by someone with forestry knowledge. In this lesson, you will implement a theoretical inventory based on a systematic sampling plot design.

The goal for this lesson: To create a systematic sampling plot design to survey the forest area.

14.5.1 Inventorying the Forest

There are several methods to inventory forests, each of them suiting different purposes and conditions. For example, one very accurate way to inventory a forest (if you consider only tree species) would be to visit the forest and make a list of every tree and their characteristics. As you can imagine this is not commonly applicable except for some small areas or some special situations.

The most common way to find out about a forest is by sampling it, that is, taking measurements in different locations at the forest and generalizing that information to the whole forest. These measurements are often made in *sample plots* that are smaller forest areas that can be easily measured. The sample plots can be of any size (for ex. 50 m2, 0.5 ha) and form (for ex. circular, rectangular, variable size), and can be located in the forest in different ways (for ex. randomly, systematically, along lines). The size, form and location of the sample plots are usually decided following statistical, economical and practical considerations. If you have no forestry knowledge, you might be interested in reading this Wikipedia article.

14.5.2  Follow Along: Implementing a Systematic Sampling Plot Design

For the forest you are working with, the manager has decided that a systematic sampling design is the most appropriate for this forest and has decided that a fixed distance of 80 meters between the sample plots and sampling lines will yield reliable results (for this case, +- 5% average error at a probability of 68%). Variable size plots has been decided to be the most effective method for this inventory, for growing and mature stands, but a 4 meters fixed radius plots will be used for seedling stands.

In practice, you simply need to represented the sample plots as points that will be used by the field teams later:

1. Open your `digitizing_2012.qgs` project from the previous lesson.

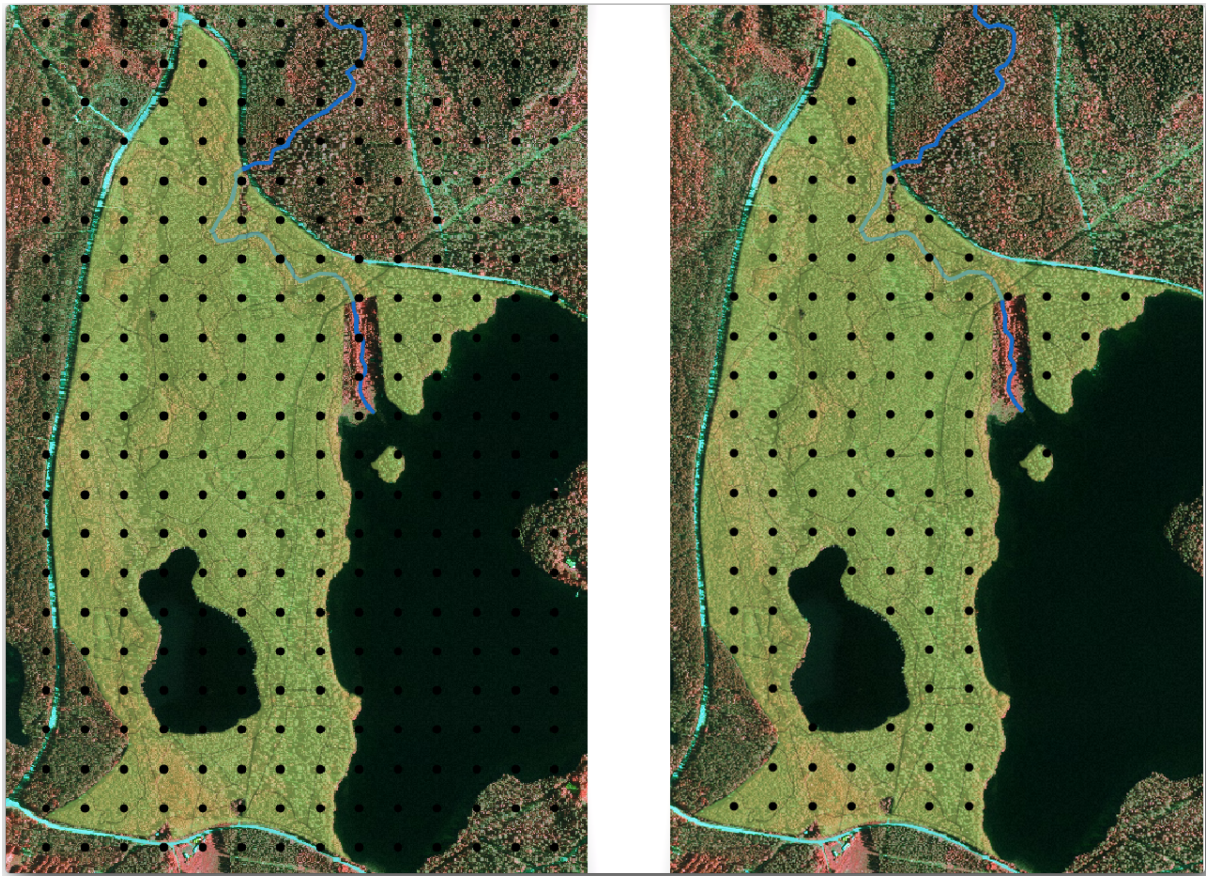
2. Remove all the layers except for *forest_stands_2012*.
3. Save your project now as `forest_inventory.qgs`

Now you need to create a rectangular grid of points separated 80 meters from each other:

1. Open *Vector* ► *Research Tools* *Regular points*.
2. Press the drop-down button next to the *Input extent* field and from the *Calculate from Layer* menu, select *forest_stands_2012*.
3. In the *Point spacing/count* settings, enter `80 meters`.
4. Check the *Use point spacing* box to indicate that the value represents the distance between the points.
5. Under *Regular points*, save the output as `systematic_plots.shp` in the `forestry\sampling\` folder.
6. Check *Open output file after running algorithm*.
7. Press *Run*.

Примечание: The suggested *Regular points* creates the systematic points starting in the upper-left corner of the extent of the selected polygon layer. If you want to add some randomness to this regular points, you could use a randomly calculated number between 0 and 80 (80 is the distance between our points), and then write it as the *Initial inset from corner (LH side)* parameter in the tool's dialog.

You notice that the tool has used the whole extent of your stands layer to generate a rectangular grid of points. But you are only interested on those points that are actually inside your forest area (see the images below):



1. From the Processing toolbox, open *GDAL* ► *Vector geoprocessing* ► *Clip vector by mask layer*.
2. Select *systematic_plots* as the *Input layer*.

3. Set *forest_stands_2012* as the *Mask layer*.
4. Save the *Clipped (mask)* result as `systematic_plots_clip.shp` in the `forestry\sampling\` folder.
5. Check *Open output file after running algorithm*.
6. Press *Run*.

You have now the points that the field teams will use to navigate to the designed sample plots locations. You can still prepare these points so that they are more useful for the field work. At the least you will have to add meaningful names for the points and export them to a format that can be used in their GPS devices.

Let's start with the naming of the sample plots. If you check the *Attribute table* for the plots inside the forest area, you can see that you have the default *id* field automatically generated by the *Regular points* tool. Label the points to see them in the map and consider if you could use those numbers as part of your sample plot naming:

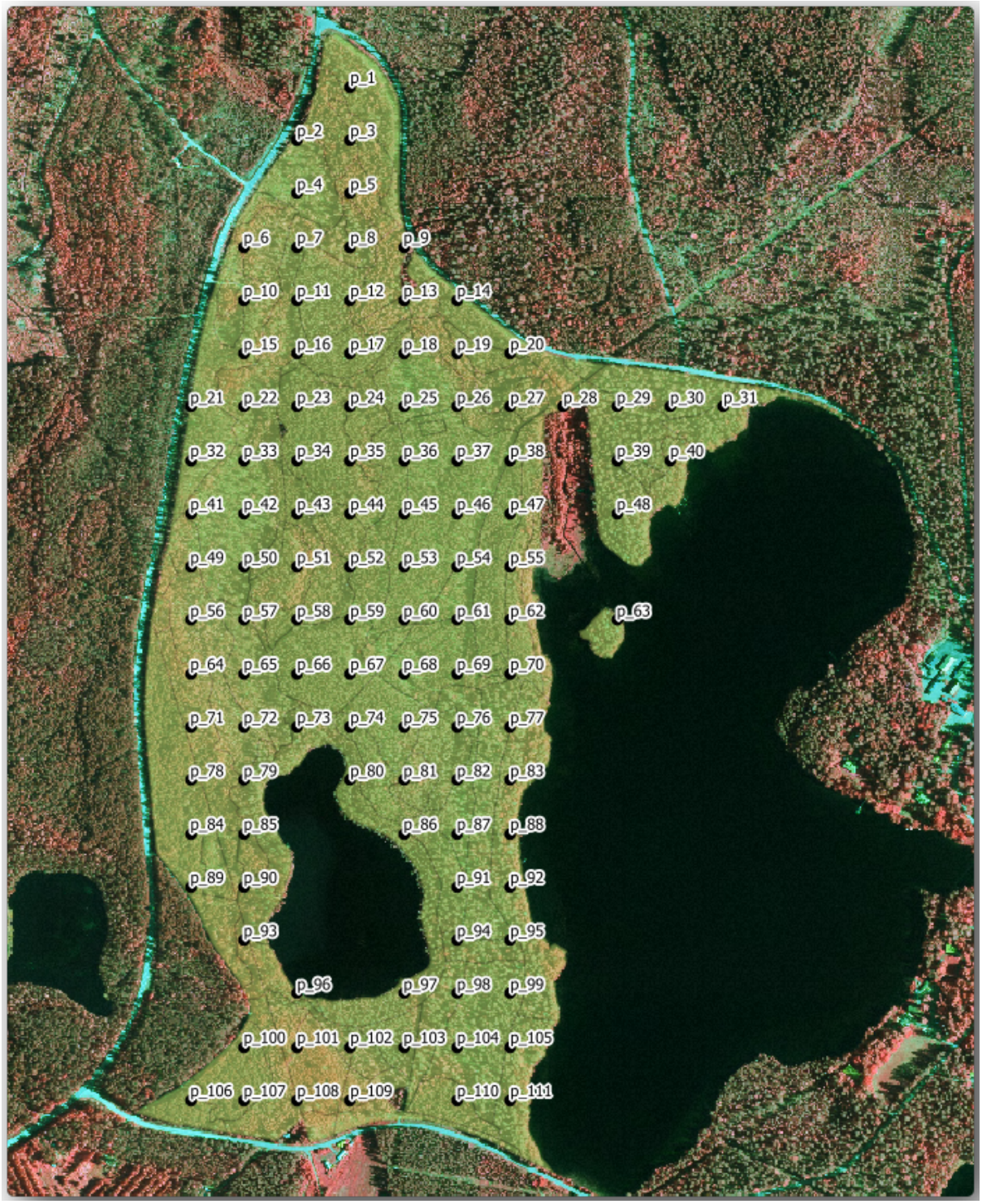
1. Open the *Layer Properties* ► *Labels* for the `systematic_plots_clip` layer.
2. Turn the top menu into *Single Labels*.
3. For the *Value* entry, select the field `id`.
4. Go to the *Buffer* tab, check the *Draw text buffer* and set the buffer *Size* to `1`.
5. Click *OK*.

Now look at the labels on your map. You can see that the points have been created and numbered first West to East and then North to South. If you look at the attribute table again, you will notice that the order in the table is following also that pattern. Unless you would have a reason to name the sample plots in a different way, naming them in a West-East/North-South fashion follows a logical order and is a good option.

Nevertheless, the number values in the `id` field are not so good. It would be better if the naming would be something like `p_1, p_2...`. You can create a new column for the `systematic_plots_clip` layer:

1. Go to the *Attribute table* for `systematic_plots_clip`.
2. Enable the edit mode.
3. Open the Field calculator:
    1. Check *Create a new field*
    2. Enter `Plot_id` as *Output field name*
    3. Set the *Output field type* to *Text (string)*.
    4. In the *Expression* field, write, copy or construct this formula `concat('P_', @rownum )`. Remember that you can also double click on the elements inside the *Function list*. The `concat` function can be found under *String* and `@rownum` is under the *Variables and values* group.
4. Click *OK*.
5. Disable the edit mode and save your changes.

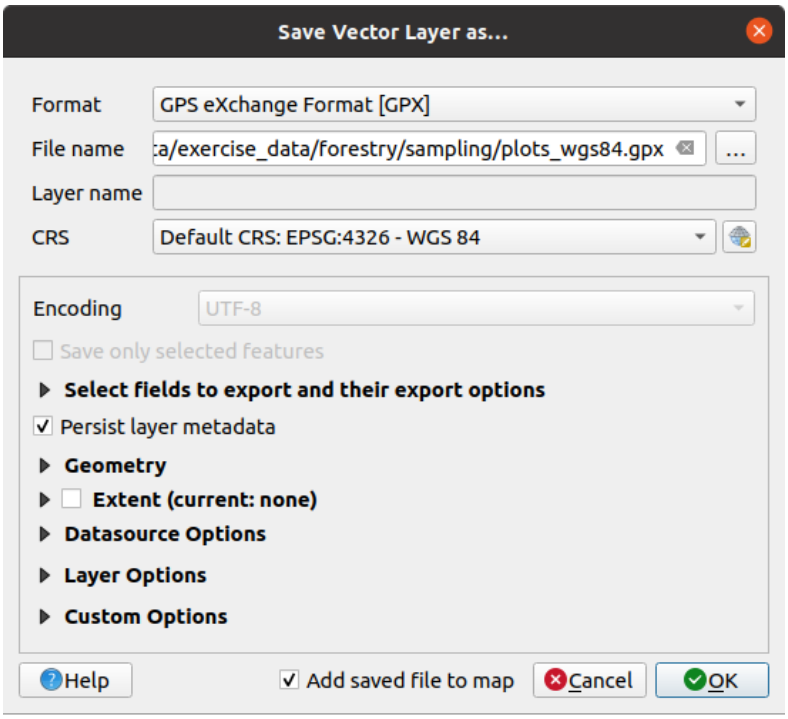
Now you have a new column with plot names that are meaningful to you. For the `systematic_plots_clip` layer, change the field used for labeling to your new `Plot_id` field.


p_1
p_2
p_3
p_4
p_5
p_6
p_7
p_8
p_9
p_10
p_11
p_12
p_13
p_14
p_15
p_16
p_17
p_18
p_19
p_20
p_21
p_22
p_23
p_24
p_25
p_26
p_27
p_28
p_29
p_30
p_31
p_32
p_33
p_34
p_35
p_36
p_37
p_38
p_39
p_40
p_41
p_42
p_43
p_44
p_45
p_46
p_47
p_48
p_49
p_50
p_51
p_52
p_53
p_54
p_55
p_56
p_57
p_58
p_59
p_60
p_61
p_62
p_63
p_64
p_65
p_66
p_67
p_68
p_69
p_70
p_71
p_72
p_73
p_74
p_75
p_76
p_77
p_78
p_79
p_80
p_81
p_82
p_83
p_84
p_85
p_86
p_87
p_88
p_89
p_90
p_91
p_92
p_93
p_94
p_95
p_96
p_97
p_98
p_99
p_100
p_101
p_102
p_103
p_104
p_105
p_106
p_107
p_108
p_109
p_110
p_111

14.5.3 Follow Along: Exporting Sample Plots as GPX format

The field teams will be probably using a GPS device to locate the sample plots you planned. The next step is to export the points you created to a format that your GPS can read. QGIS allows you to save your point and line vector data in GPS eXchange Format (GPX), which is an standard GPS data format that can be read by most of the specialized software. You need to be careful with selecting the CRS when you save your data:

1. Right-click `systematic_plots_clip` layer and select *Export* ► *Save features as...*.



2. In *Format* select *GPS eXchange Format [GPX]*.
3. Save the output *File name* as `plots_wgs84.gpx` in the `forestry\sampling\` folder.
4. In *CRS* select *Selected CRS*.
5. Browse for *EPSG:4326 - WGS 84*.

> **Примечание:** The GPX format accepts only this CRS, if you select a different one, QGIS will give no error but you will get an empty file.

6. Click *OK*.
7. In the dialog that opens, select only the `waypoints` layer (the rest of the layers are empty).

The inventory sample plots are now in a standard format that can be managed by most of the GPS software. The field teams can now upload the locations of the sample plots to their devices. That would be done by using the specific devices own software and the `plots_wgs84.gpx` file you just saved. Other option would be to use the *GPS Tools* plugin but it would most likely involve setting the tool to work with your specific GPS device. If you are working with your own data and want to see how the tool works you can find out information about it in the section working_gps in the **QGIS User Manual**.

Save your QGIS project now.

14.5.4 In Conclusion

You just saw how easily you can create a systematic sampling design to be used in a forest inventory. Creating other types of sampling designs will involve the use of different tools within QGIS, spreadsheets or scripting to calculate the coordinates of the sample plots, but the general idea remains the same.

14.5.5 What's Next?

In the next lesson you will see how to use the Atlas capabilities in QGIS to automatically generate detailed maps that the field teams will be using to navigate to the sample plots assigned to them.

14.6 Lesson: Creating Detailed Maps with the Atlas Tool

The systematic sampling design is ready and the field teams have loaded the GPS coordinates in their navigation devices. They also have a field data form where they will collect the information measured at every sample plot. To easier find their way to every sample plot, they have requested a number of detail maps where some ground information can be clearly seen along with a smaller subset of sample plots and some information about the map area. You can use the Atlas tool to automatically generate a number of maps with a common format.

The goal for this lesson: Learn to use the Atlas tool in QGIS to generate detailed printable maps to assist in the field inventory work.

14.6.1  Follow Along: Preparing the Print Layout

Before we can automate the detailed maps of the forest area and our sampling plots, we need to create a map template with all the elements we consider useful for the field work. Of course the most important will be a properly styled but, as you have seen before, you will also need to add lots of other elements that complete the printed map.

Open the QGIS project from the previous lesson `forest_inventory.qgs`. You should have at least the following layers:

- `forest_stands_2012` (with a 50% transparency, green fill and darker green border lines).
- `systematic_plots_clip`.
- `rautjarvi_aerial`.

Save the project with a new name, `map_creation.qgs`.

To create a printable map, remember that you use the *Layout Manager*:

- Open *Project* ► *Layout Manager…*.
- In the *Layout manager* dialog.
- Click the *Add* button and name your print layout `forest_map`.
- Click *OK*.
- Click the *Show* button.

Set up the printer options so that your maps will suit your paper and margins, for an A4 paper:

- Open menuselection:*Layout –> Page Setup…*.
- *Size* is *A4 (217 x 297 mm)*.
- *Orientation* is *Landscape*.
- *Margins (millimeters)* are all set to `5`.

In the *Print Layout* window, go to the *Composition* tab (on the right panel) and make sure that these settings for *Paper and quality* are the same you defined for the printer:

- *Size*: `A4 (210x297mm)`.
- *Orientation*: `Landscape`.
- *Quality*: `300dpi`.

Composing a map is easier if you make use of the canvas grid to position the different elements. Review the settings for the layout grid:

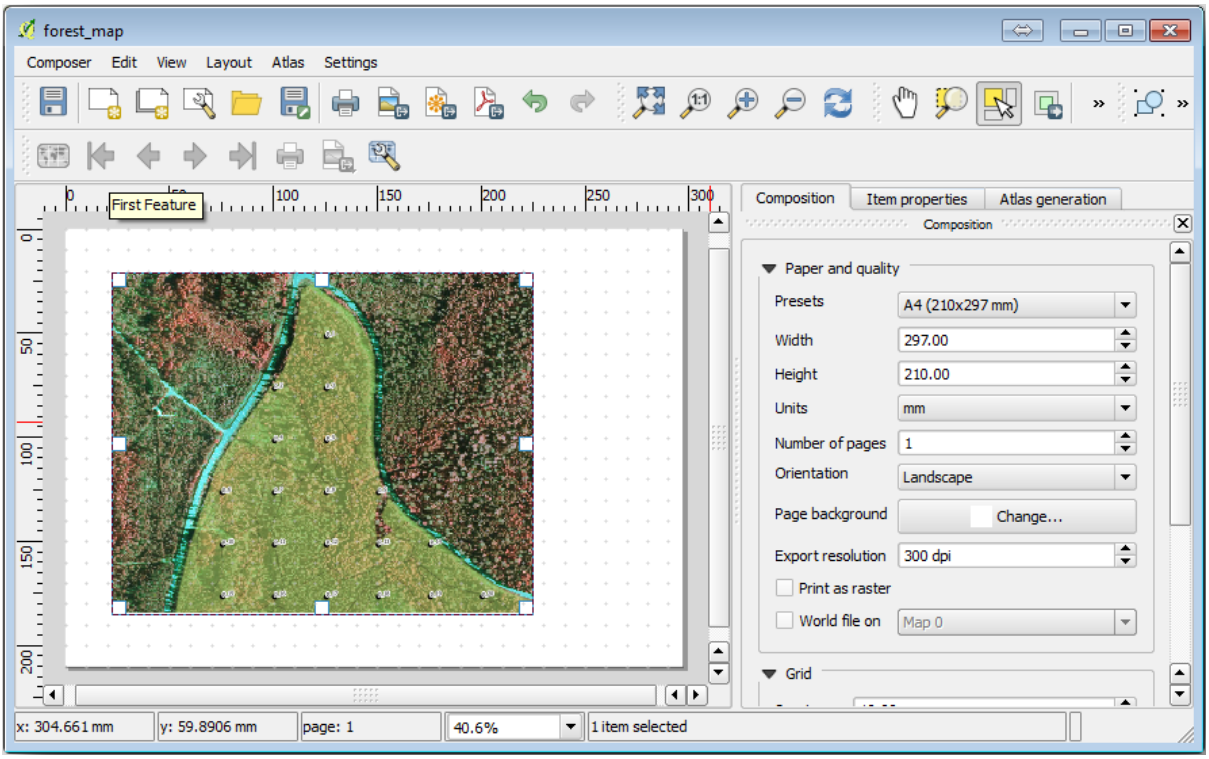
- In the *Composition* tab expand the *Grid* region.
- Check that *Spacing* is set to `10 mm`.
- And that *Tolerance* is set to `2 mm`.

You need to activate the use of the grid:

- Open the *View* menu.
- Check *Show grid*.
- Check *Snap to grid*.
- Notice that options for using *guides* are checked by default, which allows you to see red guiding lines when you are moving elements in the layout.

Now you can start to add elements to your layout. Add first a map element so you can review how the map canvas looks as you will be making changes in the layers symbology:

- Click on the  Add Map button.
- Click and drag a box on the canvas so that the map occupies most of it.



Notice how the mouse cursor snaps to the canvas grid. Use this function when you add other elements. If you want to have more accuracy, change the grid *Spacing* setting. If for some reason you don't want to snap to the grid at some point, you can always check or uncheck it in the *View* menu.

14.6.2 Follow Along: Adding Background Map

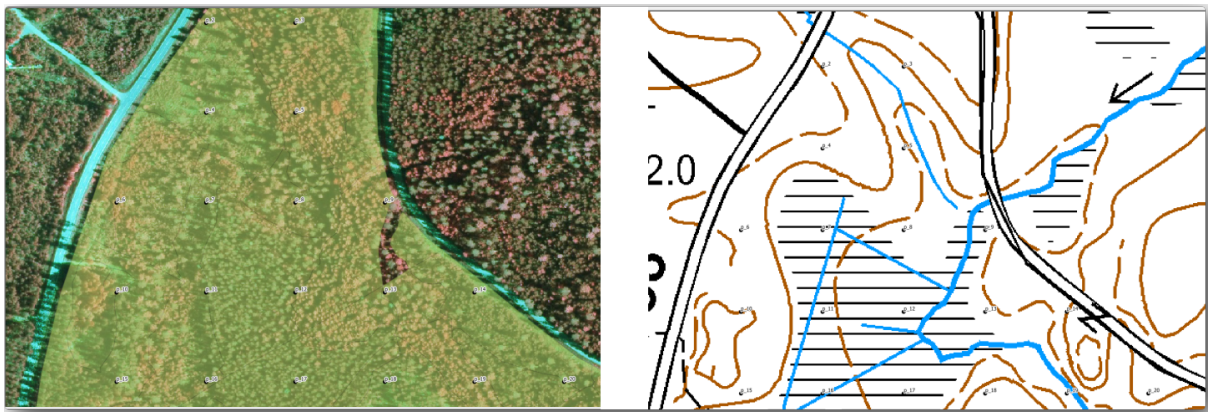
Leave the layout open but go back to the map. Lets add some background data and create some styling so that the map content is as clear as possible.

- Add the background raster `basic_map.tif` that you can find in the `exercise_data\forestry\` folder.
- When prompted select the `ETRS89 / ETRS-TM35FIN` CRS for the raster.

As you can see the background map is already styled. This type of ready to use cartography raster is very common. It is created from vector data, styled in a standard format and stored as a raster so that you don't have to bother styling several vector layers and worrying about getting a good result.

- Now zoom to your sample plots, so that you can see only about four or five lines of plots.

The current styling of the sample plots is not the best, but how does it look in the print layout?:



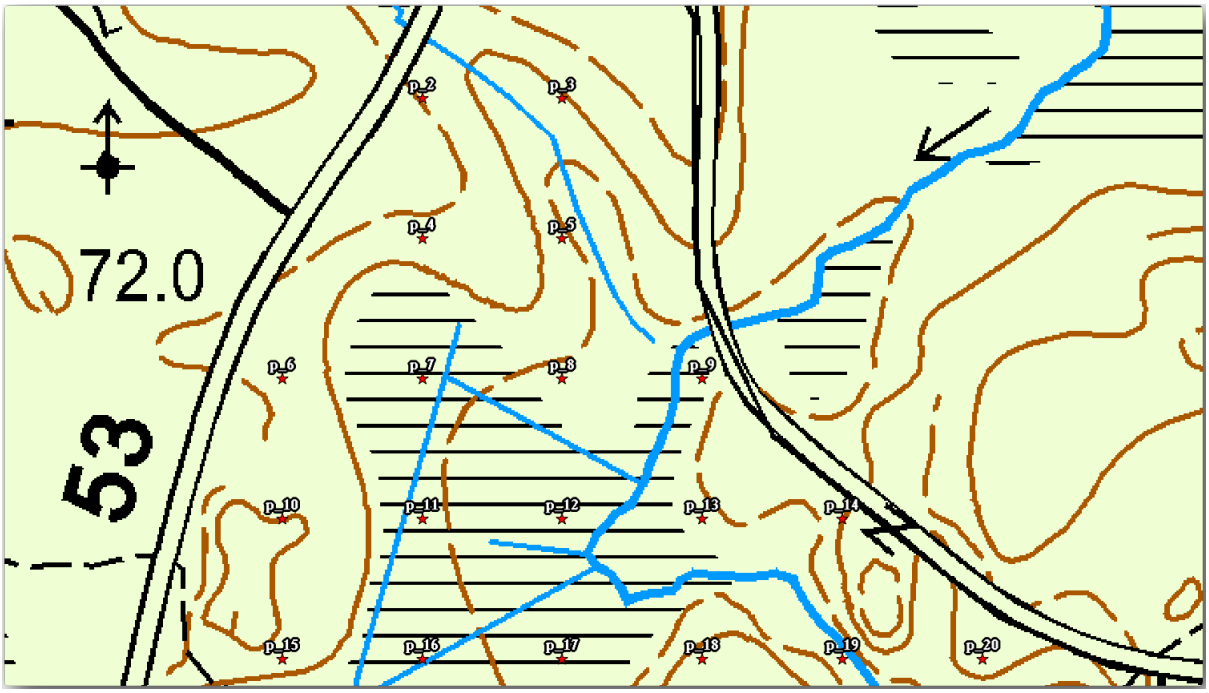
While during the last exercises, the white buffer was OK on top of the aerial image, now that the background image is mostly white you barely can see the labels. You can also check how it looks like on the layout:

- Go to the *Print Layout* window.
- Use the  button to select the map element in the layout.
- Go to the *Item properties* tab.
- Under *Extents* click on *Set to map canvas extent*.
- If you need to refresh the element, under *Main properties* click on the *Update preview*.

Obviously this is not good enough, you want to make the plot numbers as clearly visible as possible for the field teams.

14.6.3 Try Yourself Changing the Symbology of the Layers

You have been working in *Модуль: Создание и изучение базовой карты* with symbology and in *Модуль: Классификация векторных данных* with labeling. Go back to those modules if you need to refresh about some of the available options and tools. Your goal is to get the plots locations and their name to be as clearly visible as possible but always allowing to see the background map elements. You can take some guidance from this image:

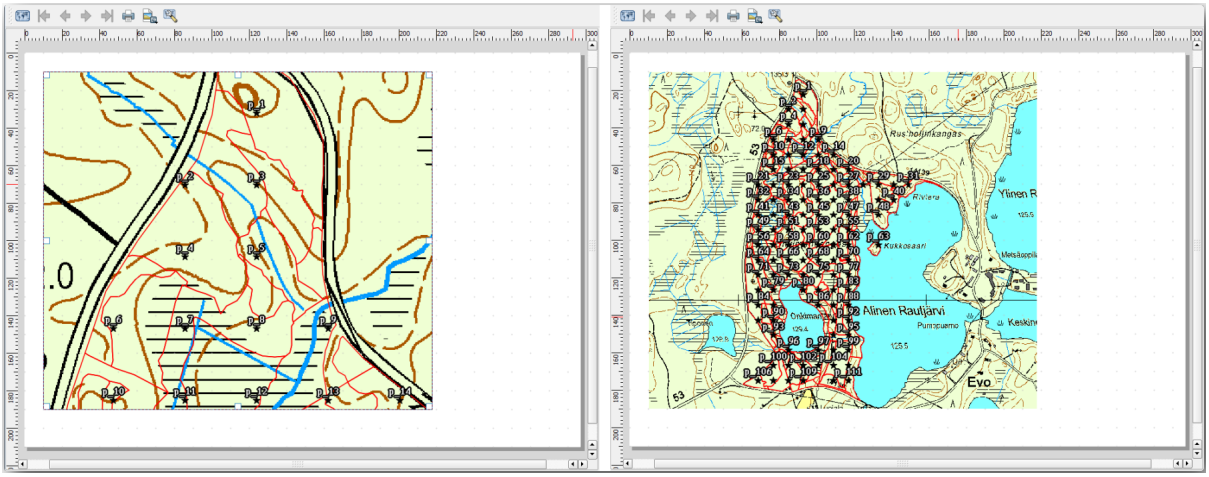


You will use later the the green styling of the `forest_stands_2012` layer. In order to keep it, and have a visualization of it that shows only the stand borders:

- Right click on `forest_stands_2012` and select *Duplicate*
- you get a new layer named `forest_stands_2012 copy` that you can use to define a different style, for example with no filling and red borders.

Now you have two different visualizations of the forest stands and you can decide which one to display for your detail map.

Go back to the *Print Layout* window often to see what the map would look like. For the purposes of creating detailed maps, you are looking for a symbology that looks good not at the scale of the whole forest area (left image below) but at a closer scale (right image below). Remember to use *Update preview* and *Set to map canvas extent* whenever you change the zoom in your map or the layout.

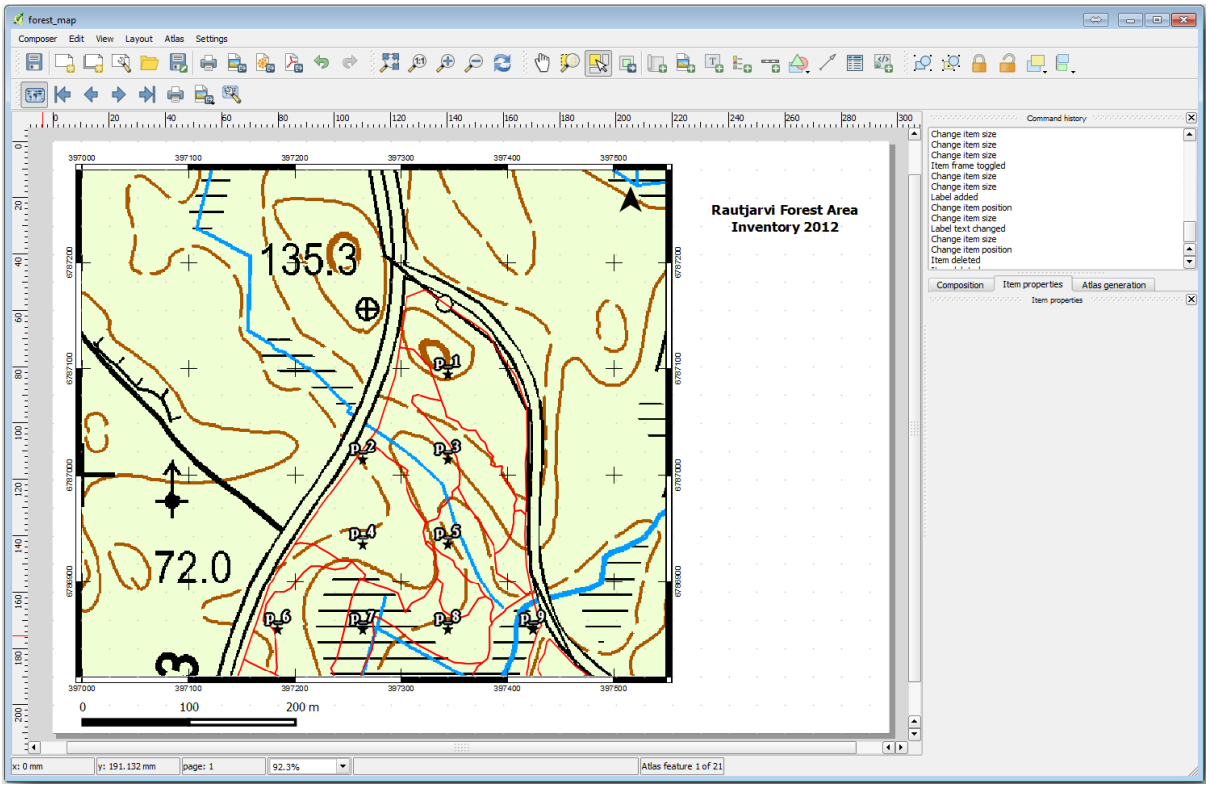


14.6.4 Try Yourself Create a Basic Map Template

Once you have a symbology your happy with, you are ready to add some more information to your printed map. Add at least the following elements:

- Title.
- A scale bar.
- Grid frame for your map.
- Coordinates on the sides of the grid.

You have created a similar composition already in *Модуль: Составление базовой карты*. Go back to that module as you need. You can look at this example image for reference:



Export your map as an image and look at it.

- *Layout* ► *Export as Image...*.
- Use for example the `JPG format`.

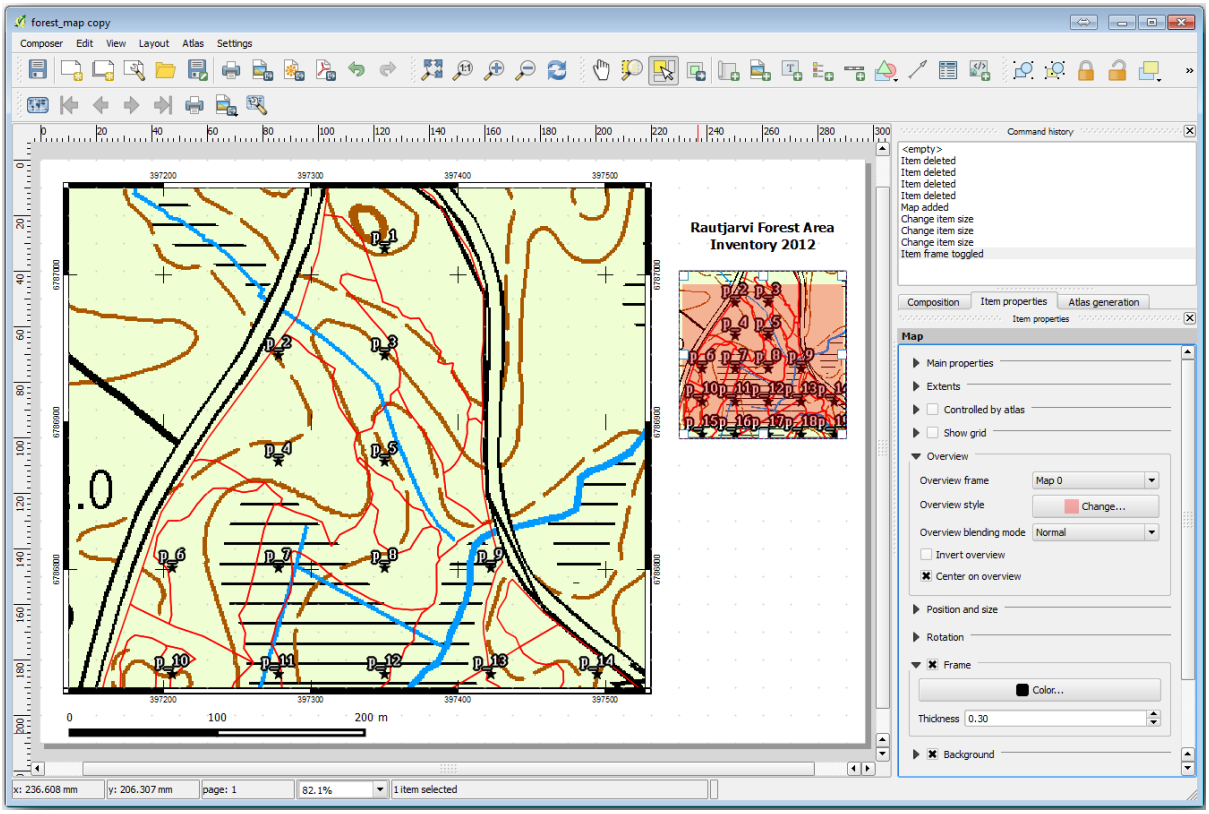
That is what it will look like when printed.

14.6.5 Follow Along: Adding More Elements to the Print Layout

As you probably noticed in the suggested map template images, there are plenty of room on the right side of the canvas. Lets see what else could go in there. For the purposes of our map, a legend is not really necessary, but an overview map and some text boxes could add value to the map.

The overview map will help the field teams place the detail map inside the general forest area:

- Add another map element to the canvas, right under the title text.
- In the *Item properties* tab, open the *Overview* dropdown.
- Set the *Overview frame* to *Map 0*. This creates a shadowed rectangle over the smaller map representing the extent visible in the bigger map.
- Check also the *Frame* option with a black color and a *Thickness* of `0.30`.



Notice that your overview map is not really giving an overview of the forest area which is what you want. You want this map to represent the whole forest area and you want it to show only the background map and the `forest_stands_2012` layer, and not display the sample plots. And also you want to lock its view so it does not change anymore whenever you change the visibility or order of the layers.

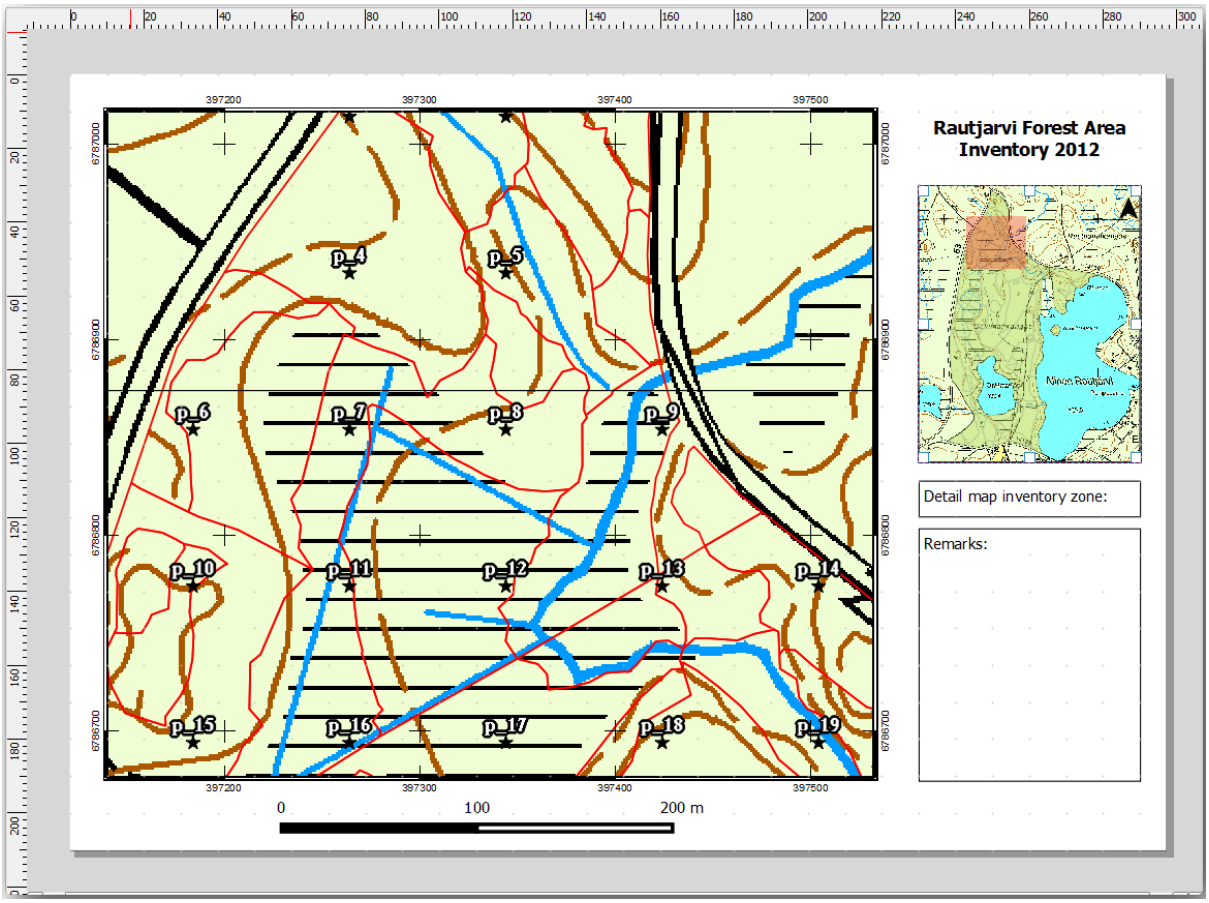
- Go back to the map, but don't close the *Print Layout*.
- Right click the `forest_stands_2012` layer and click on *Zoom to Layer Extent*.
- Deactivate all layers except for `basic_map` and `forest_stands_2012`.
- Go back to the *Print Layout*.
- With the small map selected, click the *Set to map canvas extent* to set its extents to what you can see in the map window.
- Lock the view for the overview map by checking *Lock layers for map item* under *Main properties*.

Now your overview map is more what you expected and its view will not change anymore. But, of course, now your detail map is not showing anymore the stand borders nor the sample plots. Lets fix that:

- Go to the map window again and select the layers you want to be visible (`systematic_plots_clip`, `forest_stands_2012 copy` and `Basic_map`).
- Zoom again to have only a few lines of sample plots visible.
- Go back to the *Print Layout* window.
- Select the bigger map in your layout ().
- In *Item properties* click on *Update preview* and *Set to map canvas extent*.

Notice that only the bigger map is displaying the current map view, and the small overview map is keeping the same view you had when you locked it.

Note also that the overview is showing a shaded frame for the extent shown in the detail map.



Your template map is almost ready. Add now two text boxes below the map, one containing the text „Detailed map zone: „ and the other one „Remarks: „. Place them as you can see in the image above.

You can also add a North arrow to the overview map:

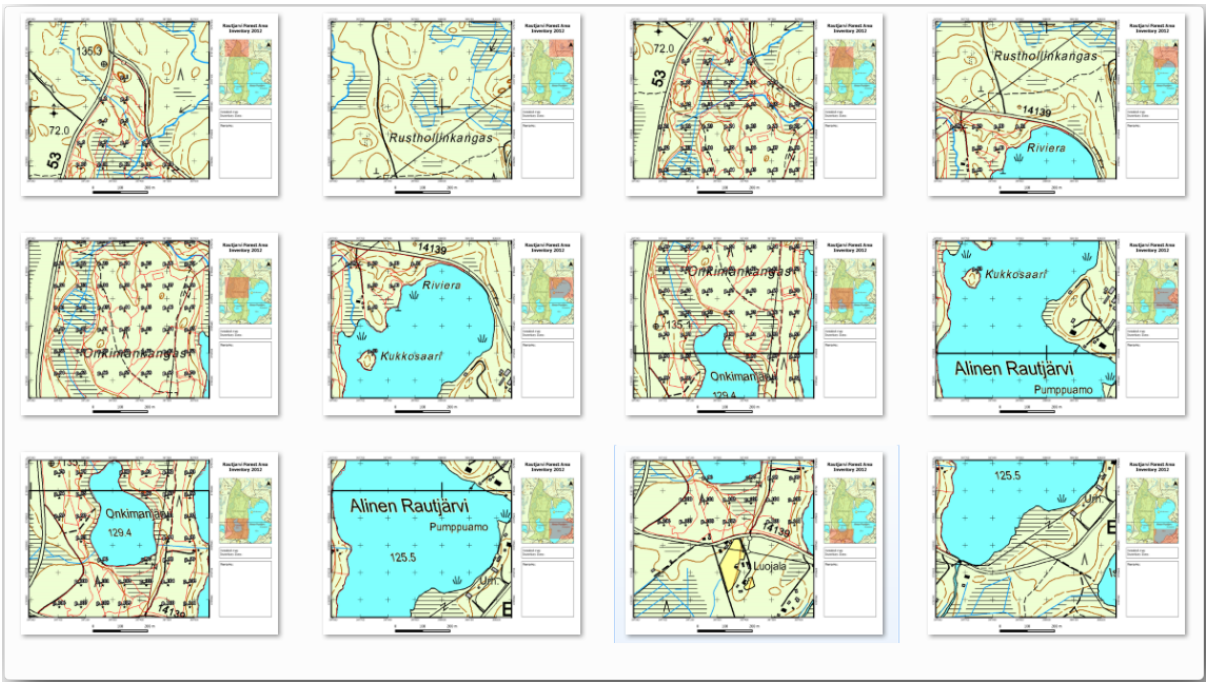
- Use the *Add image* tool, .
- Click at the upper right corner of the overview map.
- In *Item properties* open *Search directories* and browse for an arrow image.
- Under *Image rotation*, check the *Sync with map* and select `Map 1` (the overview map).
- Uncheck *Background*.

- Resize the arrow image to a size that looks good on the small map.

The basic map layout is ready, now you want to make use of the Atlas tool to generate as many detail maps in this format as you consider necessary.

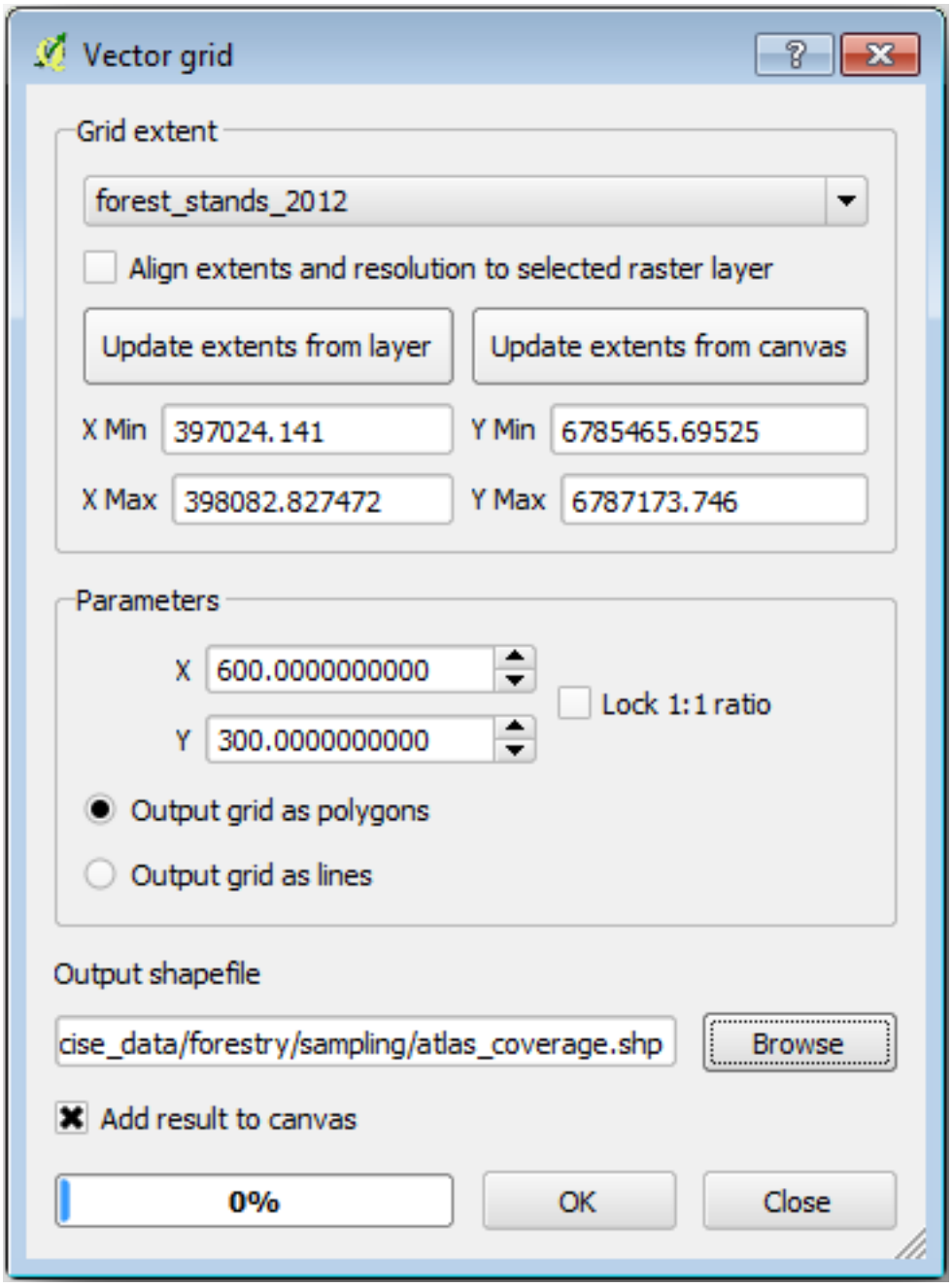
14.6.6 Follow Along: Creating an Atlas Coverage

The Atlas coverage is just a vector layer that will be used to generate the detail maps, one map for every feature in the coverage. To get an idea of what you will do next, here is a full set of detail maps for the forest area:



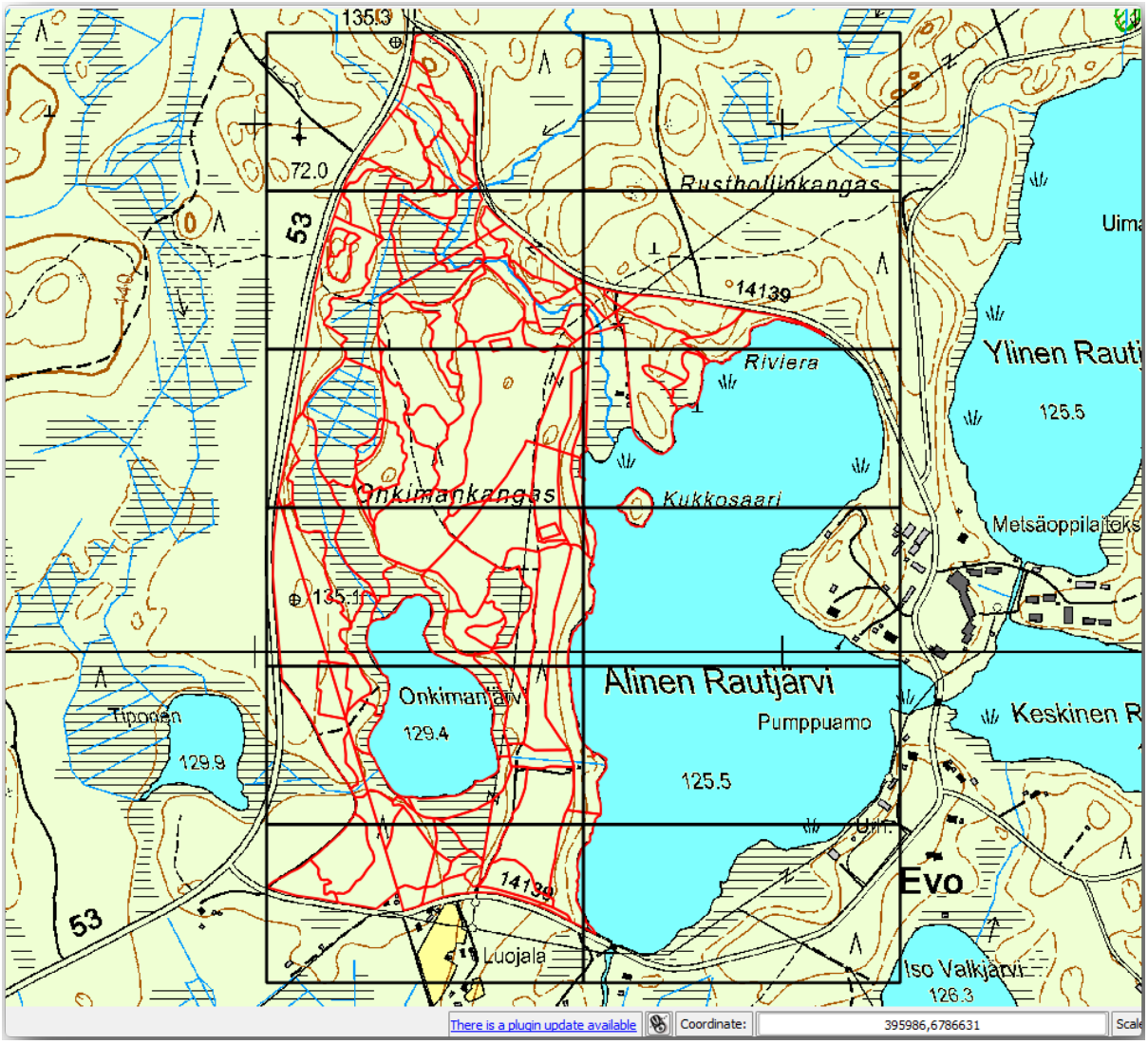
The coverage could be any existing layer, but usually it makes more sense to create one for the specific purpose. Let's create a grid of polygons covering the forest area:

- In the QGIS map view, open *Vector* ► *Research Tools* ► *Vector grid*.
- Set the tool as shown in this image:



- Save the output as `atlas_coverage.shp`.
- Style the new `atlas_coverage` layer so that the polygons have no filling.

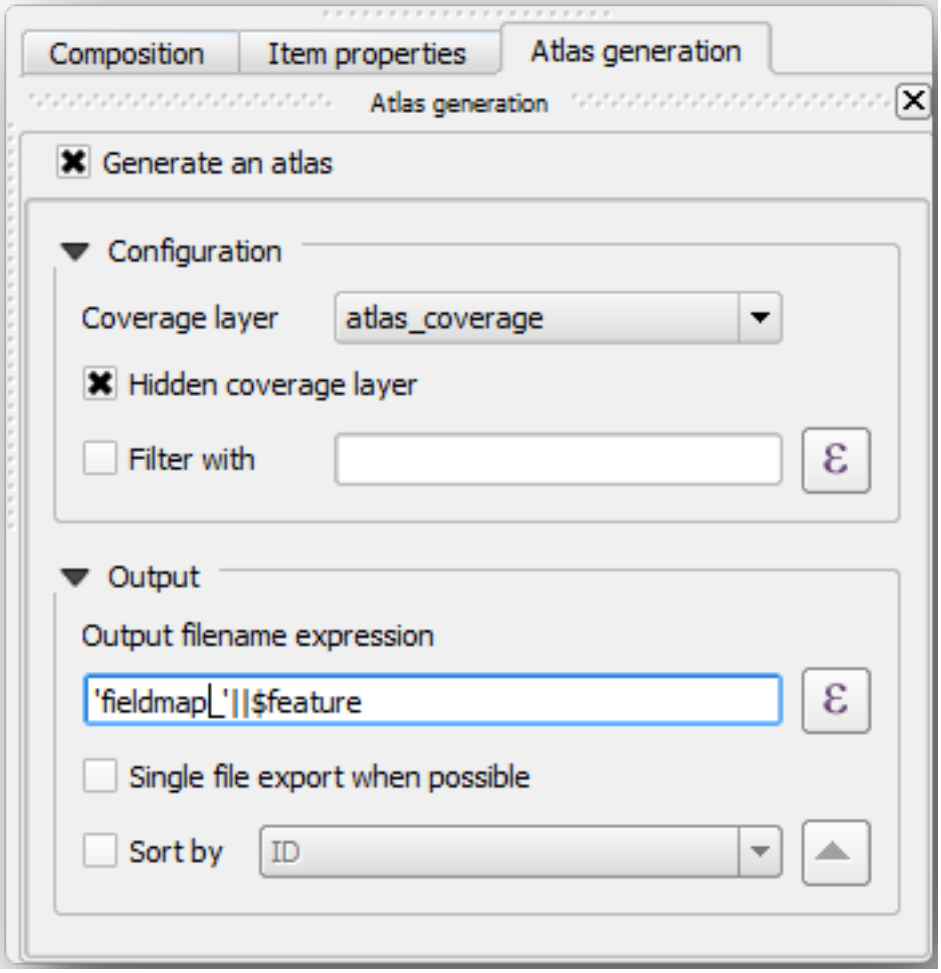
The new polygons are covering the whole forest area and they give you an idea of what each map (created from each polygon) will contain.



14.6.7 Follow Along: Setting Up the Atlas Tool

The last step is to set up the Atlas tool:

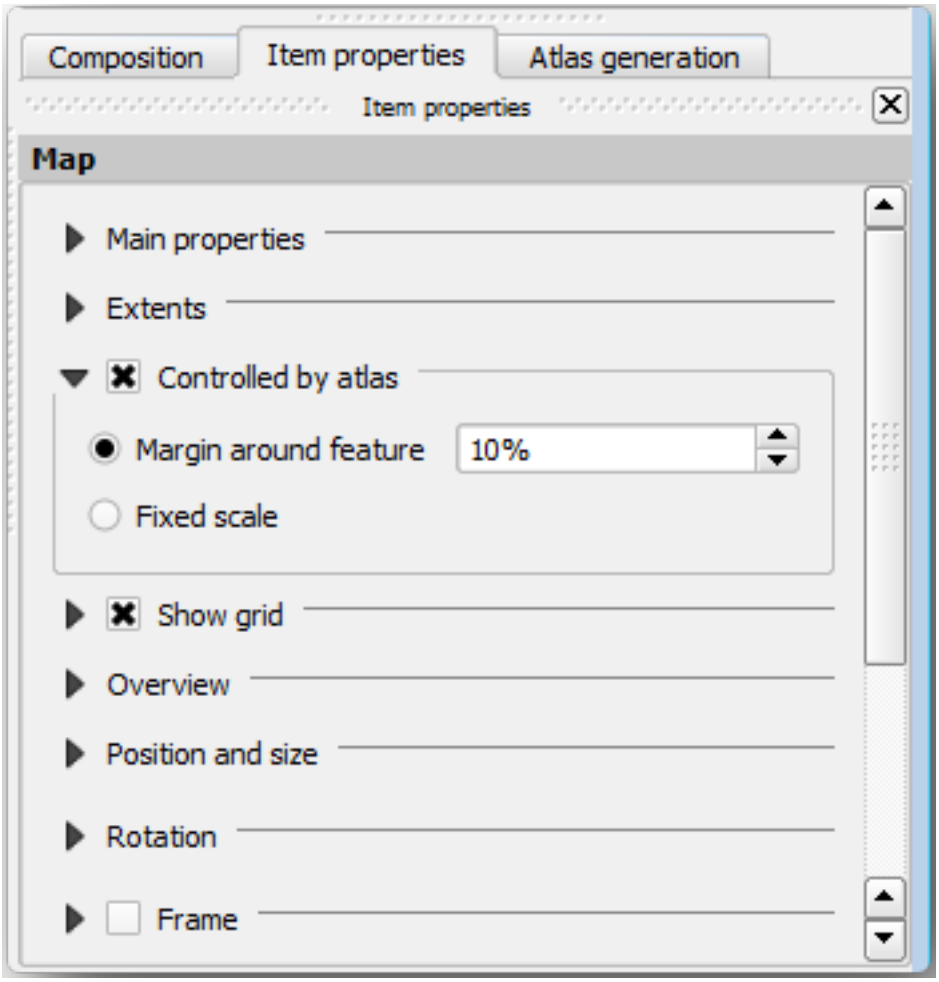
- Go back to the *Print Layout*.
- In the panel on the right, go to the *Atlas generation* tab.
- Set the options as follows:



That tells the Atlas tool to use the features (polygons) inside `atlas_coverage` as the focus for every detail map. It will output one map for every feature in the layer. The *Hidden coverage layer* tells the Atlas to not show the polygons in the output maps.

One more thing needs to be done. You need to tell the Atlas tool what map element is going to be updated for every output map. By now, you probably can guess that the map to be changed for every feature is the one you have prepared to contain detail views of the sample plots, that is the bigger map element in your canvas:

- Select the bigger map element.
- Go to the *Item properties* tab.
- In the list, check *Controlled by atlas*.
- And set the *Marging around feature* to `10%`. The view extent will be 10% bigger than the polygons, which means that your detail maps will have a 10% overlap.



Now you can use the preview tool for Atlas maps to review what your maps will look like:

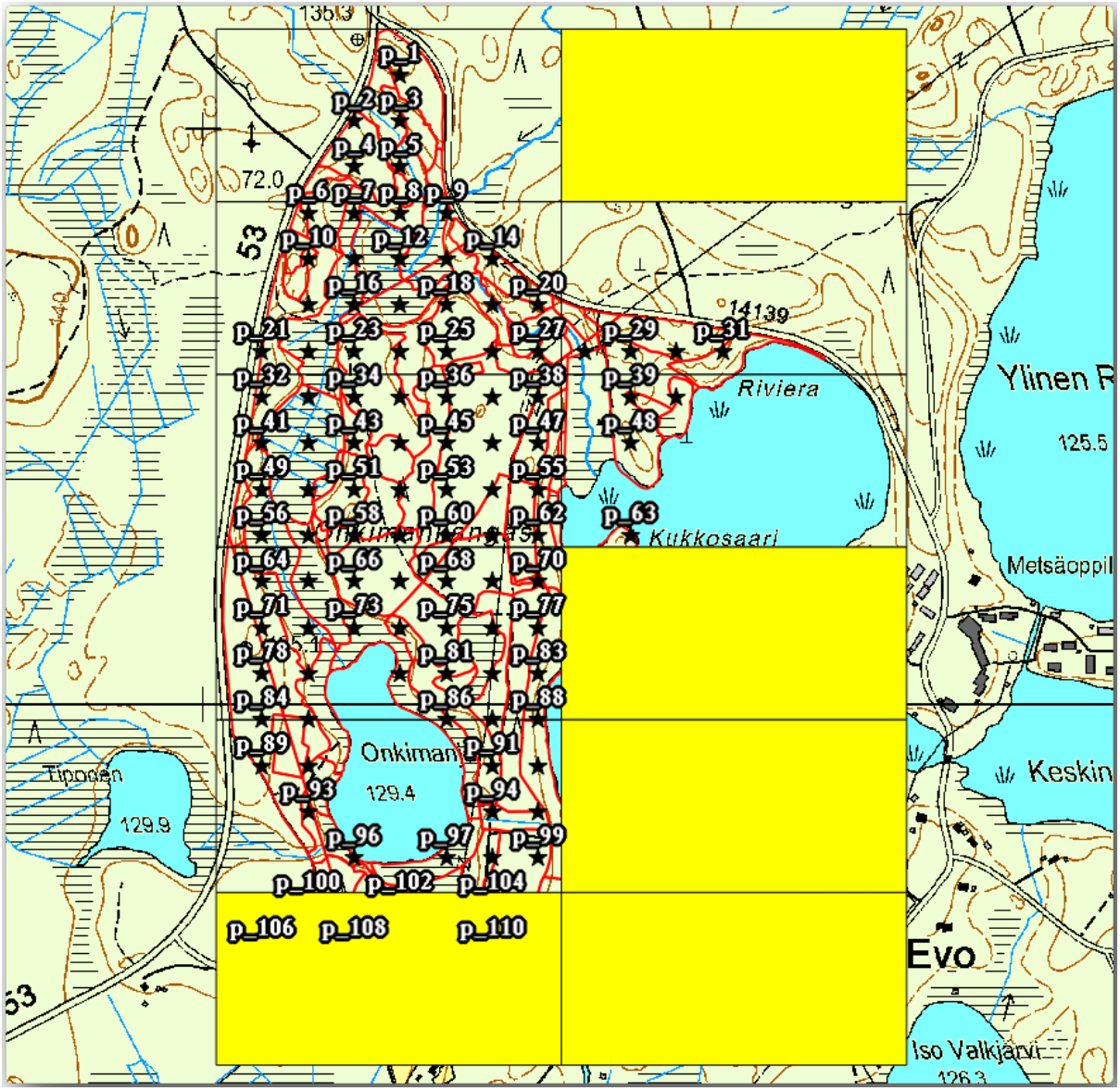
- Activate the Atlas previews using the button or if your Atlas toolbar is not visible, via *Atlas* ► *Preview Atlas*.
- You can use the arrows in the Atlas tool bar or in the *Atlas* menu to move through maps that will be created.

Note that some of them cover areas that are not interesting. Lets do something about it and save some trees by not printing those useless maps.

14.6.8 Follow Along: Editing the Coverage Layer

Besides removing the polygons for those areas that are not interesting, you can also customize the text labels in your map to be generated with content from the *Attribute table* of your coverage layer:

- Go back to the map view.
- Enable editing for the `atlas_coverage` layer.
- Select the polygons that are selected (in yellow) in the image below.
- Remove the selected polygons.
- Disable editing and save the edits.



You can go back to the *Print Layout* and check that the previews of the Atlas use only the polygons you left in the layer.

The coverage layer you are using does not yet have useful information that you could use to customize the content of the labels in your map. The first step is to create them, you can add for example a zone code for the polygon areas and a field with some remarks for the field teams to have into account:

- Open the *Attribute table* for the `atlas_coverage` layer.
- Enable editing.
- Use the  calculator to create and populate the following two fields.
- Create a field named `Zone` and type `Whole number (integer)`.
- In the *Expression* box write/copy/construct `$rownum`.
- Create another field named `Remarks`, of type `Text (string)` and a width of `255`.
- In the *Expression* box write `'No remarks.'`. This will set all the default value for all the polygons.

The forest manager will have some information about the area that might be useful when visiting the area. For example, the existence of a bridge, a swamp or the location of a protected species. The `atlas_coverage` layer

is probably in edit mode still, add the following text in the `Remarks` field to the corresponding polygons (double click the cell to edit it):

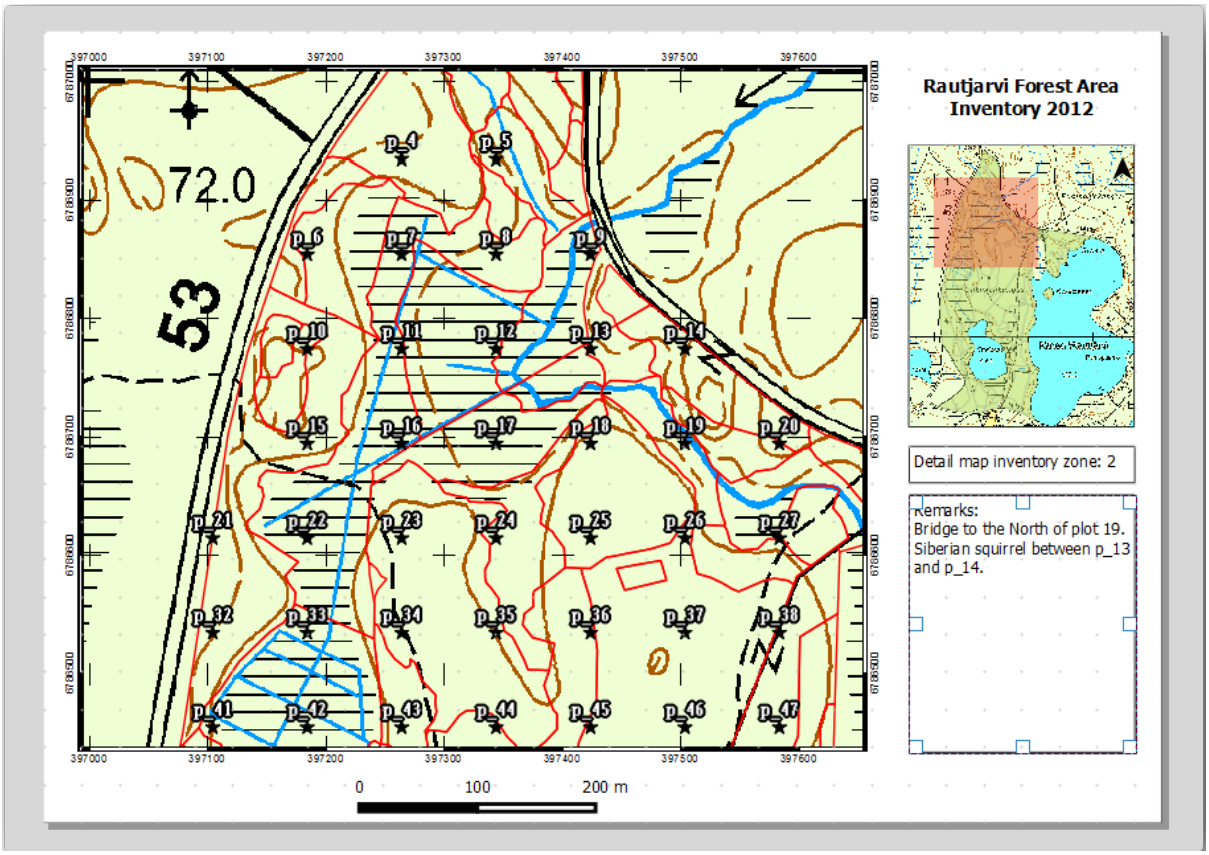
- For the `Zone` 2: `Bridge to the North of plot 19. Siberian squirrel between p_13 and p_14.`.
- For the `Zone` 6: `Difficult to transit in swamp to the North of the lake.`.
- For the `Zone` 7: `Siberian squirrel to the South East of p_94.`.
- Disable editing and save your edits.

Almost ready, now you have to tell the Atlas tool that you want some of the text labels to use the information from the `atlas_coverage` layer's attribute table.

- Go back to the *Print Layout*.
- Select the text label containing `Detailed map...`.
- Set the *Font* size to `12`.
- Set the cursor at the end of the text in the label.
- In the *Item properties* tab, inside the *Main properties* click on *Insert or Edit an Expression*....
- In the *Function list* double click on the field `Zone` under *Field and Values*.
- Click *OK*.
- The text inside the box in the *Item properties* should show `Detail map inventory zone: [% "Zone" %]`. Note that the `[% "Zone" %]` will be substituted by the value of the field `Zone` for the corresponding feature from the layer `atlas_coverage`.

Test the contents of the label by looking at the different Atlas preview maps.

Do the same for the labels with the text `Remarks:` using the field with the zone information. You can leave a break line before you enter the expression. You can see the result for the preview of zone 2 in the image below:



Use the Atlas preview to browse through all the maps you will be creating soon and enjoy!

14.6.9 Follow Along: Printing the Maps

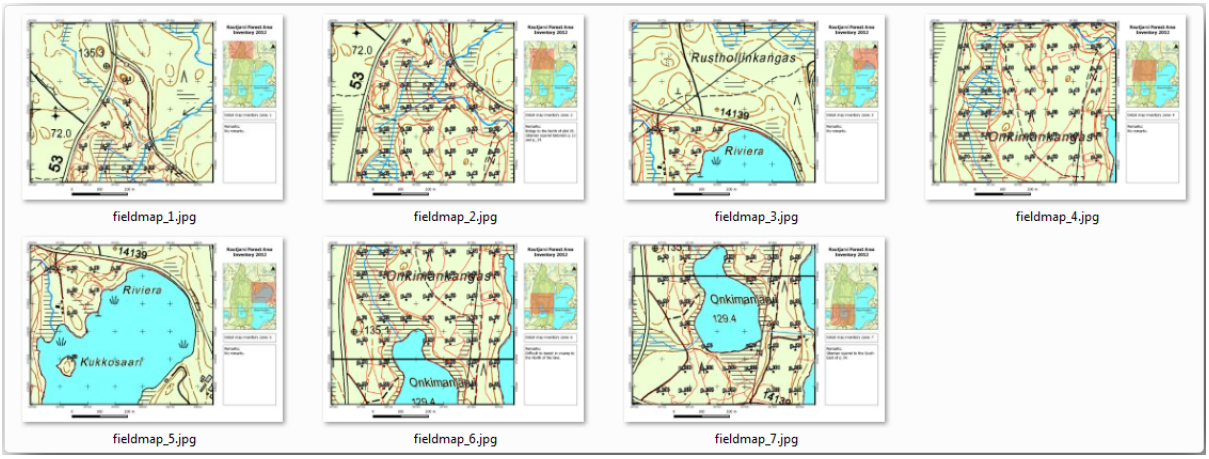
Last but not least, printing or exporting your maps to image files or PDF files. You can use the *Atlas* ► *Export Atlas as Images…* or *Atlas* ► *Export Atlas as PDF…*. Currently the SVG export format is not working properly and will give a poor result.

Lets print the maps as a single PDF that you can send to the field office for printing:

- Go to the *Atlas generation* tab on the right panel.
- Under the *Output* check the *Single file export when possible*. This will put all the maps together into a PDF file, if this option is not checked you will get one file for every map.
- Open *Layout* ► *Export as PDF…*.
- Save the PDF file as `inventory_2012_maps.pdf` in your `exercise_data\forestry\samplig\map_creation\` folder.

Open the PDF file to check that everything went as expected.

You could just as easily create separate images for every map (remember to uncheck the single file creation), here you can see the thumbnails of the images that would be created:



In the *Print Layout*, save your map as a layout template as `forestry_atlas.qpt` in your `exercise_data\forestry\map_creation\` folder. Use *Layout* ► *Save as Template*. You will be able to use this template again and again.

Close the *Print Layout* and save your QGIS project.

14.6.10 In Conclusion

You have managed to create a template map that can be used to automatically generate detail maps to be used in the field to help navigate to the different plots. As you noticed, this was not an easy task but the benefit will come when you need to create similar maps for other regions and you can use the template you just saved.

14.6.11 What's Next?

In the next lesson, you will see how you can use LiDAR data to create a DEM and then use it to your enhance your data and maps visibility.

14.7 Lesson: Calculating the Forest Parameters

Estimating the parameters of the forest is the goal of the forest inventory. Continuing the example from previous lesson, you will use the inventory information gathered in the field to calculate the forest parameters, for the whole forest first, and then for the stands you digitized before.

The goal for this lesson: Calculate forest parameters at general and stand level.

14.7.1  Follow Along: Adding the Inventory Results

The field teams visited the forest and with the help of the information you provided, gathered information about the forest at every sample plot.

Most often the information will be collected into paper forms in the field, then typed to a spreadsheet. The sample plots information has been condensed into a `.csv` file that can be easily open in QGIS.

Continue with the QGIS project from the lesson about designing the inventory, you probably named it `forest_inventory.qgs`.

First, add the sample plots measurements to your QGIS project:

1. Go to *Layer* ► *Add Delimited Text Layer…*.
2. Browse to the file `systematic_inventory_results.csv` located in `exercise_data/forestry/results/`.
3. Make sure that the *Point coordinates* option is checked.
4. Set the fields for the coordinates to the *X* and *Y* fields.
5. Click *OK*.
6. When prompted, select *ETRS89 / ETRS-TM35FIN* as the CRS.
7. Open the new layer's *Attribute table* and have a look at the data.

You can read the type of data that is contained in the sample plots measurements in the text file `legend_2012_inventorydata.txt` located in the `exercise_data/forestry/results/` folder.

The `systematic_inventory_results` layer you just added is actually just a virtual representation of the text information in the `.csv` file. Before you continue, convert the inventory results to a real spatial dataset:

1. Right click on the `systematic_inventory_results` layer.
2. Browse to `exercise_data/forestry/results/` folder.
3. Name the file `sample_plots_results.shp`.
4. Check *Add saved file to map*.
5. Remove the `systematic_inventory_results` layer from your project.

14.7.2 Follow Along: Whole Forest Parameters Estimation

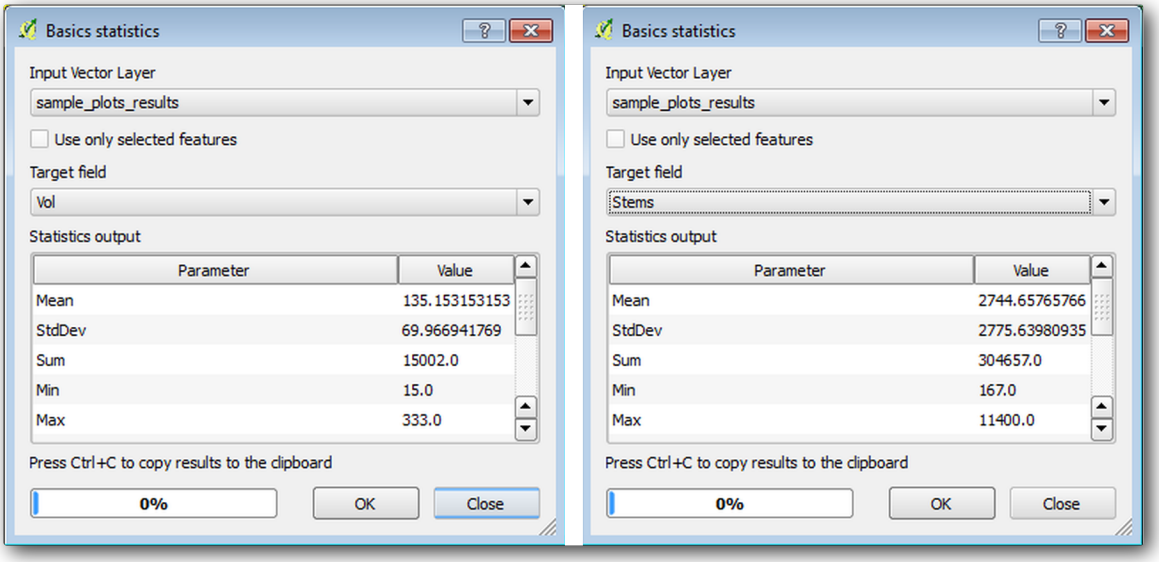
You can calculate the averages for this whole forest area from the inventory results for the some interesting parameters, like the volume and the number of stems per hectare. Since the systematic sample plots represent equal areas, you can directly calculate the averages of the volumes and number of stems per hectare from the `sample_plots_results` layer.

You can calculate the average of a field in a vector layer using the *Basic statistics* tool:

1. Open *Vector* ► *Analysis Tools* ► *Basic statistics for Fields*.
2. Select `sample_plots_results` as the *Input Vector Layer*.
3. Select `Vol` as *Target field*.
4. Click *OK*.

The average volume in the forest is `135.2 m3/ha`.

You can calculate the average for the number of stems in the same way, `2745 stems/ha`.



14.7.3 Follow Along: Estimating Stand Parameters

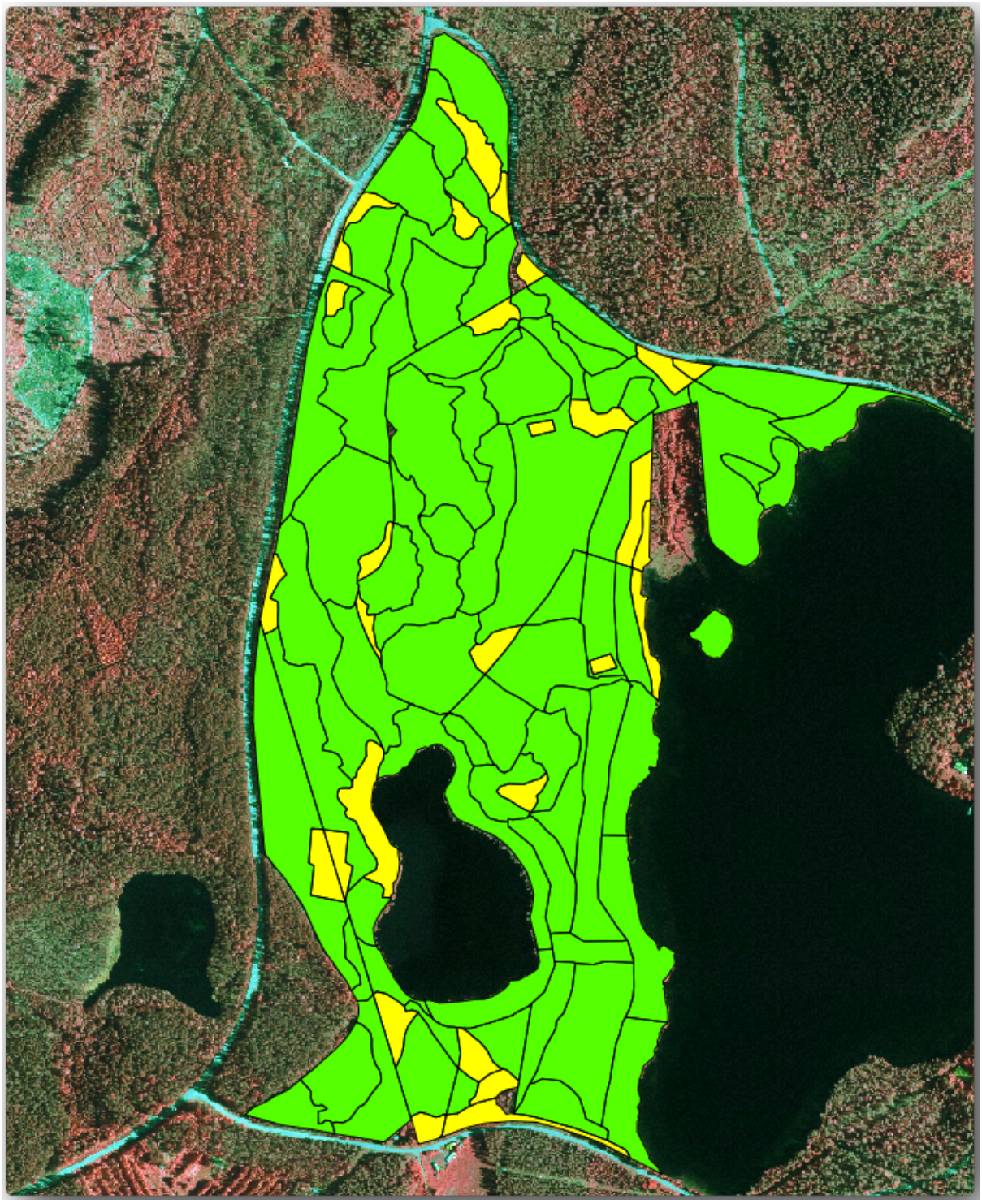
You can make use of those same systematic sample plots to calculate estimates for the different forest stands you digitized previously. Some of the forest stands did not get any sample plot and for those you will not get information. You could have planned some extra sample plots when you planned the systematic inventory, so that the field teams would have measured a few extra sample plots for this purpose. Or you could send a field team later to get estimates of the missing forest stands to complete the stand inventory. Nevertheless, you will get information for a good number of stands just using the planned plots.

What you need is to get the averages of the sample plots that are falling within each of the forest stands. When you want to combine information based on their relative locations, you perform a spatial join:

1. Open the *Vector* ► *Data Management* ► *Join attributes by location* tool.
2. Set `forest_stands_2012` as the *Target vector layer*. The layer you want the results for.
3. Set `sample_plots_results` as the *Join vector layer*. The layer you want to calculate estimates from.
4. Check *Take summary of intersecting features*.

5. Check to calculate only the *Mean*.
6. Name the result as `forest_stands_2012_results.shp` and save it in the `exercise_data/forestry/results/` folder.
7. Finally select *Keep all records...*, so you can check later what stands did not get information.
8. Click *OK*.
9. Accept adding the new layer to your project when prompted.
10. Close the *Join attributes by location* tool.

Open the *Attribute table* for *forest_stands_2012_results* and review the results you got. Note that a number of forest stands have `NULL` as the value for the calculations, those are the ones having no sample plots. Select them all and view them in the map, they are some of the smaller stands:



Lets calculate now the same averages for the whole forest as you did before, only this time you will use the averages you got for the stands as the bases for the calculation. Remember that in the previous situation, each sample plot represented a theoretical stand of `80x80 m`. Now you have to consider the area of each of the stands individually instead. That way, again, the average values of the parameters that are in, for example, m3/ha for the volumes are converted to total volumes for the stands.

You need to first calculate the areas for the stands and then calculate total volumes and stem numbers for each of them:

1. In the *Attribute table* enable editing.
2. Open the *Field calculator*.

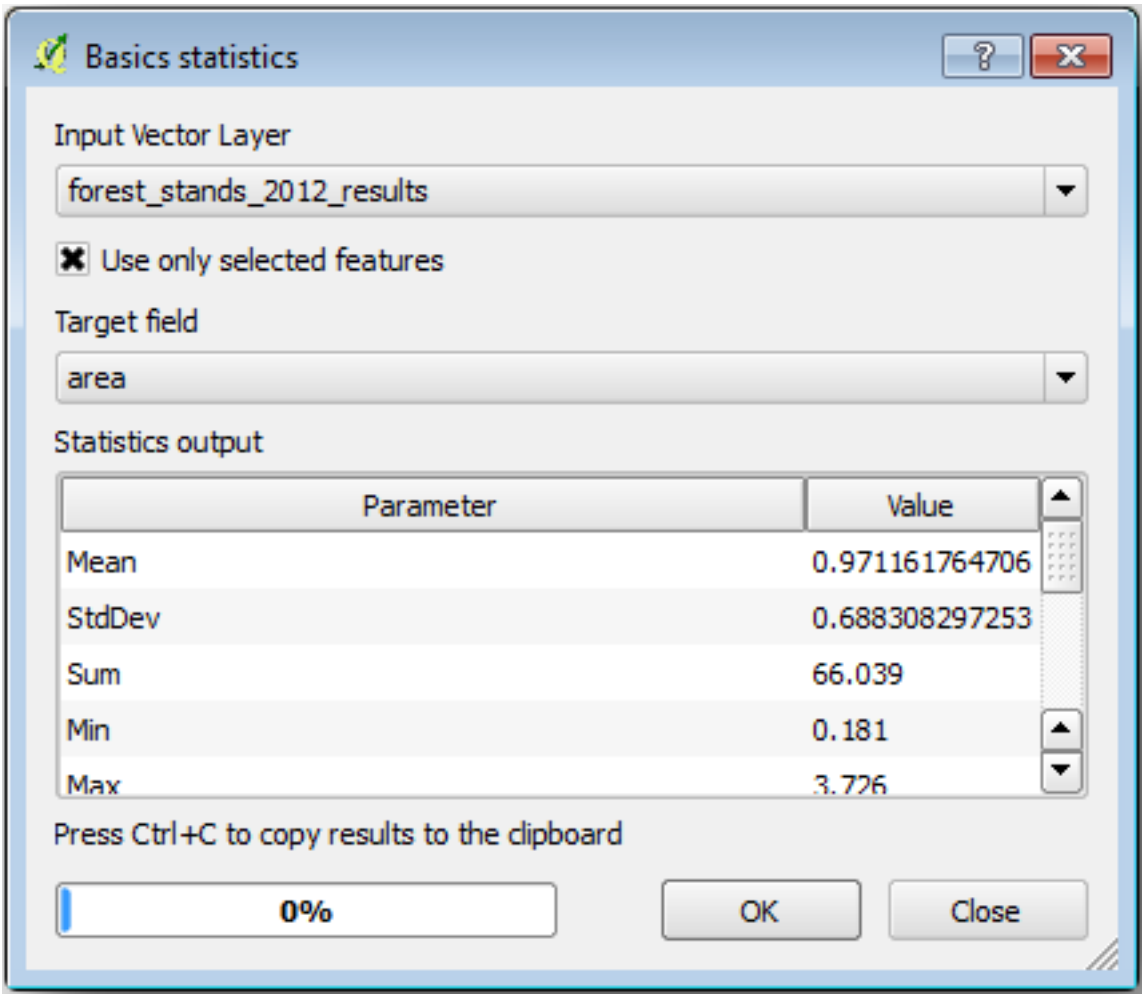
3. Create a new field called `area`.
4. Set the *Output field type* to `Decimal number (real)`.
5. Set the *Precision* to `2`.
6. In the *Expression* box, write `$area / 10000`. This will calculate the area of the forest stands in ha.
7. Click *OK*.

Now calculate a field with the total volumes and number of stems estimated for every stand:

1. Name the fields `s_vol` and `s_stem`.
2. The fields can be integer numbers or you can use real numbers also.
3. Use the expressions `"area" * "MEANVol"` and `"area" * "MEANStems"` for total volumes and total stems respectively.
4. Save the edits when you are finished.
5. Disable editing.

In the previous situation, the areas represented by every sample plot were the same, so it was enough to calculate the average of the sample plots. Now to calculate the estimates, you need to divide the sum of the stands volumes or number of stems by the sum of the areas of the stands containing information.

1. In the *Attribute table* for the `forest_stands_2012_results` layer, select all the stands containing information.
2. Open *Vector* ► *Analysis Tools* ► *Basic statistics for fields*.
3. Select the `forest_stands_2012_results` as the *Input layer*.
4. Select `area` as *Field to calculate statistics on*.
5. Check the *Selected features only*
6. Click *OK*.



As you can see, the total sum of the stands“ areas is `66.04 ha`. Note that the area of the missing forest stands is only about `7 ha`.

In the same way, you can calculate that the total volume for these stands is `8908 m3/ha` and the total number of stems is `179594 stems`.

Using the information from the forest stands, instead of directly using that from the sample plots, gives the following average estimates:

- `184.9 m3/ha` and
- `2719 stems/ha`.

Save your QGIS project, `forest_inventory.qgs`.

14.7.4 In Conclusion

You managed to calculate forest estimates for the whole forest using the information from your systematic sample plots, first without considering the forest characteristics and also using the interpretation of the aerial image into forest stands. And you also got some valuable information about the particular stands, which could be used to plan the management of the forest in the coming years.

14.7.5 What's Next?

In the following lesson, you will first create a hillshade background from a LiDAR dataset which you will use to prepare a map presentation with the forest results you just calculated.

14.8 Lesson: DEM from LiDAR Data

You can improve the look of your maps by using different background images. You could use the basic map or the aerial image you have been using before, but a hillshade raster of the terrain will look nicer in some situations.

You will use LAStools to extract a DEM from a LiDAR dataset and then create a hillshade raster to use in your map presentation later.

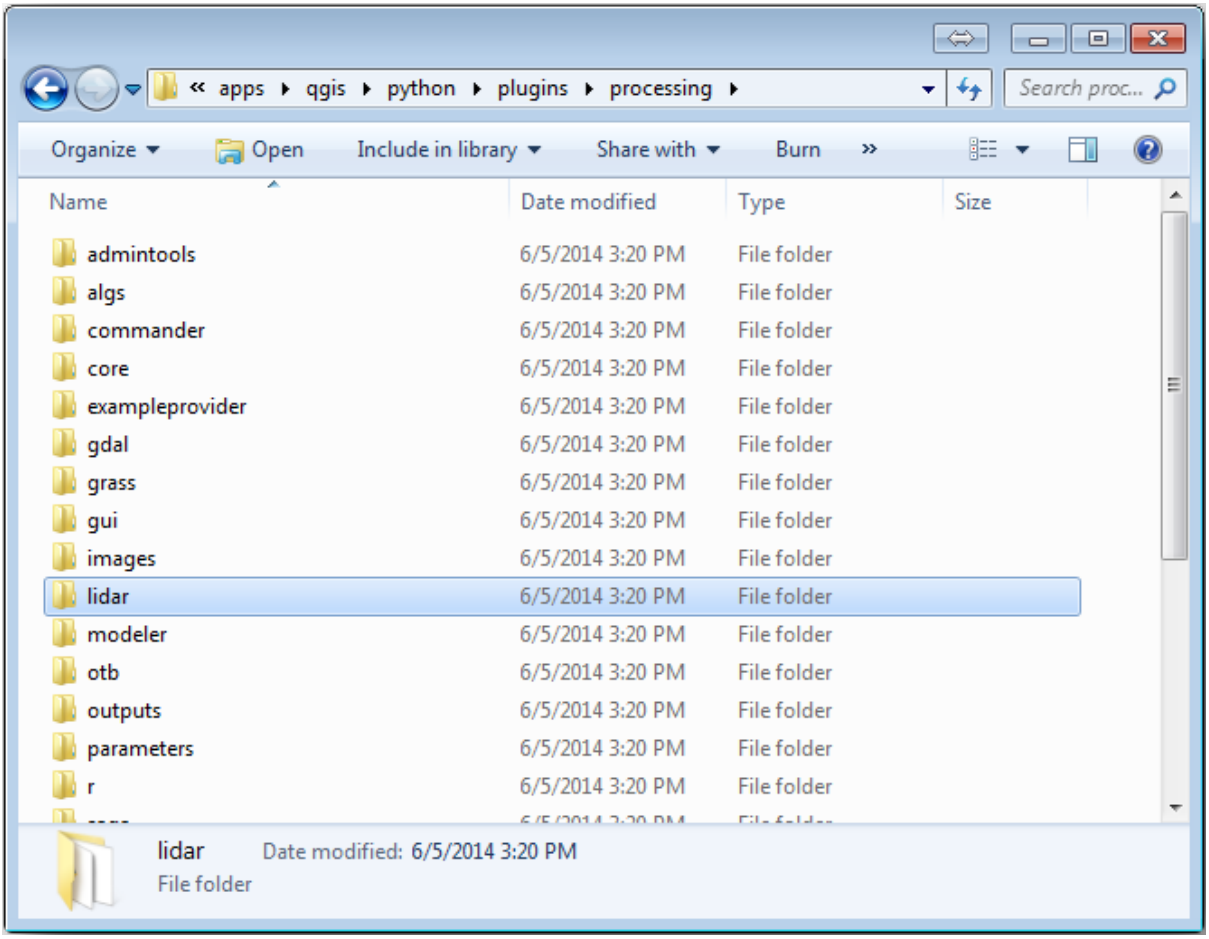
The goal for this lesson: Install LAStools and calculate a DEM from LiDAR data and a hillshade raster.

14.8.1  Follow Along: Installing Lastools

Managing LiDAR data within QGIS is possible using the Processing framework and the algorithms provided by LAStools.

You can obtain a digital elevation model (DEM) from a LiDAR point cloud and then create a hillshade raster that is visually more intuitive for presentation purposes. First you will have to set up the *Processing* framework settings to properly work with LAStools:

- Close QGIS, if you have already started it.
- An old lidar plugin might be installed by default in your system in the folder `C:/Program Files/QGIS Valmiera/apps/qgis/python/plugins/processing/`.
- If you have a folder named `lidar`, delete it. This is valid for some installations of QGIS 2.2 and 2.4.



- Go to the `exercise_data\forestry\lidar\` folder, there you can find the file `QGIS_2_2_toolbox.zip`. Open it and extract the `lidar` folder to replace the one you just deleted.
- If you are using a different QGIS version, you can see more installation instructions in this tutorial.

Now you need to install the LAStools to your computer. Get the newest *lastools* version here and extract the content of the `lastools.zip` file into a folder in your system, for example, `C:\lastools\`. The path to the `lastools` folder cannot have spaces or special characters.

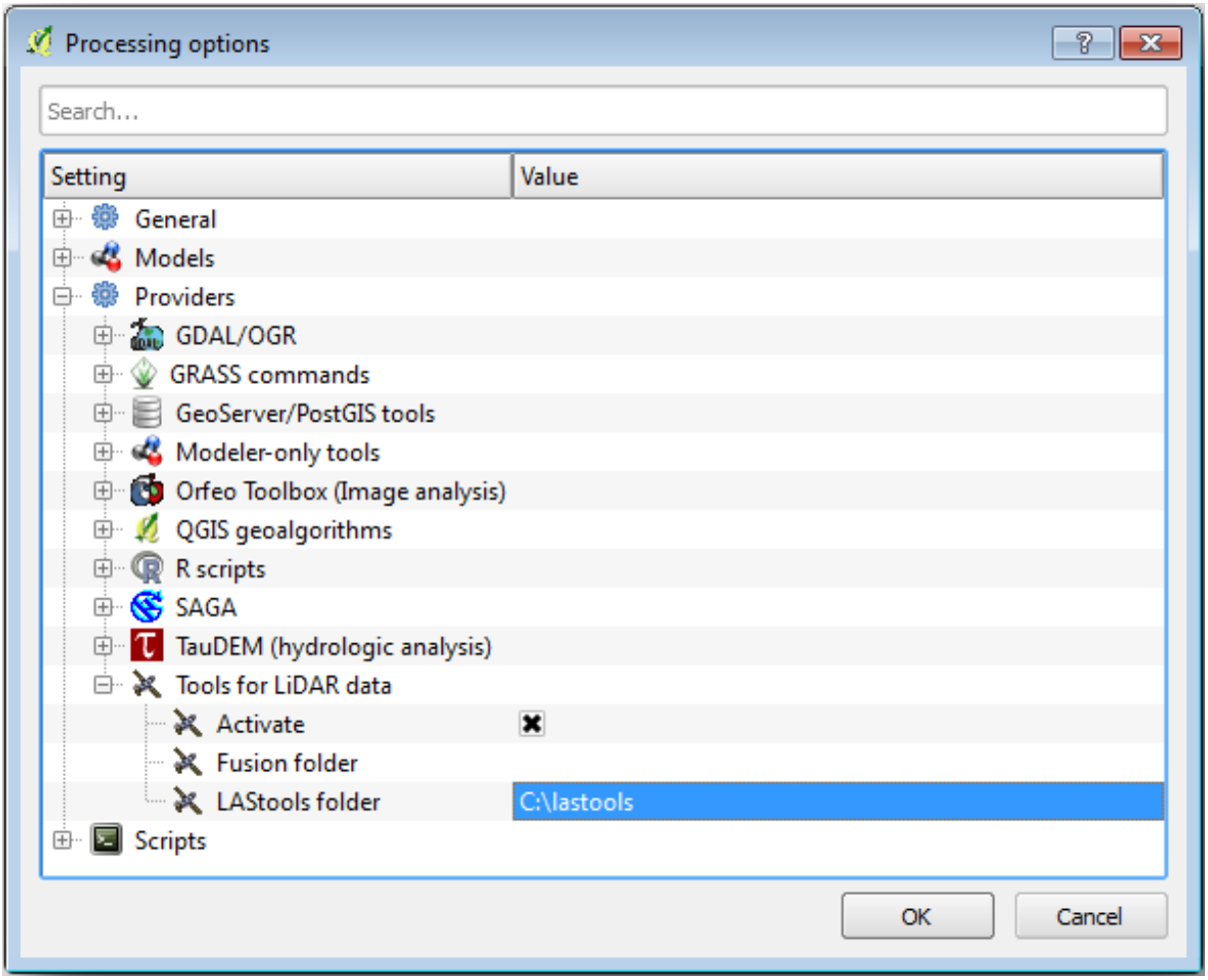
Примечание: Read the `LICENSE.txt` file inside the `lastools` folder. Some of the LAStools are open source and other are closed source and require licensing for most commercial and governmental use. For education and evaluation purposes you can use and test LAStools as much as you need to.

The plugin and the actual algorithms are now installed in your computer and almost ready to use, you just need to set up the Processing framework to start using them:

- Open a new project in QGIS.
- Set the project's CRS to `ETRS89 / ETRS-TM35FIN`.
- Save the project as `forest_lidar.qgs`.

To setup the LAStools in QGIS:

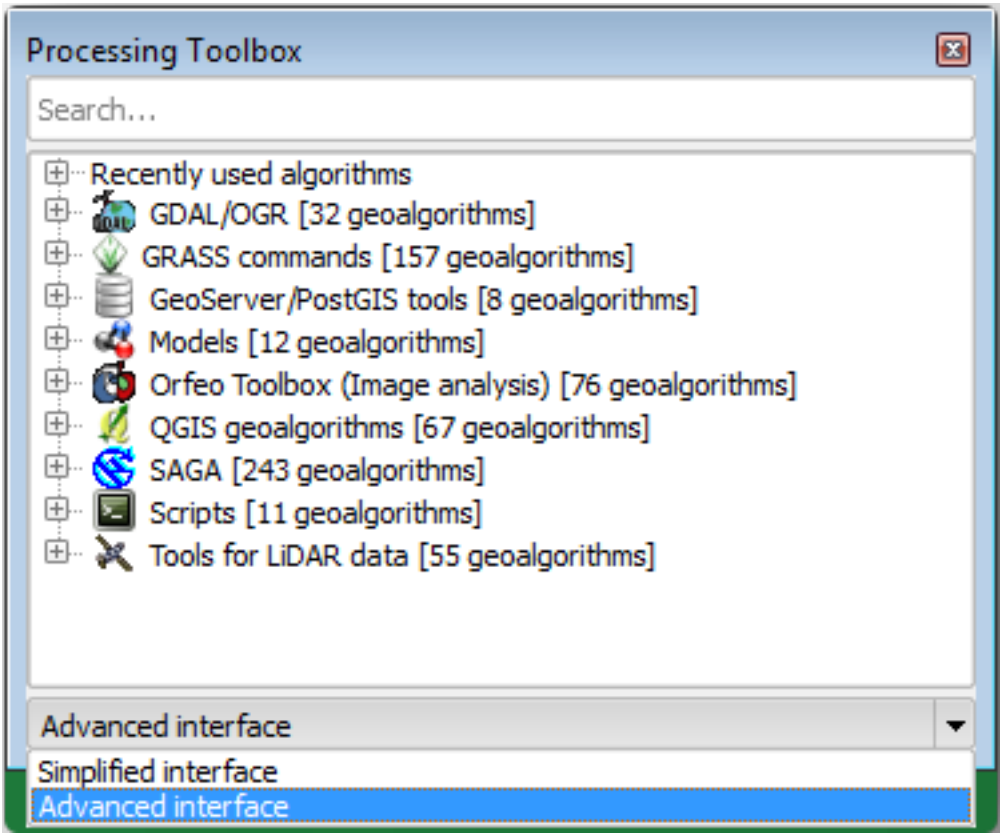
- Go to *Processing* ► *Options and configuration*.
- In the *Processing options* dialog, go to *Providers* and then to *Tools for LiDAR data*.
- Check *Activate*.
- For *LAStools folder* set `c:\lastools\` (or the folder you extracted LAStools to).



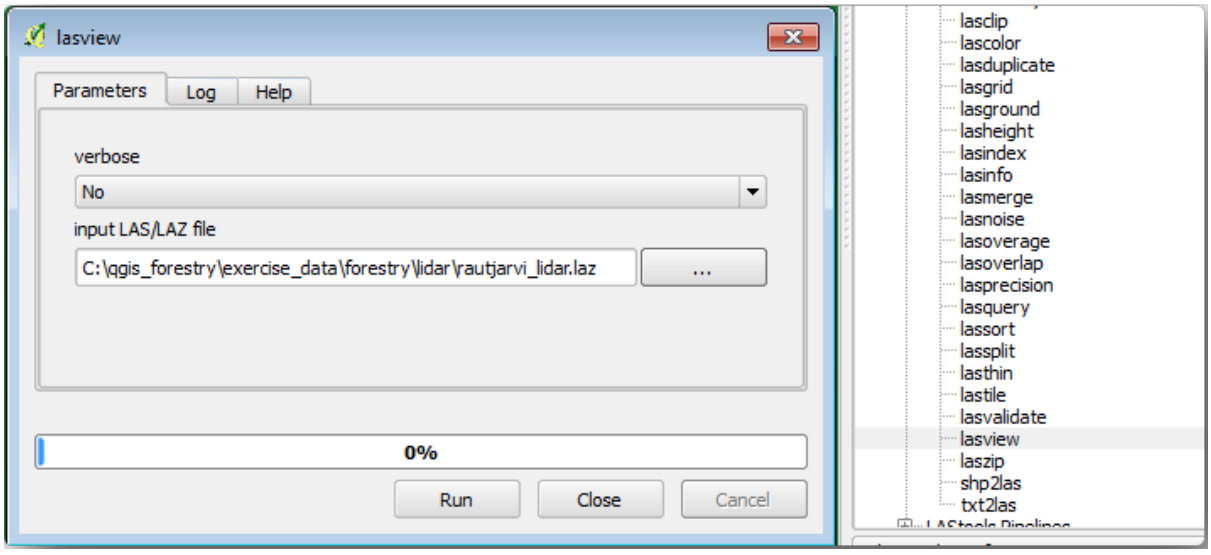
14.8.2 Follow Along: Calculating a DEM with LAStools

You have already used the *Processing* toolbox in *Занятие: Пространственная статистика* to run some SAGA algorithms. Now you are going to use it to run LAStools programs:

- Open *Processing* ► *Toolbox*.
- In the dropdown menu at the bottom, select *Advanced interface*.
- You should see the *Tools for LiDAR data* category.

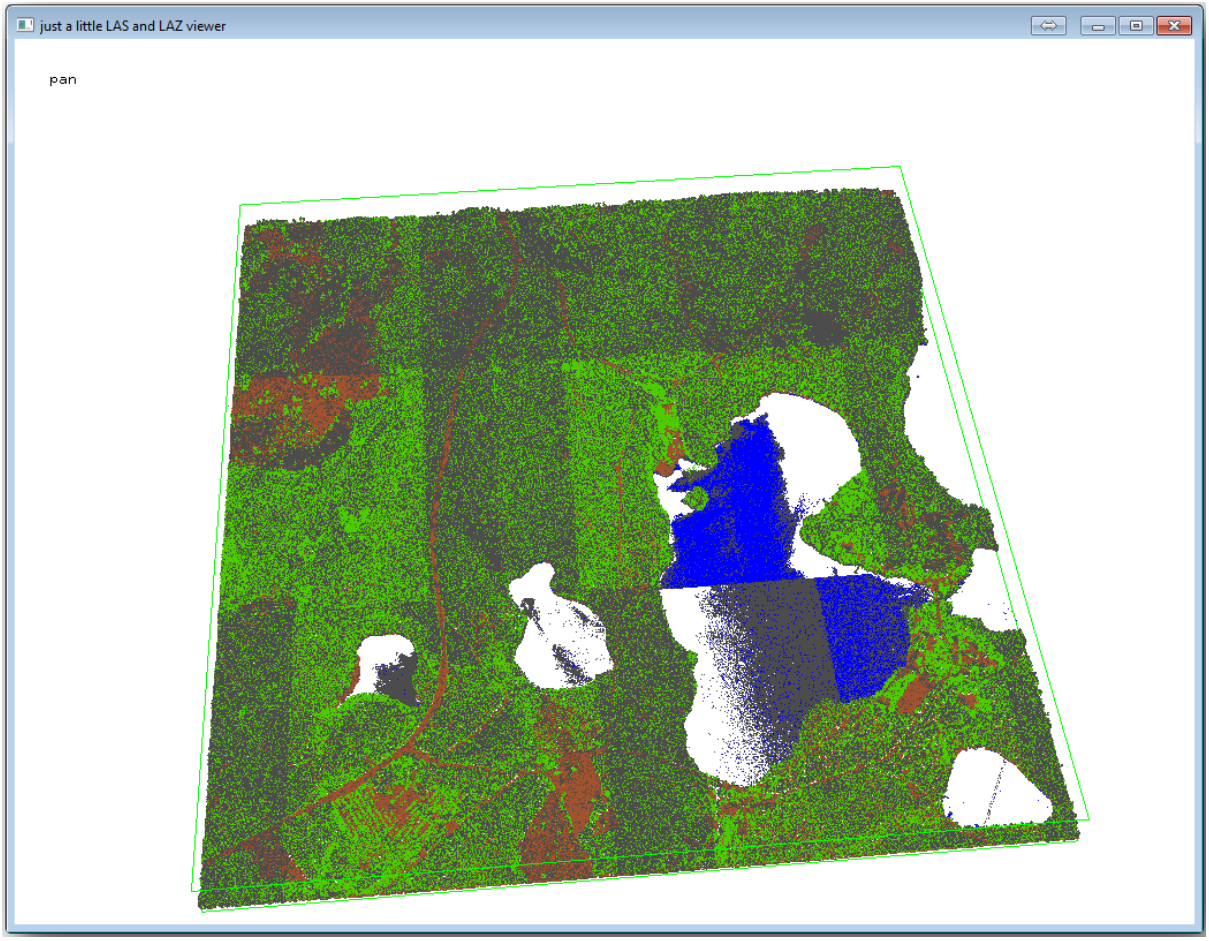


- Expand it to see the tools available, and expand also the *LAStools* category (the number of algorithms may vary).
- Scroll down until you find the *lasview* algorithm, double click it to open.
- At *Input LAS/LAZ file*, browse to `exercise_data\forestry\lidar\` and select the `rautjarvi_lidar.laz` file.



- Click *Run*.

Now you can see the LiDAR data in the *just a little LAS and LAZ viewer* dialog window:



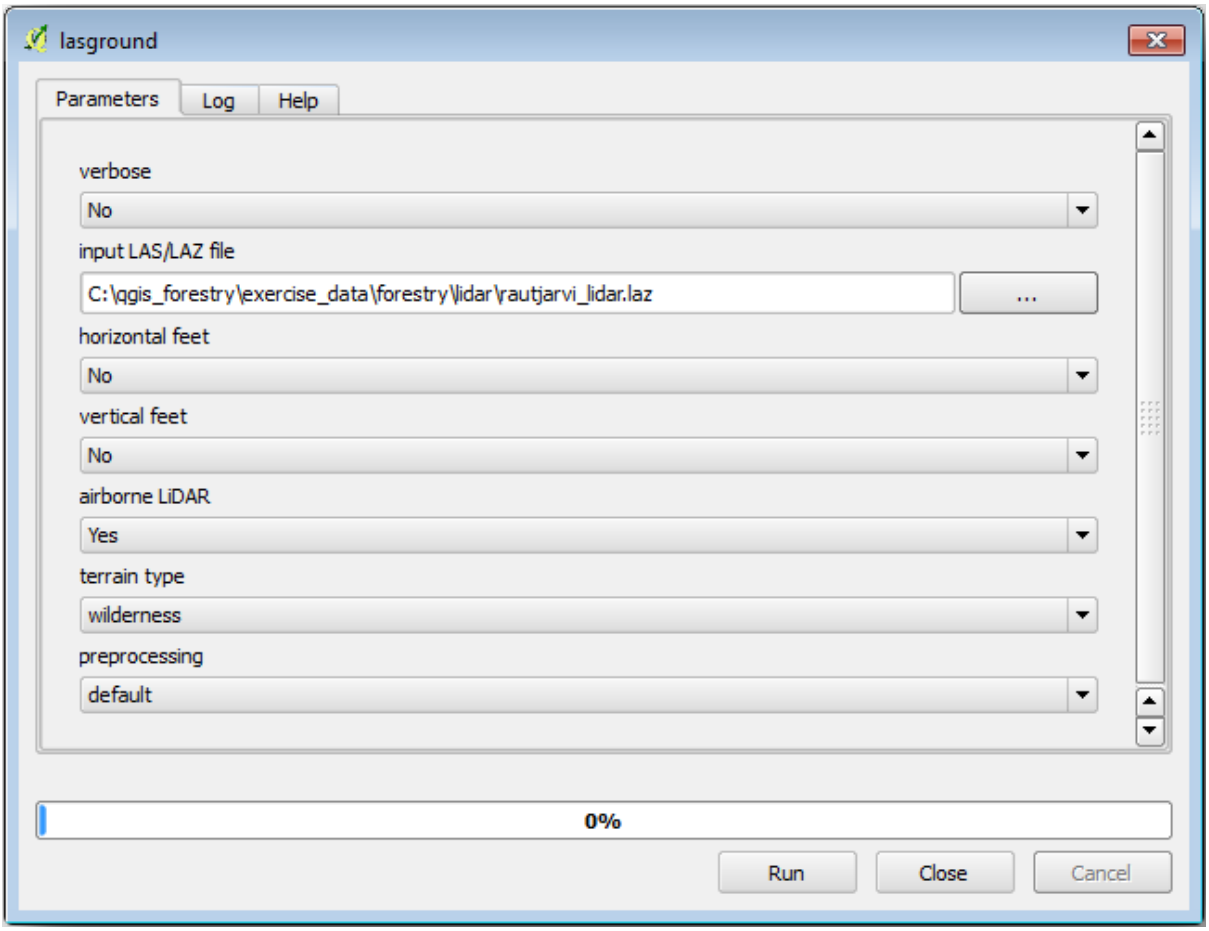
There are many things you can do within this viewer, but for now you can just click and drag on the viewer to pan the LiDAR point cloud to see what it looks like.

Примечание: If you want to know further details on how the LAStools work, you can read the `README` text files about each of the tools, in the `C:\lastools\bin\` folder. Tutorials and other materials are available at the Rapidlasso webpage.

- Close the viewer when you are ready.

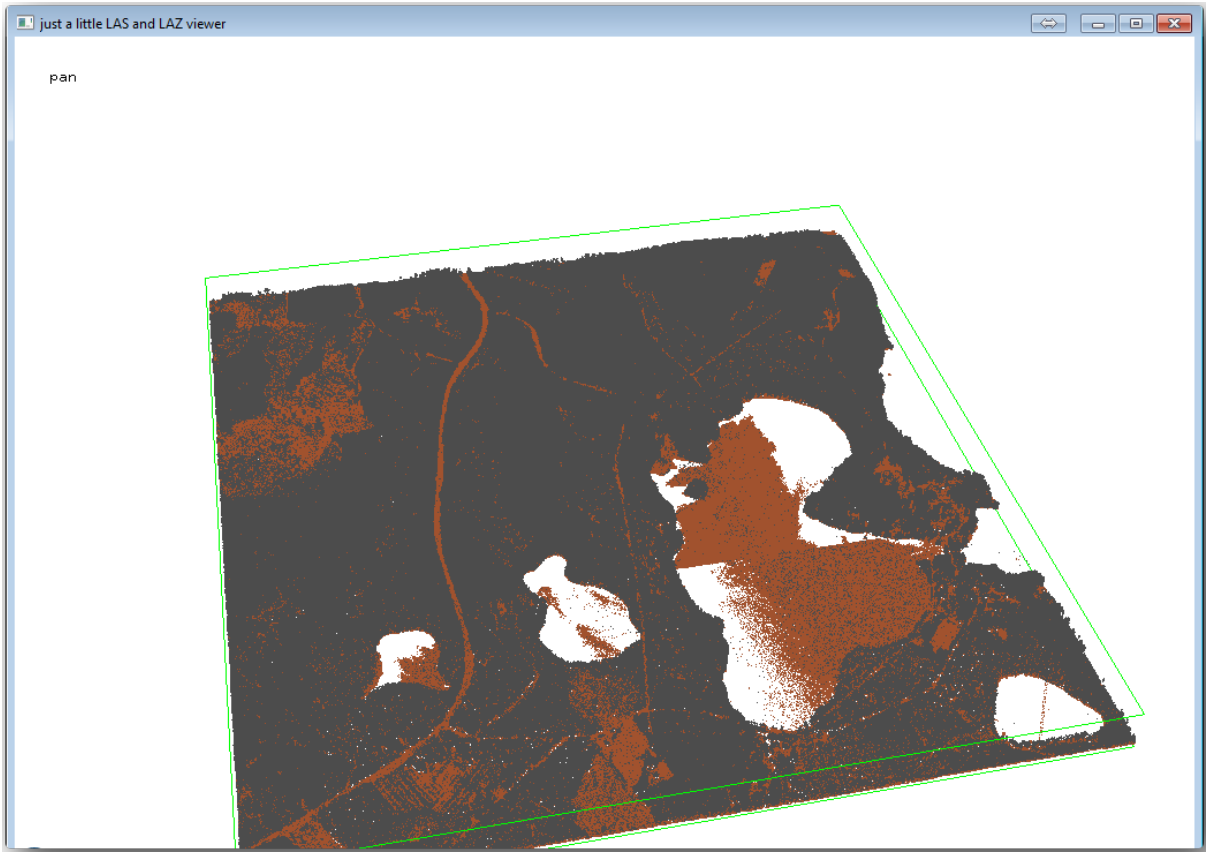
Creating a DEM with LAStools can be done in two steps, first one to classify the point cloud into `ground` and `no ground` points and then calculating a DEM using only the `ground` points.

- Go back to the *Processing Toolbox*.
- Note the *Search...* box, write `lasground`.
- Double click to open the *lasground* tool and set it as shown in this image:



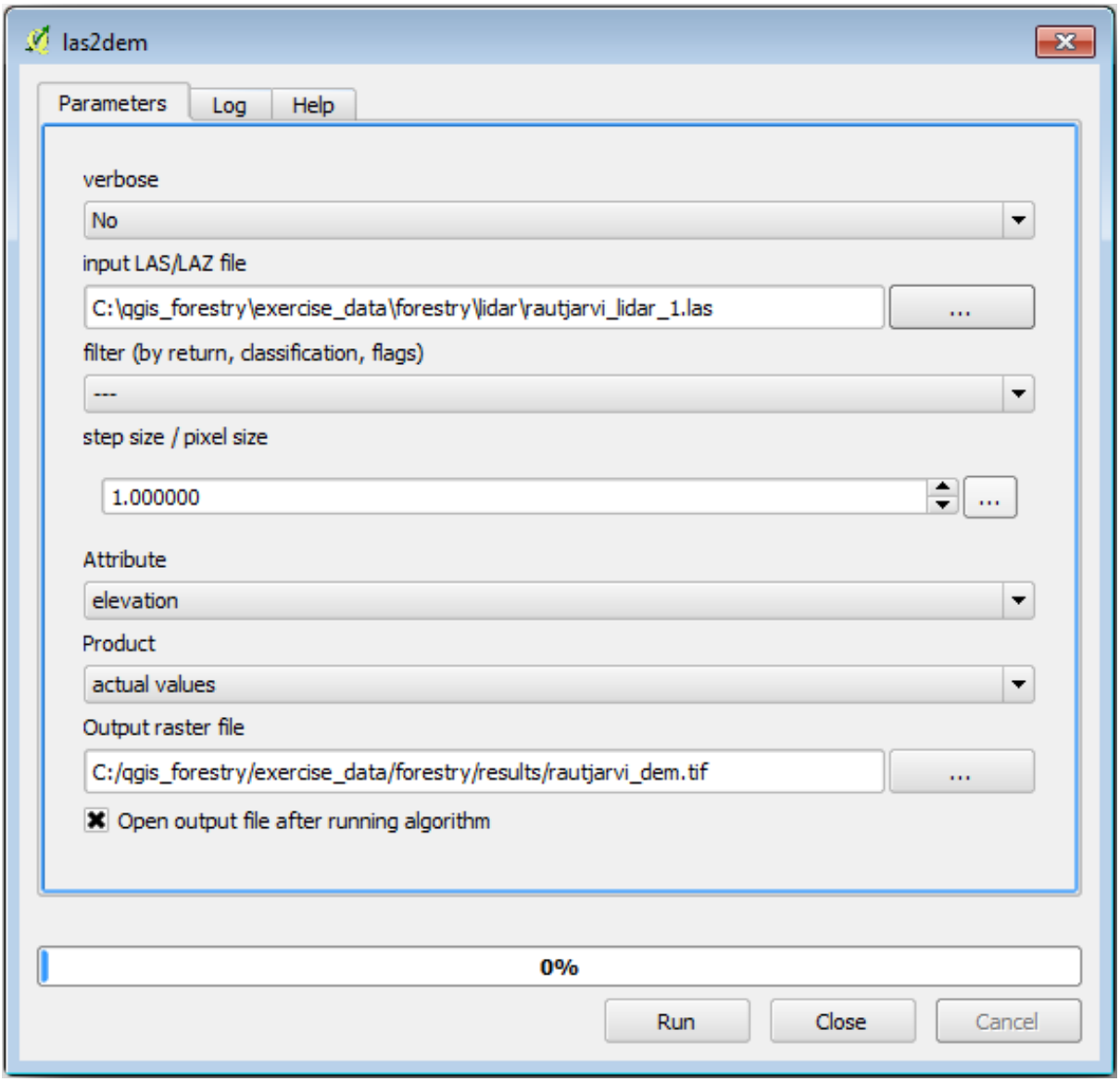
- The output file is saved to the same folder where the `rautjarvi_lidar.laz` is located and it is named `rautjarvi_lidar_1.las`.

You can open it with *lasview* if you want to check it.



The brown points are the points classified as ground and the gray ones are the rest, you can click the letter `g` to visualize only the ground points or the letter `u` to see only the unclassified points. Click the letter `a` to see all the points again. Check the `lasview_README.txt` file for more commands. If you are interested, also this tutorial about editing LiDAR points manually will show you different operations within the viewer.

- Close the viewer again.
- In the *Processing Toolbox*, search for `las2dem`.
- Open the *las2dem* tool and set it as shown in this image:



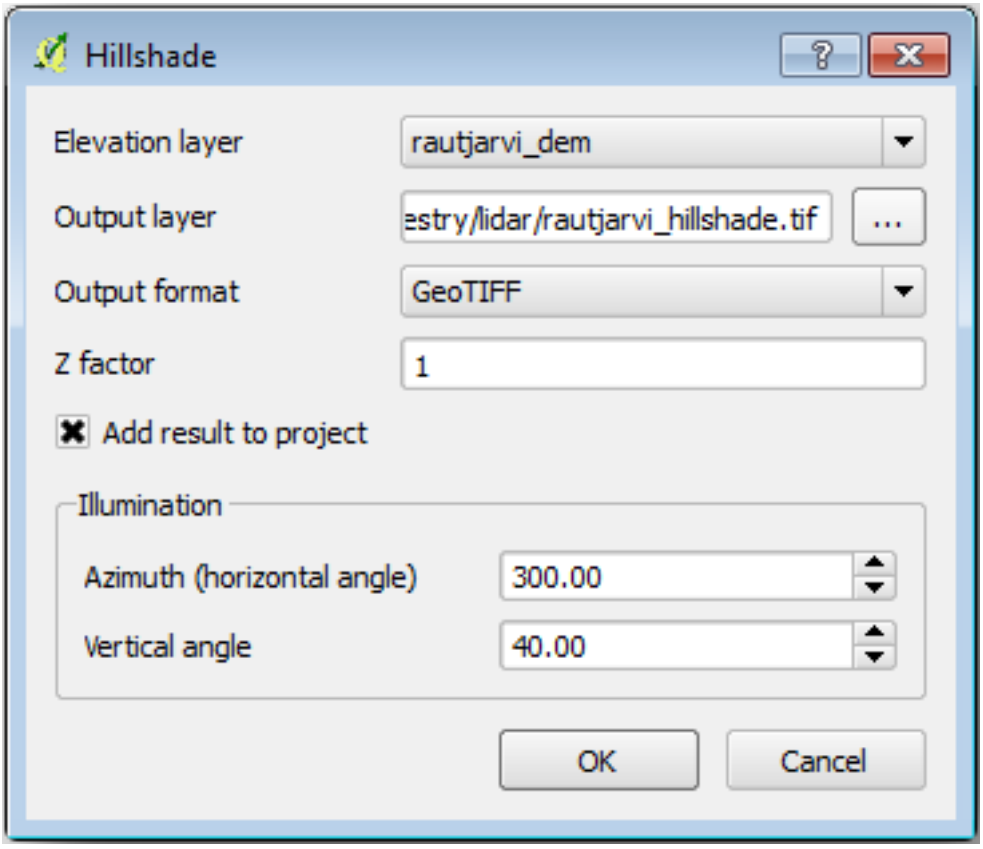
The result DEM is added to your map with the generic name `Output raster file`.

Примечание: The *lasground* and *las2dem* tools require licensing. You can use the unlicensed tool as indicated in the license file, but you get the diagonals you can appreciate in the image results.

14.8.3 Follow Along: Creating a Terrain Hillshade

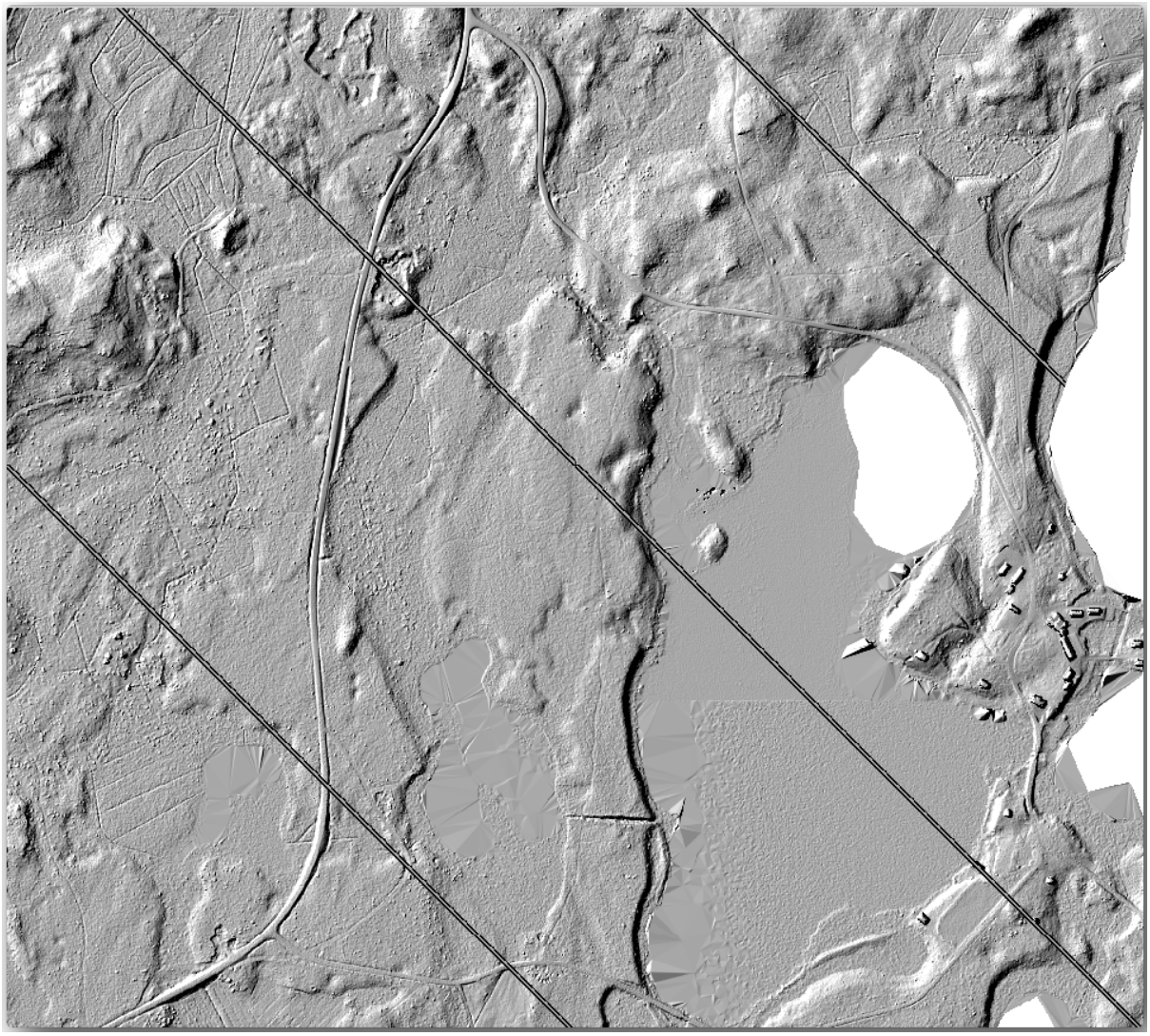
For visualization purposes, a hillshade generated from a DEM gives a better visualization of the terrain:

- Open *Raster* ► *Terrain analysis* ► *Hillshade*.
- As the *Output layer*, browse to `exercise_data\forestry\lidar\` and name the file `hillshade.tif`.
- Leave the rest of parameters with the default settings.



- Select `ETRS89 / ETRS-TM35FIN` as the CRS when prompted.

Despite the diagonal lines remaining in the hillshade raster result, you can clearly see an accurate relief of the area. You can even see the different soil drains that have been dug in the forests.



14.8.4 In Conclusion

Using LiDAR data to get a DEM, specially in forested areas, gives good results with not much effort. You could also use ready LiDAR derived DEMs or other sources like the SRTM 9m resolution DEMs. Either way, you can use them to create a hillshade raster to use in your map presentations.

14.8.5 What's Next?

In the next, and final step in this module, lesson you will use the hillshade raster and the forest inventory results to create a map presentation of the results.

14.9 Lesson: Map Presentation

In the previous lessons you have imported an old forest inventor as a GIS project, updated it to the current situation, designed a forest inventory, created maps for the field work and calculated forest parameters from the field measurements.

It is often important to create maps with the results of a GIS project. A map presenting the results of the forest inventory will make it easier for anyone to have a good idea of what the results are in a quick glance, without looking at the specific numbers.

The goal for this lesson: Create a map to present the inventory results using a hillshade raster as background.

14.9.1  Follow Along: Preparing the Map Data

Open the QGIS project from the parameters calculations lesson, `forest_inventory.qgs`. Keep at least the following layers:

- `forest_stands_2012_results`.
- `basic_map`.
- `rautjarvi_aerial`.
- `lakes` (if you don't have it, add it from the `exercise_data\forestry\` folder).

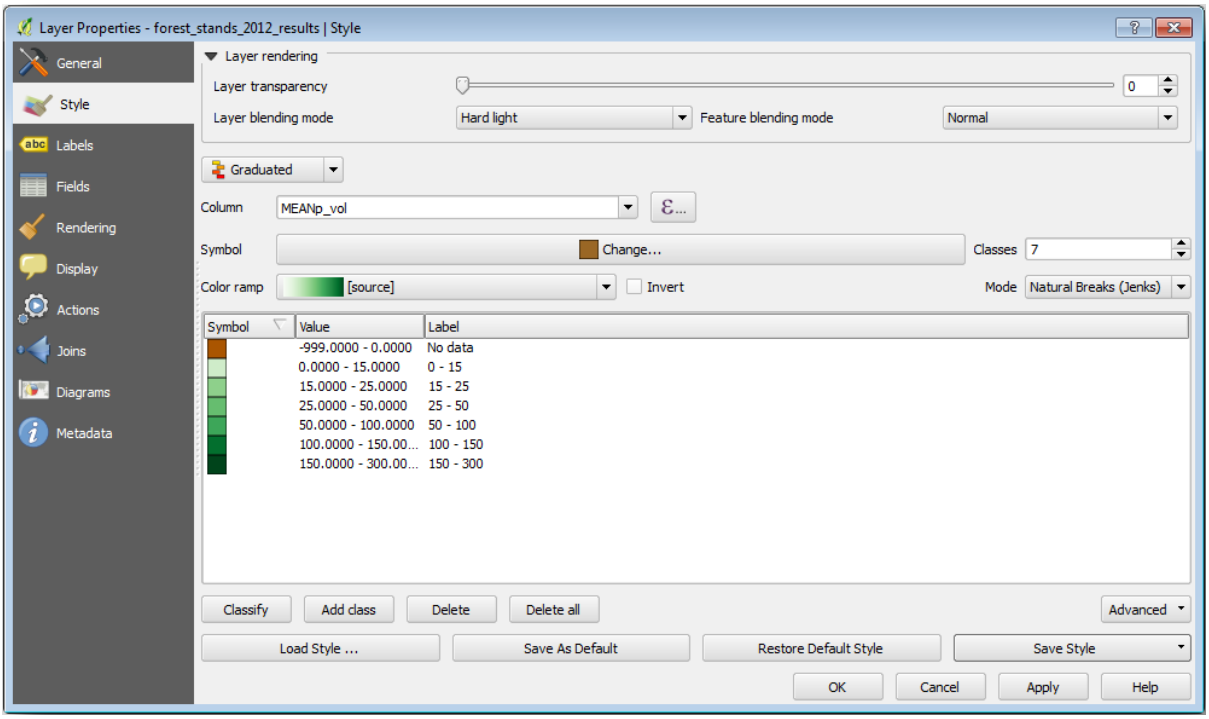
You are going to present the average volumes of your forest stands in a map. If you open the *Attribute table* for the `forest_stands_2012_results` layer, you can see the `NULL` values for the stands without information. To be able to get also those stands into your symbology you should change the `NULL` values to, for example, `-999`, knowing that those negative numbers mean there is no data for those polygons.

For the `forest_stands_2012_results` layer:

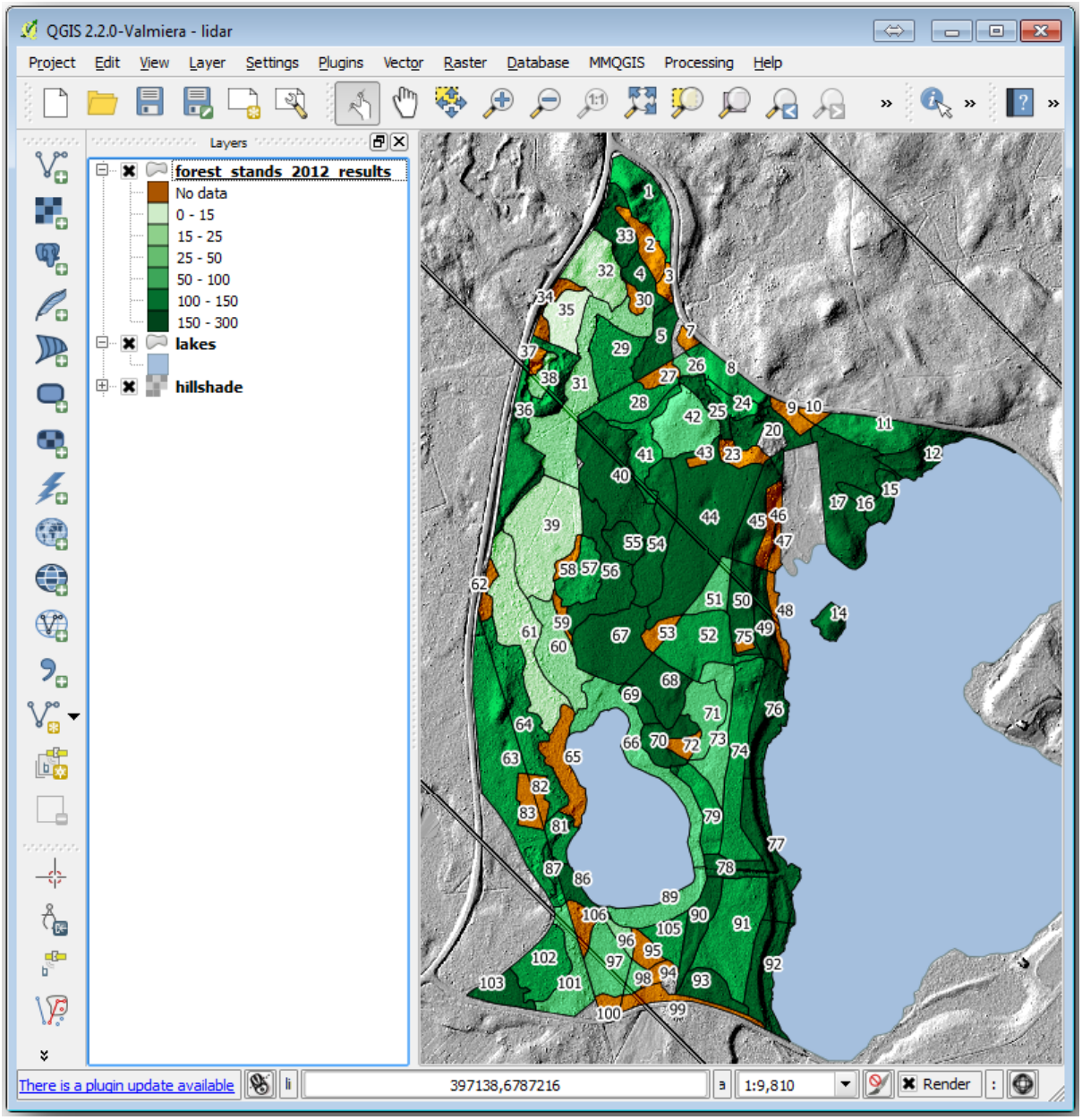
- Open the *Attribute table* and enable editing.
- Select the polygons with `NULL` values.
- Use the calculator to update the values of the `MEANVol` field to `-999` only for the selected features.
- Disable editing and save the changes.

Now you can use a saved style for this layer:

- Go to the *Symbology* tab.
- Click on *Style* ► *Load Style…*.
- Select the `forest_stands_2012_results.qml` from the `exercise_data\forestry\results\` folder.
- Click *OK*.

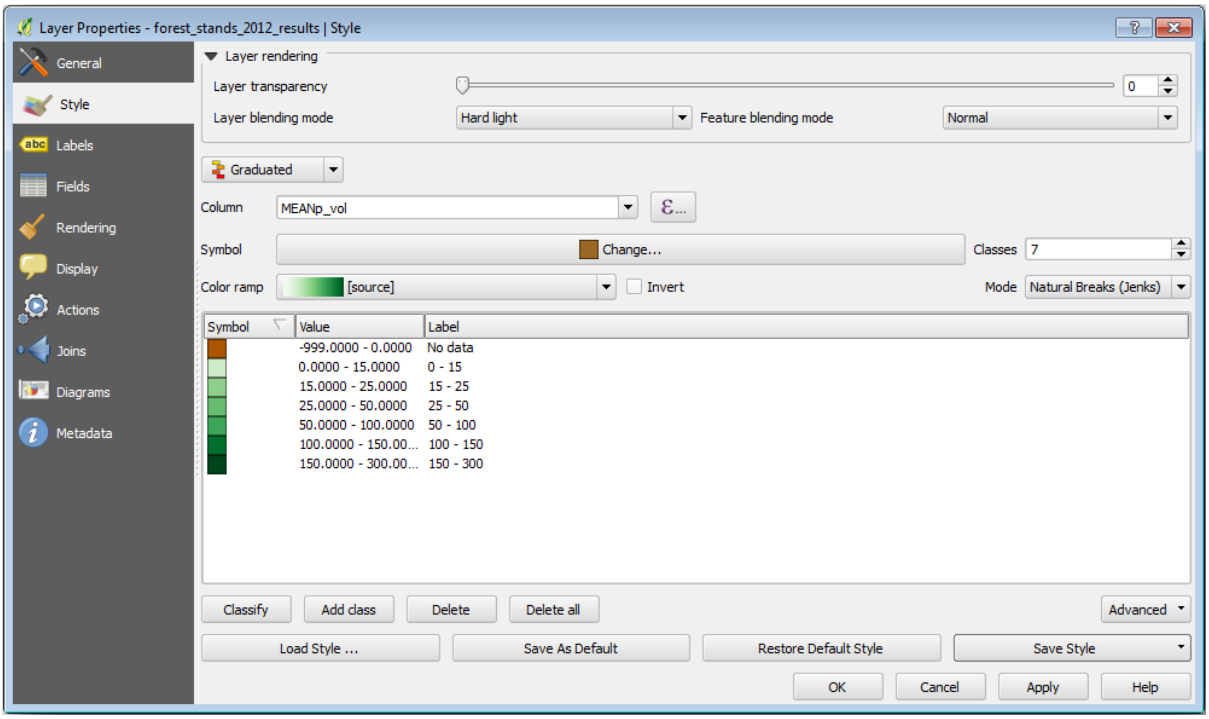


Your map will look something like this:



14.9.2 Try Yourself Try Different Blending Modes

The style you loaded:

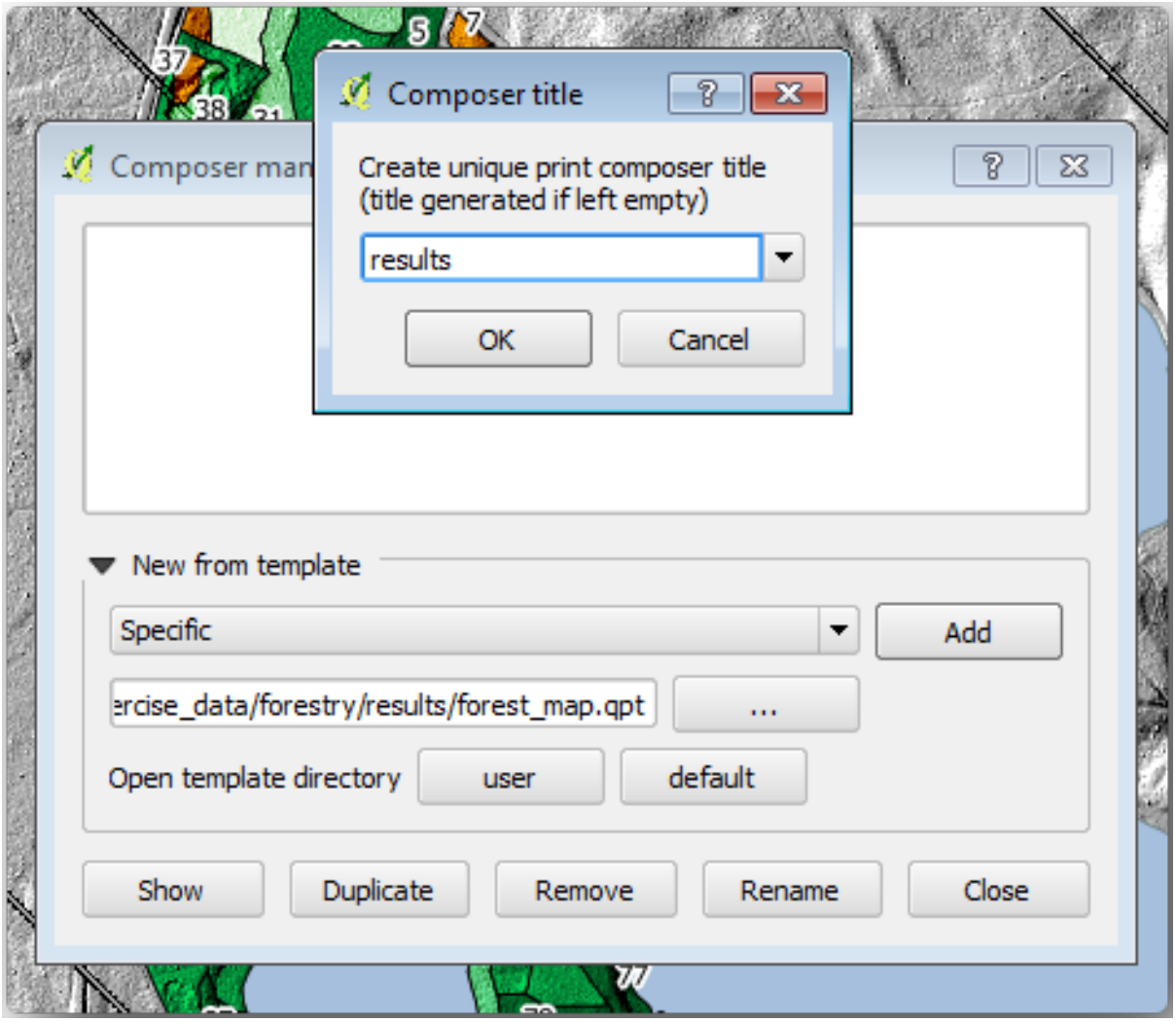


is using the `Hard light` mode for the *Layer blending mode*. Note that the different modes apply different filters combining the underlying and overlying layers, in this case the hillshade raster and your forest stands are used. You can read about these modes in the User Guide.

Try with different modes and see the differences in your map. Then choose the one you like better for your final map.

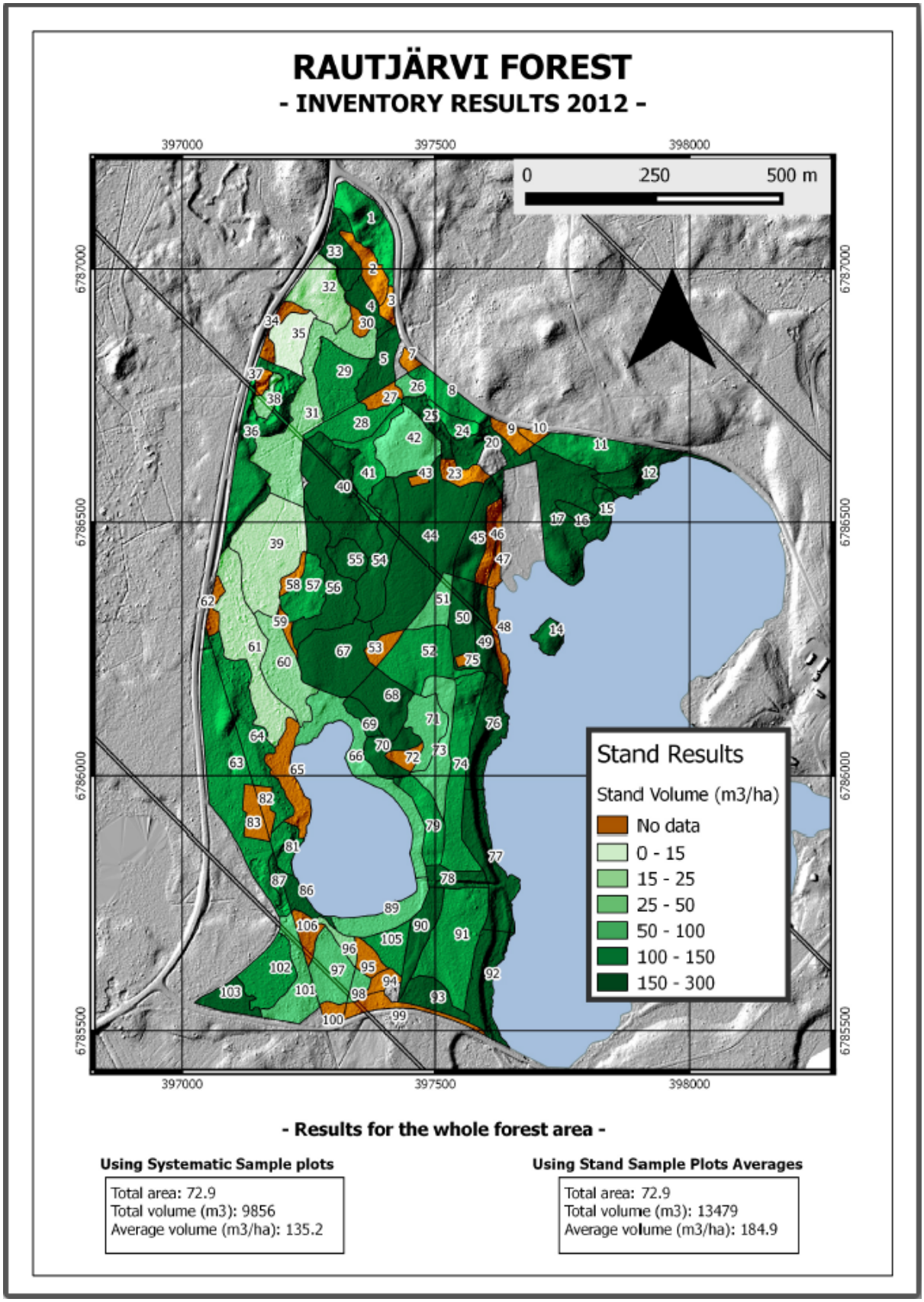
14.9.3 Try Yourself Using a Layout Template to Create the Map result

Use a template prepared in advanced to present the results. The template `forest_map.qpt` is located in the `exercise_data\forestry\results\` folder. Load it using the *Project* ► *Layout Manager…* dialog.



Open the print layout and edit the final map to get a result you are happy with.

The map template you are using will give a map similar to this one:



Save your QGIS project for future references.

14.9.4 In Conclusion

Through this module you have seen how a basic forest inventory can be planned and presented with QGIS. Many more forest analysis are possible with the variety of tools that you can access, but hopefully this manual has given you a good starting point to explore how you could achieve the specific results you need.

ГЛАВА 15

Module: Database Concepts with PostgreSQL

Relational Databases are an important part of any GIS system. In this module, you'll learn about Relational Database Management System (RDBMS) concepts and you will use PostgreSQL to create a new database to store data, as well as learning about other typical RDBMS functions.

15.1 Lesson: Introduction to Databases

Before using PostgreSQL, let's make sure of our ground by covering general database theory. You will not need to enter any of the example code; it's only there for illustration purposes.

The goal for this lesson: To understand fundamental database concepts.

15.1.1 What is a Database?

A database consists of an organized collection of data for one or more uses, typically in digital form. - *Wikipedia*

A database management system (DBMS) consists of software that operates databases, providing storage, access, security, backup and other facilities. - *Wikipedia*

15.1.2 Tables

In relational databases and flat file databases, a table is a set of data elements (values) that is organized using a model of vertical columns (which are identified by their name) and horizontal rows. A table has a specified number of columns, but can have any number of rows. Each row is identified by the values appearing in a particular column subset which has been identified as a candidate key. - *Wikipedia*

```
 id | name  | age
----+-------+-----
  1 | Tim   |  20
  2 | Horst |  88
(2 rows)
```

In SQL databases a table is also known as a **relation.**

15.1.3 Columns / Fields

A column is a set of data values of a particular simple type, one for each row of the table. The columns provide the structure according to which the rows are composed. The term field is often used interchangeably with column, although many consider it more correct to use field (or field value) to refer specifically to the single item that exists at the intersection between one row and one column. - *Wikipedia*

A column:

```
| name  |
+-------+
| Tim   |
| Horst |
```

A field:

```
| Horst |
```

15.1.4 Records

A record is the information stored in a table row. Each record will have a field for each of the columns in the table.

```
2 | Horst |  88  <-- one record
```

15.1.5 Datatypes

Datatypes restrict the kind of information that can be stored in a column. - *Tim and Horst*

There are many kinds of datatypes. Let's focus on the most common:

- `String` - to store free-form text data
- `Integer` - to store whole numbers
- `Real` - to store decimal numbers
- `Date` - to store Horst's birthday so no one forgets
- `Boolean` - to store simple true/false values

You can tell the database to allow you to also store nothing in a field. If there is nothing in a field, then the field content is referred to as a **„null" value**:

```
insert into person (age) values (40);

select * from person;
```

Результат:

```
id | name  | age
---+-------+-----
 1 | Tim   |  20
 2 | Horst |  88
 4 |       |  40  <-- null for name
(3 rows)
```

There are many more datatypes you can use - check the PostgreSQL manual!

15.1.6 Modelling an Address Database

Let's use a simple case study to see how a database is constructed. We want to create an address database.

Try Yourself 

Write down the properties which make up a simple address and which we would want to store in our database.

Answer

For our theoretical address table, we might want to store the following properties:

```
House Number
Street Name
Suburb Name
City Name
Postcode
Country
```

When creating the table to represent an address object, we would create columns to represent each of these properties and we would name them with SQL-compliant and possibly shortened names:

```
house_number
street_name
suburb
city
postcode
country
```

Address Structure

The properties that describe an address are the columns. The type of information stored in each column is its datatype. In the next section we will analyse our conceptual address table to see how we can make it better!

15.1.7 Database Theory

The process of creating a database involves creating a model of the real world; taking real world concepts and representing them in the database as entities.

15.1.8 Normalisation

One of the main ideas in a database is to avoid data duplication / redundancy. The process of removing redundancy from a database is called Normalisation.

Normalization is a systematic way of ensuring that a database structure is suitable for general-purpose querying and free of certain undesirable characteristics - insertion, update, and deletion anomalies - that could lead to a loss of data integrity. - *Wikipedia*

There are different kinds of normalisation „forms".

Let's take a look at a simple example:

```
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)
```

Imagine you have many friends with the same street name or city. Every time this data is duplicated, it consumes space. Worse still, if a city name changes, you have to do a lot of work to update your database.

15.1.9 Try Yourself 

Redesign the theoretical *people* table above to reduce duplication and to normalise the data structure.

You can read more about database normalisation here

Answer

The major problem with the *people* table is that there is a single address field which contains a person's entire address. Thinking about our theoretical *address* table earlier in this lesson, we know that an address is made up of many different properties. By storing all these properties in one field, we make it much harder to update and query our data. We therefore need to split the address field into the various properties. This would give us a table which has the following structure:

```
id |     name      | house_no |  street_name   |    city    |   phone_no
 --+---------------+----------+----------------+------------+-----------------
 1 | Tim Sutton    |     3    | Buirski Plein  | Swellendam | 071 123 123
 2 | Horst Duester |     4    | Avenue du Roix | Geneva     | 072 121 122
```

In the next section, you will learn about Foreign Key relationships which could be used in this example to further improve our database's structure.

15.1.10 Indexes

A database index is a data structure that improves the speed of data retrieval operations on a database table. - *Wikipedia*

Imagine you are reading a textbook and looking for the explanation of a concept - and the textbook has no index! You will have to start reading at one cover and work your way through the entire book until you find the information you need. The index at the back of a book helps you to jump quickly to the page with the relevant information:

```
create index person_name_idx on people (name);
```

Now searches on name will be faster:

```
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 Sequences

A sequence is a unique number generator. It is normally used to create a unique identifier for a column in a table.

In this example, id is a sequence - the number is incremented each time a record is added to the table:

```
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 Entity Relationship Diagramming

In a normalised database, you typically have many relations (tables). The entity-relationship diagram (ER Diagram) is used to design the logical dependencies between the relations. Consider our non-normalised *people* table from earlier in the lesson:

```
select * from people;

 id |     name     |           address           |  phone_no
----+--------------+-----------------------------+-------------
 1  | Tim Sutton   | 3 Buirski Plein, Swellendam | 071 123 123
 2  | Horst Duster | 4 Avenue du Roix, Geneva    | 072 121 122
(2 rows)
```

With a little work we can split it into two tables, removing the need to repeat the street name for individuals who live in the same street:

```
select * from streets;

 id |     name
----+--------------
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 1  | Plein Street
(1 row)
```

and:

```
select * from people;

 id |     name      | house_no | street_id |  phone_no
----+---------------+----------+-----------+-------------
  1 | Horst Duster |        4 |         1 | 072 121 122
(1 row)
```

We can then link the two tables using the „keys“ `streets.id` and `people.streets_id`.

If we draw an ER Diagram for these two tables it would look something like this:



The ER Diagram helps us to express „one to many“ relationships. In this case the arrow symbol show that one street can have many people living on it.

Try Yourself

Our *people* model still has some normalisation issues - try to see if you can normalise it further and show your thoughts by means of an ER Diagram.

Answer

Our *people* table currently looks like this:

```
id |     name      | house_no | street_id |  phone_no
---+---------------+----------+-----------+-------------
 1 | Horst Duster |        4 |         1 | 072 121 122
```

The *street_id* column represents a „one to many“ relationship between the people object and the related street object, which is in the *streets* table.

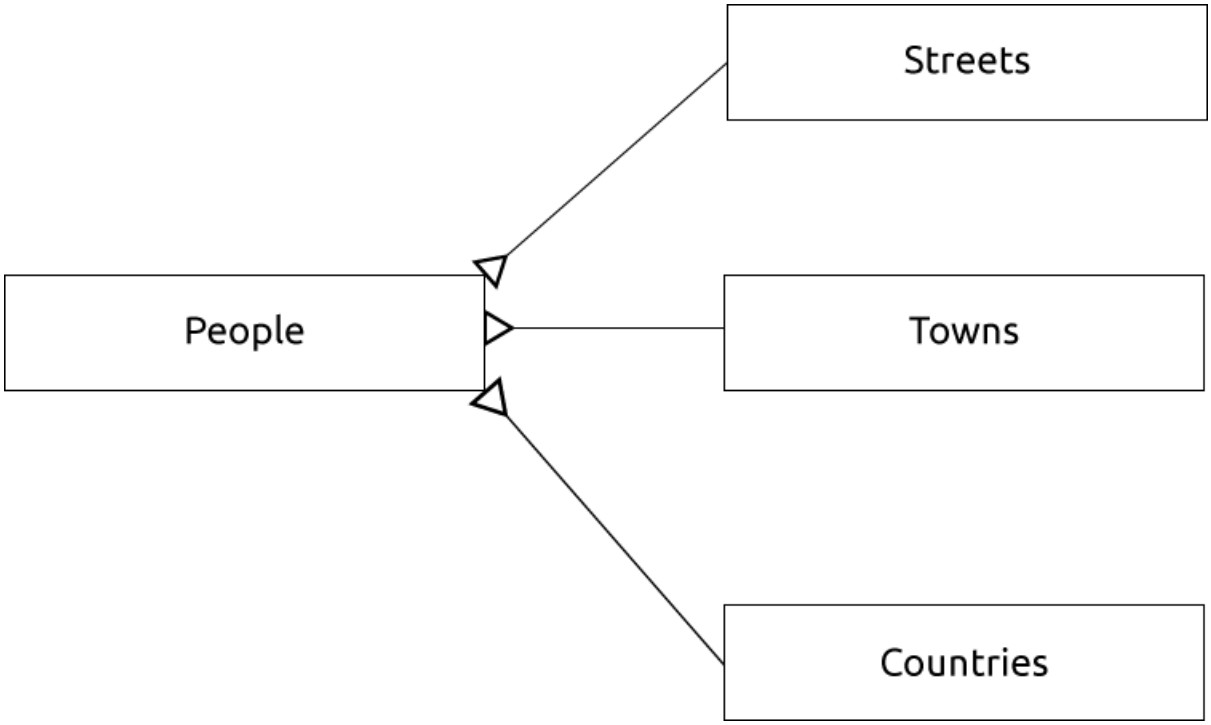
One way to further normalise the table is to split the name field into *first_name* and *last_name*:

```
id | first_name | last_name  | house_no | street_id |  phone_no
---+------------+------------+----------+-----------+------------
 1 |    Horst   |   Duster   |     4    |     1     | 072 121 122
```

We can also create separate tables for the town or city name and country, linking them to our *people* table via „one to many“ relationships:

```
id | first_name | last_name | house_no | street_id | town_id | country_id
---+------------+-----------+----------+-----------+---------+------------
 1 |    Horst   |   Duster  |     4    |     1     |    2    |     1
```

An ER Diagram to represent this would look like this:



15.1.13 Constraints, Primary Keys and Foreign Keys

A database constraint is used to ensure that data in a relation matches the modeller's view of how that data should be stored. For example a constraint on your postal code could ensure that the number falls between `1000` and `9999`.

A Primary key is one or more field values that make a record unique. Usually the primary key is called id and is a sequence.

A Foreign key is used to refer to a unique record on another table (using that other table's primary key).

In ER Diagramming, the linkage between tables is normally based on Foreign keys linking to Primary keys.

If we look at our people example, the table definition shows that the street column is a foreign key that references the primary key on the streets table:

```
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 Transactions

When adding, changing, or deleting data in a database, it is always important that the database is left in a good state if something goes wrong. Most databases provide a feature called transaction support. Transactions allow you to create a rollback position that you can return to if your modifications to the database did not run as planned.

Take a scenario where you have an accounting system. You need to transfer funds from one account and add them to another. The sequence of steps would go like this:

- remove R20 from Joe
- add R20 to Anne

If something goes wrong during the process (e.g. power failure), the transaction will be rolled back.

15.1.15 In Conclusion

Databases allow you to manage data in a structured way using simple code structures.

15.1.16 What's Next?

Now that we've looked at how databases work in theory, let's create a new database to implement the theory we've covered.

15.2 Lesson: Implementing the Data Model

Now that we've covered all the theory, let's create a new database. This database will be used for our exercises for the lessons that will follow afterwards.

The goal for this lesson: To install the required software and use it to implement our example database.

15.2.1 Install PostgreSQL

Примечание: You can find PostgreSQL packages and installation instructions for your operating system at https://www.postgresql.org/download/. Please note that the documentation will assume users are running QGIS under Ubuntu.

Under Ubuntu:

```
sudo apt install postgresql-9.1
```

You should get a message like this:

```
[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.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
After this operation, 19.0MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Press `Y` and `Enter` and wait for the download and installation to finish.

15.2.2 Справка

PostgreSQL has very good online documentation.

15.2.3 Create a database user

Under Ubuntu:

After the installation is complete, run this command to become the postgres user and then create a new database user:

```
sudo su - postgres
```

Type in your normal log in password when prompted (you need to have sudo rights).

Now, at the postgres user's bash prompt, create the database user. Make sure the user name matches your unix login name: it will make your life much easier, as postgres will automatically authenticate you when you are logged in as that user:

```
createuser -d -E -i -l -P -r -s qgis
```

Enter a password when prompted. You should use a different password to your login password.

What do those options mean?

```
-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
```

Now you should leave the postgres user's bash shell environment by typing:

```
exit
```

15.2.4 Verify the new account

```
psql -l
```

Should return something like this:

```
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)
```

Type `Q` to exit.

15.2.5 Create a database

The `createdb` command is used to create a new database. It should be run from the bash shell prompt:

```
createdb address -O qgis
```

You can verify the existence of your new database by using this command:

```
psql -l
```

Which should return something like this:

```
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)
```

Type `Q` to exit.

15.2.6 Starting a database shell session

You can connect to your database easily like this:

```
psql address
```

To exit out of the psql database shell, type:

```
\q
```

For help in using the shell, type:

```
\?
```

For help in using sql commands, type:

```
\help
```

To get help on a specific command, type (for example):

```
\help create table
```

See also the Psql cheat sheet.

15.2.7 Make Tables in SQL

Let's start making some tables! We will use our ER Diagram as a guide. First, connect to the address db:

```
psql address
```

Then create a `streets` table:

```
create table streets (id serial not null primary key, name varchar(50));
```

`serial` and `varchar` are **data types**. `serial` tells PostgreSQL to start an integer sequence (auto-number) to populate the `id` automatically for every new record. `varchar(50)` tells PostgreSQL to create a character field of 50 characters in length.

You will notice that the command ends with a `;` - all SQL commands should be terminated this way. When you press `Enter`, psql will report something like this:

```
NOTICE:  CREATE TABLE will create implicit sequence "streets_id_seq"
         for serial column "streets.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
         "streets_pkey" for table "streets"
CREATE TABLE
```

That means your table was created successfully, with a primary key `streets_pkey` using `streets.id`.

Note: If you hit return without entering a `;`, then you will get a prompt like this: `address-#`. This is because PG is expecting you to enter more. Enter `;` to run your command.

To view your table schema, you can do this:

```
\d streets
```

Which should show something like this:

```
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)
```

To view your table contents, you can do this:

```
select * from streets;
```

Which should show something like this:

```
id | name
---+------
(0 rows)
```

As you can see, our table is currently empty.

Try Yourself 

Use the approach shown above to make a table called people:

Add fields such as phone number, home address, name, etc. (these aren't all valid names: change them to make them valid). Make sure you give the table an ID column with the same data-type as above.

Answer

The SQL required to create the correct people table is:

```
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 );
```

The schema for the table (enter \\d people) looks like this:

```
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)
```

For illustration purposes, we have purposely omitted the fkey constraint.

15.2.8 Create Keys in SQL

The problem with our solution above is that the database doesn't know that people and streets have a logical relationship. To express this relationship, we have to define a foreign key that points to the primary key of the streets table.



There are two ways to do this:

- Add the key after the table has been created
- Define the key at time of table creation

Our table has already been created, so let's do it the first way:

```
alter table people
  add constraint people_streets_fk foreign key (street_id) references streets(id);
```

That tells the people table that its street_id fields must match a valid street id from the streets table.

The more usual way to create a constraint is to do it when you create the table:

```
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
```

After adding the constraint, our table schema looks like this now:

```
Table "public.people"

  Column    |         Type          |            Modifiers
------------+-----------------------+---------------------------------
 id         | integer               | not null default
            |                       | nextval('people_id_seq'::regclass)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 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 Create Indexes in SQL

We want lightning fast searches on peoples names. To provide for this, we can create an index on the name column of our people table:

```
create index people_name_idx on people(name);

\d people
```

Which results in:

```
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 Dropping Tables in SQL

If you want to get rid of a table you can use the `drop` command:

```
drop table streets;
```

In our current example, the above command would not work. Why not?

Answer

The reason the DROP command would not work in this case is because the *people* table has a Foreign Key constraint to the *streets* table. This means that dropping (or deleting) the *streets* table would leave the *people* table with references to non-existent *streets* data.

It is possible to „force“ the *streets* table to be deleted by using the `CASCADE` command, but this would also delete the *people* and any other table which had a relationship to the *streets* table. Use with caution!

If you used the same `drop table` command on the *people* table, it would be successful:

```
drop table people;
```

Примечание: If you actually did enter that command and dropped the `people` table, now would be a good time to rebuild it, as you will need it in the next exercises.

15.2.11 A word on pgAdmin III

We are showing you the SQL commands from the *psql* prompt because it's a very useful way to learn about databases. However, there are quicker and easier ways to do a lot of what we are showing you. Install pgAdmin III and you can create, drop, alter etc tables using „point and click" operations in a GUI.

Under Ubuntu, you can install it like this:

```
sudo apt install pgadmin3
```

pgAdmin III will be covered in more detail in another module.

15.2.12 In Conclusion

You have now seen how to create a brand new database, starting completely from scratch.

15.2.13 What's Next?

Next you'll learn how to use the DBMS to add new data.

15.3 Lesson: Adding Data to the Model

The models we've created will now need to be populated with the data they're intended to contain.

The goal for this lesson: To learn how to insert new data into the database models.

15.3.1 Insert statement

How do you add data to a table? The sql `INSERT` statement provides the functionality for this:

```
insert into streets (name) values ('High street');
```

A couple of things to note:

- After the table name (`streets`), you list the column names that you will be populating (in this case only the `name` column).
- After the `values` keyword, place the list of field values.
- Strings should be quoted using single quotes.
- Note that we did not insert a value for the `id` column; this is because it is a sequence and will be auto-generated.
- If you do manually set the `id`, you may cause serious problems with the integrity of your database.

You should see `INSERT 0 1` if it is successful.

You can see the result of your insert action by selecting all the data in the table:

```
select * from streets;
```

Результат:

```
select * from streets;
 id |     name
----+--------------
  1 | High street
(1 row)
```

Try Yourself 

Use the `INSERT` command to add a new street to the `streets` table.

Answer

The SQL command you should use looks like this (you can replace the street name with a name of your choice):

```
insert into streets (name) values ('Low Road');
```

15.3.2 Sequencing Data Addition According to Constraints

15.3.3 Try Yourself 

Try to add a person object to the `people` table with the following details:

```
Name: Joe Smith
House Number: 55
Street: Main Street
Phone: 072 882 33 21
```

Примечание: Recall that in this example, we defined phone numbers as strings, not integers.

At this point, you should have an error report if you try to do this without first creating a record for Main Street in the `streets` table.

You should have also noticed that:

- You can't add the street using its name
- You can't add a street using a street `id` before first creating the street record on the streets table

Remember that our two tables are linked via a Primary/Foreign Key pair. This means that no valid person can be created without there also being a valid corresponding street record.

Using the above knowledge, add the new person to the database.

Answer

Here is the correct SQL statement:

```
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');
```

If you look at the streets table again (using a select statement as before), you'll see that the *id* for the `Main Road` entry is *2*.

That's why we could merely enter the number `2` above. Even though we're not seeing `Main Road` written out fully in the entry above, the database will be able to associate that with the *street_id* value of *2*.

If you have already added a new street object, you might find that the new `Main Road` has an *id* of *3* not *2*.

15.3.4 Select data

We have already shown you the syntax for selecting records. Let's look at a few more examples:

```
select name from streets;
```

```
select * from streets;
```

```
select * from streets where name='Main Road';
```

In later sessions we will go into more detail on how to select and filter data.

15.3.5 Update data

What if you want to make a change to some existing data? For example, a street name is changed:

```
update streets set name='New Main Road' where name='Main Road';
```

Be very careful using such update statements - if more than one record matches your `WHERE` clause, they will all be updated!

A better solution is to use the primary key of the table to reference the record to be changed:

```
update streets set name='New Main Road' where id=2;
```

It should return `UPDATE 1`.

Примечание: the `WHERE` statement criteria are case sensitive `Main Road` is not the same as `Main road`

15.3.6 Delete Data

In order to delete an object from a table, use the `DELETE` command:

```
delete from people where name = 'Joe Smith';
```

Let's look at our people table now:

```
address=# select * from people;

 id | name | house_no | street_id | phone_no
----+------+----------+-----------+----------
(0 rows)
```

15.3.7 Try Yourself 

Use the skills you have learned to add some new friends to your database:

```
      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

Now you know how to add new data to the existing models you created previously. Remember that if you want to add new kinds of data, you may want to modify and/or create new models to contain that data.

15.3.9 What's Next?

Now that you've added some data, you'll learn how to use queries to access this data in various ways.

15.4 Lesson: Queries

When you write a `SELECT ...` command it is commonly known as a query - you are interrogating the database for information.

The goal of this lesson: To learn how to create queries that will return useful information.

Примечание: If you did not do so in the previous lesson, add the following people objects to your `people` table. If you receive any errors related to foreign key constraints, you will need to add the „Main Road“ object to your streets table first

---

```
insert into people (name,house_no, street_id, phone_no)
          values ('Joe Bloggs',3,2,'072 887 23 45');
insert into people (name,house_no, street_id, phone_no)
          values ('Jane Smith',55,3,'072 837 33 35');
insert into people (name,house_no, street_id, phone_no)
          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 Ordering Results

Let's retrieve a list of people ordered by their house numbers:

```
select name, house_no from people order by house_no;
```

Результат:

```
     name      | house_no
---------------+----------
 Joe Bloggs    |        3
 Roger Jones   |       33
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 Jane Smith   |       55
 Sally Norman |       83
(4 rows)
```

You can sort the results by the values of more than one column:

```
select name, house_no from people order by name, house_no;
```

Результат:

```
     name      | house_no
---------------+----------
 Jane Smith    |       55
 Joe Bloggs    |        3
 Roger Jones   |       33
 Sally Norman  |       83
(4 rows)
```

15.4.2 Filtering

Often you won't want to see every single record in the database - especially if there are thousands of records and you are only interested in seeing one or two.

Here is an example of a numerical filter which only returns objects whose `house_no` is less than 50:

```
select name, house_no from people where house_no < 50;

     name      | house_no
  -------------+----------
   Joe Bloggs  |        3
   Roger Jones |       33
  (2 rows)
```

You can combine filters (defined using the `WHERE` clause) with sorting (defined using the `ORDER BY` clause):

```
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)
```

You can also filter based on text data:

```
select name, house_no from people where name like '%s%';

     name      | house_no
  -------------+----------
   Joe Bloggs  |        3
   Roger Jones |       33
  (2 rows)
```

Here we used the `LIKE` clause to find all names with an `s` in them. You'll notice that this query is case-sensitive, so the `Sally Norman` entry has not been returned.

If you want to search for a string of letters regardless of case, you can do a case in-sensitive search using the `ILIKE` clause:

```
select name, house_no from people where name ilike '%r%';

      name      | house_no
  --------------+----------
   Roger Jones  |       33
   Sally Norman |       83
  (2 rows)
```

That query returned every **people** object with an `r` or `R` in their name.

15.4.3 Joins

What if you want to see the person's details and their street's name instead of the ID? In order to do that, you need to join the two tables together in a single query. Lets look at an example:

```
select people.name, house_no, streets.name
from people,streets
where people.street_id=streets.id;
```

Примечание: With joins, you will always state the two tables the information is coming from, in this case people and streets. You also need to specify which two keys must match (foreign key & primary key). If you don't specify that, you will get a list of all possible combinations of people and streets, but no way to know who actually lives on which street!

Here is what the correct output will look like:

```
     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)
```

We will revisit joins as we create more complex queries later. Just remember they provide a simple way to combine the information from two or more tables.

15.4.4 Sub-Select

Sub-selections allow you to select objects from one table based on the data from another table which is linked via a foreign key relationship. In our case, we want to find people who live on a specific street.

First, let's do a little tweaking of our data:

```
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;
```

Let's take a quick look at our data after those changes: we can reuse our query from the previous section:

```
select people.name, house_no, streets.name
from people,streets
where people.street_id=streets.id;
```

Результат:

```
     name      | house_no |     name
---------------+----------+-------------
 Roger Jones   |       33 | High street
 Sally Norman  |       83 | High street
 Jane Smith    |       55 | Main Road
 Joe Bloggs    |        3 | Low Street
(4 rows)
```

Now let's show you a sub-selection on this data. We want to show only people who live in `street_id` number `1`:

```
select people.name
from people, (
    select *
    from streets
    where id=1
  ) as streets_subset
where people.street_id = streets_subset.id;
```

Результат:

```
     name
--------------
 Roger Jones
 Sally Norman
(2 rows)
```

Although this is a very simple example and unnecessary with our small data-sets, it illustrates how useful and important sub-selections can be when querying large and complex data-sets.

15.4.5 Aggregate Queries

One of the powerful features of a database is its ability to summarise the data in its tables. These summaries are called aggregate queries. Here is a typical example which tells us how many people objects are in our people table:

```
select count(*) from people;
```

Результат:

```
 count
-------
     4
(1 row)
```

If we want the counts to be summarised by street name we can do this:

```
select count(name), street_id
from people
group by street_id;
```

Результат:

```
 count | street_id
-------+-----------
     2 |         1
     1 |         3
     1 |         2
(3 rows)
```

Примечание: Because we have not used an `ORDER BY` clause, the order of your results may not match what is shown here.

Try Yourself 

Summarise the people by street name and show the actual street names instead of the street_ids.

Answer

Here is the correct SQL statement you should use:

```
select count(people.name), streets.name
from people, streets
where people.street_id=streets.id
group by streets.name;
```

Result:

```
count |    name
------+-------------
    1 | Low Street
    2 | High street
    1 | Main Road
(3 rows)
```

You will notice that we have prefixed field names with table names (e.g. people.name and streets.name). This needs to be done whenever the field name is ambiguous (i.e. not unique across all tables in the database).

15.4.6 In Conclusion

You've seen how to use queries to return the data in your database in a way that allows you to extract useful information from it.

15.4.7 What's Next?

Next you'll see how to create views from the queries that you've written.

15.5 Lesson: Представления

Когда Вы пишите запрос, Вам приходится тратить много времени и усилий для его создания. Используя представления, Вы можете сохранить текст SQL запроса в „виртуальной таблице“ для многократного использования.

Цель этого урока: Сохранить запрос как представление.

15.5.1 Создание представления

Вы можете использовать представление точно так же, как и таблицу, но при этом источником данных является запрос. Давайте создадим простое представление на основе вышеизложенного:

```
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;
```

Как Вы можете видеть, добавилось только `create view roads_count_v as` перед началом запроса. Теперь мы можем получить данные из представления:

```
select * from roads_count_v;
```

Результат:

```
 count |     name
-------+-------------
     1 | Main Road
     2 | High street
     1 | Low Street
(3 rows)
```

15.5.2 Изменение представления

Представление не является неизменяемым, и в нем не содержится „реальных данных“. Это означает, что Вы можете легко изменять его, не воздействуя на данные в Вашей базе данных:

```
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;
```

(Этот пример также показывает использование общепринятого правила написания всех ключевых слов SQL в ВЕРХНЕМ РЕГИСТРЕ).

Вы увидите, что когда мы добавили предложение `ORDER BY` , строки в нашем представлении стали красиво отсортированы:

```
select * from roads_count_v;

 count |     name
-------+-------------
     2 | High street
     1 | Low Street
     1 | Main Road
(3 rows)
```

15.5.3 Удаление представления

Если Вам больше не нужно представление, Вы можете удалить его таким способом:

```
drop view roads_count_v;
```

15.5.4 In Conclusion

Используя представление, Вы можете сохранить запрос и использовать результаты его выполнения так, как если бы это была таблица.

15.5.5 What's Next?

Иногда, при изменении данных, вам хочется, чтобы Ваши изменения отразились где-то еще в базе данных. В следующем уроке будет показано, как это сделать.

15.6 Lesson: Rules

Rules allow the «query tree» of an incoming query to be rewritten. One common usage is to implement views, including updatable view. - *Wikipedia*

The goal for this lesson: To learn how to create new rules for the database.

15.6.1 Creating a logging rule

Say you want to log every change of phone_no in your people table in to a people_log table. So you set up a new table:

```
create table people_log (name text, time timestamp default NOW());
```

In the next step, create a rule that logs every change of a phone_no in the people table into the people_log table:

```
create rule people_log as on update to people
  where NEW.phone_no <> OLD.phone_no
  do insert into people_log values (OLD.name);
```

To test that the rule works, let's modify a phone number:

```
update people set phone_no = '082 555 1234' where id = 2;
```

Check that the `people` table was updated correctly:

```
select * from people where id=2;

 id |    name     | house_no | street_id |   phone_no
----+-------------+----------+-----------+--------------
  2 | Joe Bloggs |        3 |         2 | 082 555 1234
(1 row)
```

Now, thanks to the rule we created, the `people_log` table will look like this:

```
select * from people_log;

    name     |            time
-------------+----------------------------
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 Joe Bloggs | 2014-01-11 14:15:11.953141
(1 row)
```

Примечание: The value of the `time` field will depend on the current date and time.

15.6.2 In Conclusion

Rules allow you to automatically add or change data in your database to reflect changes in other parts of the database.

15.6.3 What's Next?

The next module will introduce you to Spatial Database using PostGIS, which takes these database concepts and applies them to GIS data.

ГЛАВА 16

Module: Spatial Database Concepts with PostGIS

Spatial Databases allow the storage of the geometries of records inside a Database as well as providing functionality for querying and retrieving the records using these Geometries. In this module we will use PostGIS, an extension to PostgreSQL, to learn how to setup a spatial database, import data into the database and make use of the geographic functions that PostGIS offers.

While working through this section, you may want to keep a copy of the **PostGIS cheat sheet** available from Boston GIS user group. Another useful resource is the online PostGIS documentation.

There are also some more extensive tutorials on PostGIS and Spatial Databases available from Boundless Geo:

- Introduction to PostGIS
- Spatial Database Tips and Tricks

See also PostGIS In Action.

16.1 Lesson: PostGIS Setup

Setting up PostGIS functions will allow you to access spatial functions from within PostgreSQL.

The goal for this lesson: To install spatial functions and briefly demo their effects.

Примечание: We will assume the use of PostGIS version 2.1 or newer in this exercise. The installation and database configuration are different for older versions, but the rest of this material in this module will still work. Consult the documentation for your platform for help with installation and database configuration.

16.1.1 Installing under Ubuntu

Postgis is easily installed from apt.

```
$ sudo apt install postgresql
$ sudo apt install postgis
```

Серьёзно, это так просто…

Примечание: The exact versions that will be installed depend on which version of Ubuntu you are using and which repositories you have configured. After installing you can check the version by issuing a `select PostGIS_full_version();` query with psql or another tool.

To install a specific version (eg, PostgreSQL version 13 and PostGIS 3), you can use the following commands.

```
$ sudo apt install wget ca-certificates
$ sudo lsb_release -a
$ wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-
↪key add -
$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-
↪pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
$ sudo apt-get update
$ sudo apt install postgis postgresql-13-postgis-3
```

16.1.2 Installing under Windows

Installing on Windows can be done from binary packages using a normal Windows installation dialogs.

First Visit the download page. Then follow this guide.

More information about installing on Windows can be found on the PostGIS website.

16.1.3 Installing on Other Platforms

The PostGIS website download has information about installing on other platforms including macOS and on other Linux distributions

16.1.4 Configuring Databases to use PostGIS

Once PostGIS is installed, you will need to configure your database to use the extensions. If you have installed PostGIS version > 2.0, this is as simple as issuing the following command with psql using the address database from our previous exercise.

```
$ psql -d address -c "CREATE EXTENSION postgis;"
```

Примечание: Depending on your version, you could find more instructions on how to spatially enable a database at https://postgis.net/docs/postgis_administration.html#create_spatial_db.

16.1.5 Looking at the installed PostGIS functions

PostGIS can be thought of as a collection of in-database functions that extend the core capabilities of PostgreSQL so that it can deal with spatial data. By „deal with“, we mean store, retrieve, query and manipulate. In order to do this, a number of functions are installed into the database.

Our PostgreSQL `address` database is now geospatially enabled, thanks to PostGIS. We are going to delve a lot deeper into this in the coming sections, but let's give you a quick little taster. Let's say we want to create a point from text. First we use the psql command to find functions relating to point. If you are not already connected to the `address` database, do so now. Then run:

```
\df *point*
```

This is the command we're looking for: `st_pointfromtext`. To page through the list, use the down arrow, then press `Q` to quit back to the psql shell.

Try running this command:

```
select st_pointfromtext('POINT(1 1)');
```

Результат:

```
st_pointfromtext
----------------------------------------------
0101000000000000000000F03F000000000000F03F
(1 row)
```

Three things to note:

- We defined a point at position 1,1 (EPSG:4326 is assumed) using `POINT(1 1)`,
- We ran an sql statement, but not on any table, just on data entered from the SQL prompt,
- The resulting row does not make much sense.

The resulting row is in the OGC format called „Well Known Binary“ (WKB). We will look at this format in detail in the next section.

To get the results back as text, we can do a quick scan through the function list for something that returns text:

```
\df *text
```

The query we're looking for now is `st_astext`. Let's combine it with the previous query:

```
select st_astext(st_pointfromtext('POINT(1 1)'));
```

Результат:

```
 st_astext
------------
  POINT(1 1)
   (1 row)
```

Here, we entered the string `POINT(1,1)`, turned it into a point using `st_pointfromtext()`, and turned it back into a human-readable form with `st_astext()`, which gave us back our original string.

One last example before we really get into the detail of using PostGIS:

```
select st_astext(st_buffer(st_pointfromtext('POINT(1 1)'),1.0));
```

What did that do? It created a buffer of 1 degree around our point, and returned the result as text.

16.1.6 Spatial Reference Systems

In addition to the PostGIS functions, the extension contains a collection of spatial reference system (SRS) definitions as defined by the European Petroleum Survey Group (EPSG). These are used during operations such as coordinate reference system (CRS) conversions.

We can inspect these SRS definitions in our database as they are stored in normal database tables.

First, let's look at the schema of the table by entering the following command in the psql prompt:

```
\d spatial_ref_sys
```

The result should be this:

```
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)
```

You can use standard SQL queries (as we have learned from our introductory sections), to view and manipulate this table - though its not a good idea to update or delete any records unless you know what you are doing.

One SRID you may be interested in is EPSG:4326 - the geographic / lat lon reference system using the WGS 84 ellipsoid. Let's take a look at it:

```
select * from spatial_ref_sys where srid=4326;
```

Результат:

```
srid      | 4326
auth_name | EPSG
auth_srid | 4326
srtext    | GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS
84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],TOWGS84[0,
0,0,0,0,0,0],AUTHORITY["EPSG","6326"]],PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],UNIT["degree",0.01745329251994328,
AUTHORITY["EPSG","9122"]],AUTHORITY["EPSG","4326"]]
proj4text | +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
```

The `srtext` is the projection definition in well known text (you may recognise this from .prj files in your shapefile collection).

16.1.7 In Conclusion

You now have PostGIS functions installed in your copy of PostgreSQL. With this you'll be able to make use of PostGIS" extensive spatial functions.

16.1.8 What's Next?

Next you'll learn how spatial features are represented in a database.

16.2 Lesson: Simple Feature Model

How can we store and represent geographic features in a database? In this lesson we'll cover one approach, the Simple Feature Model as defined by the OGC.

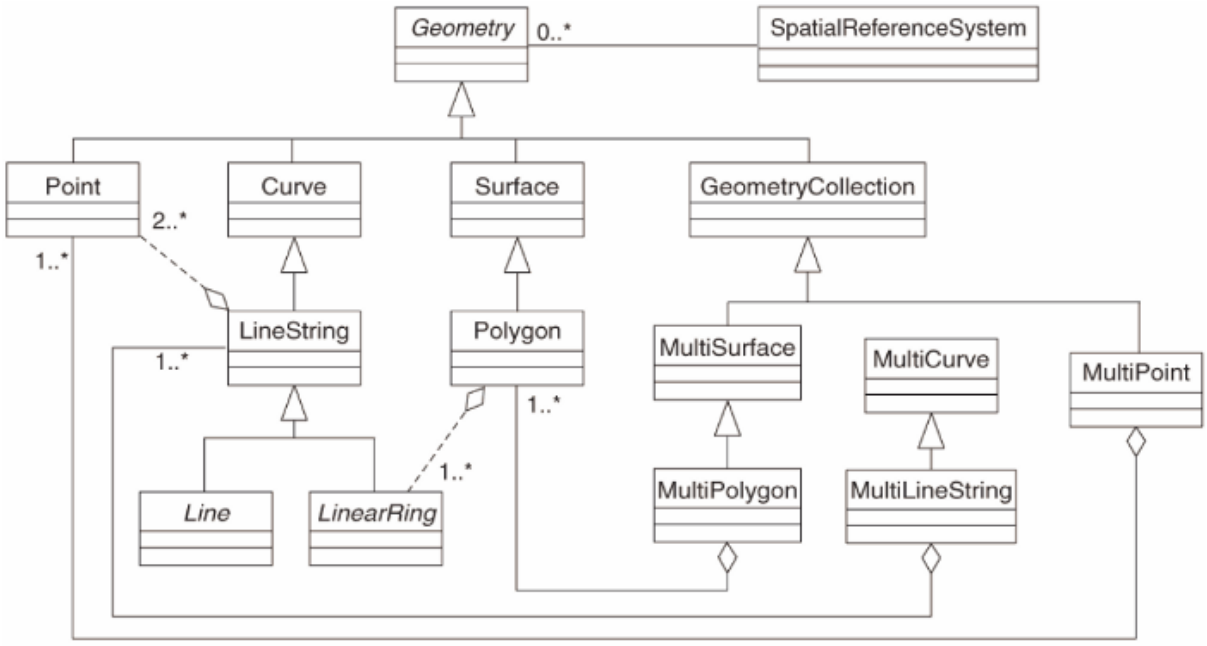
The goal for this lesson: To learn what the SFS Model is and how to use it.

16.2.1 What is OGC

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 What is the SFS Model

The Simple Feature for SQL (SFS) Model is a *non-topological* way to store geospatial data in a database and defines functions for accessing, operating, and constructing these data.



The model defines geospatial data from Point, Linestring, and Polygon types (and aggregations of them to Multi objects).

For further information, have a look at the OGC Simple Feature for SQL standard.

16.2.3 Add a geometry field to table

Let's add a point field to our people table:

```
alter table people add column the_geom geometry;
```

16.2.4 Add a constraint based on geometry type

You will notice that the geometry field type does not implicitly specify what *type* of geometry for the field - for that we need a constraint:

```
alter table people
add constraint people_geom_point_chk
    check(st_geometrytype(the_geom) = 'ST_Point'::text
          OR the_geom IS NULL);
```

This adds a constraint to the table so that it will only accept a point geometry or a null value.

16.2.5 Try Yourself 

Create a new table called cities and give it some appropriate columns, including a geometry field for storing polygons (the city boundaries). Make sure it has a constraint enforcing geometries to be polygons.

Answer

```
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 );
```

16.2.6 Populate geometry_columns table

At this point you should also add an entry into the `geometry_columns` table:

```
insert into geometry_columns values
  ('','public','people','the_geom',2,4326,'POINT');
```

Why? `geometry_columns` is used by certain applications to be aware of which tables in the database contain geometry data.

Примечание: If the above `INSERT` statement causes an error, run this query first:

```
select * from geometry_columns;
```

If the column `f_table_name` contains the value `people`, then this table has already been registered and you don't need to do anything more.

The value `2` refers to the number of dimensions; in this case, two: **X** and **Y**.

The value `4326` refers to the projection we are using; in this case, WGS 84, which is referred to by the number 4326 (refer to the earlier discussion about the EPSG).

Try Yourself 

Add an appropriate *geometry_columns* entry for your new cities layer

Answer

```
insert into geometry_columns values
      ('','public','cities','the_geom',2,4326,'POLYGON');
```

16.2.7 Add geometry record to table using SQL

Now that our tables are geo-enabled, we can store geometries in them:

```
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)');
```

Примечание: In the new entry above, you will need to specify which projection (SRID) you want to use. This is because you entered the geometry of the new point using a plain string of text, which does not automatically add the correct projection information. Obviously, the new point needs to use the same SRID as the data-set it is being added to, so you need to specify it.

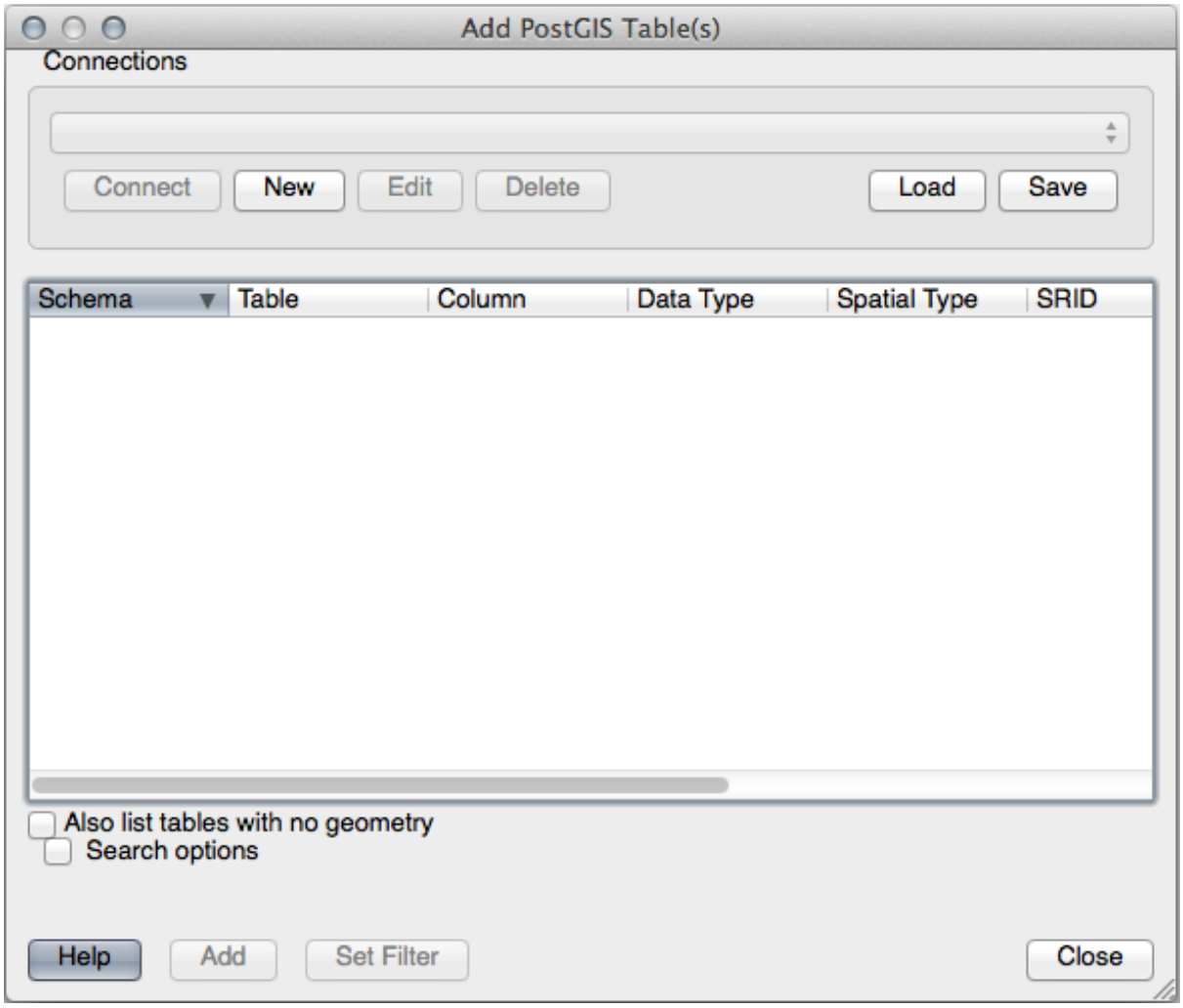
If at this point you were using a graphical interface, for example, specifying the projection for each point would be automatic. In other words, you usually won't need to worry about using the correct projection for every point you want to add if you've already specified it for that data-set, as we did earlier.

Now is probably a good time to open QGIS and try to view your `people` table. Also, we should try editing / adding / deleting records and then performing select queries in the database to see how the data has changed.

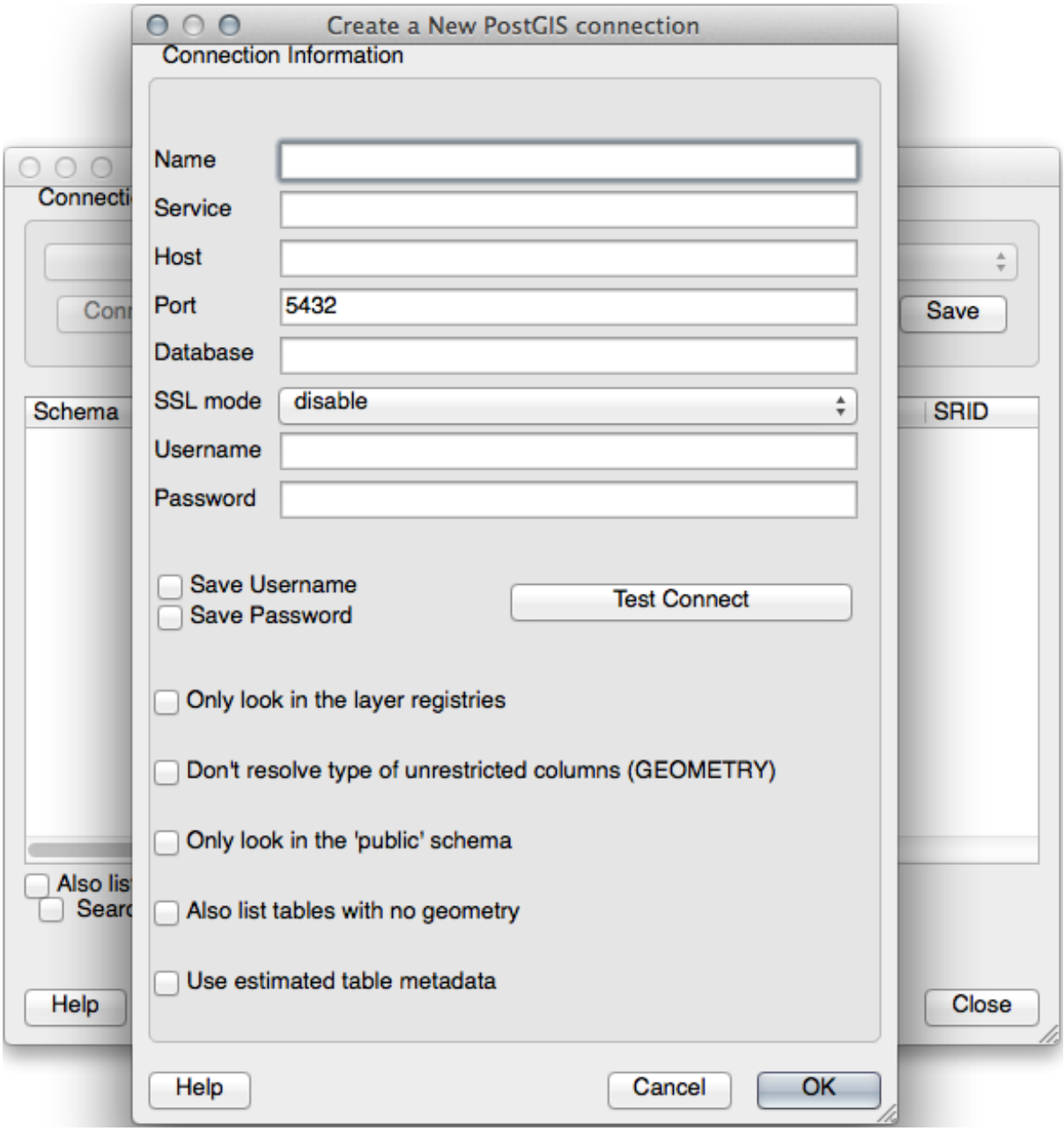
To load a PostGIS layer in QGIS, use the *Layer* ► *Add PostGIS Layers* menu option or toolbar button:



This will open the dialog:



Click on the *New* button to open this dialog:



Then define a new connection, e.g.:

```
Name: myPG
Service:
Host: localhost
Port: 5432
Database: address
User:
Password:
```

To see whether QGIS has found the `address` database and that your username and password are correct, click *Test Connect*. If it works, check the boxes next to *Save Username* and *Save Password*. Then click *OK* to create this connection.

Back in the *Add PostGIS Layers* dialog, click *Connect* and add layers to your project as usual.

Try Yourself 

Formulate a query that shows a person's name, street name and position (from the the_geom column) as plain text.

Answer

```
select people.name,
       streets.name as street_name,
       st_astext(people.the_geom) as geometry
from   streets, people
where  people.street_id=streets.id;
```

Result:

```
     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)
```

As you can see, our constraint allows nulls to be added into the database.

16.2.8 In Conclusion

You have seen how to add spatial objects to your database and view them in GIS software.

16.2.9 What's Next?

Next you'll see how to import data into, and export data from, your database.

16.3 Lesson: Import and Export

Of course, a database with no easy way to migrate data into it and out of it would not be of much use. Fortunately, there are a number of tools that will let you easily move data into and out of PostGIS.

16.3.1 shp2pgsql

shp2pgsql is a commandline tool to import ESRI Shapefile to the database. Under Unix, you can use the following command for importing a new PostGIS table:

```
shp2pgsql -s <SRID> -c -D -I <path to shapefile> <schema>.<table> | \
  psql -d <databasename> -h <hostname> -U <username>
```

Under Windows, you have to perform the import process in two steps:

```
shp2pgsql -s <SRID> -c -D -I <path to shapefile> <schema>.<table> > import.sql
psql psql -d <databasename> -h <hostname> -U <username> -f import.sql
```

You may encounter this error:

```
ERROR:  operator class "gist_geometry_ops" does not exist for access method
"gist"
```

This is a known issue regarding the creation *in situ* of a spatial index for the data you're importing. To avoid the error, exclude the `-I` parameter. This will mean that no spatial index is being created directly, and you'll need to create it in the database after the data have been imported. (The creation of a spatial index will be covered in the next lesson.)

16.3.2 pgsql2shp

pgsql2shp is a commandline tool to export PostGIS Tables, Views or SQL select queries. To do this under Unix:

```
pgsql2shp -f <path to new shapefile> -g <geometry column name> \
  -h <hostname> -U <username> <databasename> <table | view>
```

To export the data using a query:

```
pgsql2shp -f <path to new shapefile> -g <geometry column name> \
  -h <hostname> -U <username> "<query>"
```

16.3.3 ogr2ogr

ogr2ogr is a very powerful tool to convert data into and from postgis to many data formats. ogr2ogr is part of the GDAL/OGR Software and has to be installed separately. To export a table from PostGIS to GML, you can use this command:

```
ogr2ogr -f GML export.gml PG:'dbname=<databasename> user=<username>
        host=<hostname>' <Name of PostGIS-Table>
```

16.3.4 DB Manager

You may have noticed another option in the *Database* menu labeled *DB Manager*. This is a tool that provides a unified interface for interacting with spatial databases including PostGIS. It also allows you to import and export from databases to other formats. Since the next module is largely devoted to using this tool, we will only briefly mention it here.

16.3.5 In Conclusion

Importing and exporting data to and from the database can be done in many various ways. Especially when using disparate data sources, you will probably use these functions (or others like them) on a regular basis.

16.3.6 What's Next?

Next we'll look at how to query the data we've created before.

16.4 Lesson: Spatial Queries

Spatial queries are no different from other database queries. You can use the geometry column like any other database column. With the installation of PostGIS in our database, we have additional functions to query our database.

The goal for this lesson: To see how spatial functions are implemented similarly to «normal» non-spatial functions.

16.4.1 Spatial Operators

When you want to know which points are within a distance of 2 degrees to a point(X,Y) you can do this with:

```
select *
from people
where st_distance(the_geom,'SRID=4326;POINT(33 -34)') < 2;
```

Результат:

```
 id |     name      | house_no | street_id |   phone_no     |   the_geom
----+---------------+----------+-----------+----------------+---------------
  6 | Fault Towers |       34 |         3 | 072 812 31 28 | 01010008040C0
(1 row)
```

Примечание: the_geom value above was truncated for space on this page. If you want to see the point in human-readable coordinates, try something similar to what you did in the section «View a point as WKT», above.

How do we know that the query above returns all the points within 2 *degrees*? Why not 2 *meters*? Or any other unit, for that matter?

Answer

The units being used by the example query are degrees, because the CRS that the layer is using is WGS 84. This is a Geographic CRS, which means that its units are in degrees. A Projected CRS, like the UTM projections, is in meters.

Remember that when you write a query, you need to know which units the layer's CRS is in. This will allow you to write a query that will return the results that you expect.

16.4.2 Spatial Indexes

We also can define spatial indexes. A spatial index makes your spatial queries much faster. To create a spatial index on the geometry column use:

```
CREATE INDEX people_geo_idx
  ON people
  USING gist
  (the_geom);

\d people
```

Результат:

```
Table "public.people"
   Column    |          Type          |                 Modifiers
 ------------+------------------------+-------------------------------------------
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  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 

Modify the cities table so its geometry column is spatially indexed.

Answer

```
CREATE INDEX cities_geo_idx
  ON cities
  USING gist (the_geom);
```

16.4.4 PostGIS Spatial Functions Demo

In order to demo PostGIS spatial functions, we'll create a new database containing some (fictional) data.

To start, create a new database (exit the psql shell first):

```
createdb postgis_demo
```

Remember to install the postgis extensions:

```
psql -d postgis_demo -c "CREATE EXTENSION postgis;"
```

Next, import the data provided in the `exercise_data/postgis/` directory. Refer back to the previous lesson for instructions, but remember that you'll need to create a new PostGIS connection to the new database. You can import from the terminal or via DB Manager. Import the files into the following database tables:

- `points.shp` into `building`
- `lines.shp` into `road`
- `polygons.shp` into `region`

Load these three database layers into QGIS via the *Add PostGIS Layers* dialog, as usual. When you open their attribute tables, you'll note that they have both an `id` field and a `gid` field created by the PostGIS import.

Now that the tables are imported, we can use PostGIS to query the data. Go back to your terminal (command line) and enter the psql prompt by running:

```
psql postgis_demo
```

We'll demo some of these select statements by creating views from them, so that you can open them in QGIS and see the results.

Select by location

Get all the buildings in the KwaZulu region:

```
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';
```

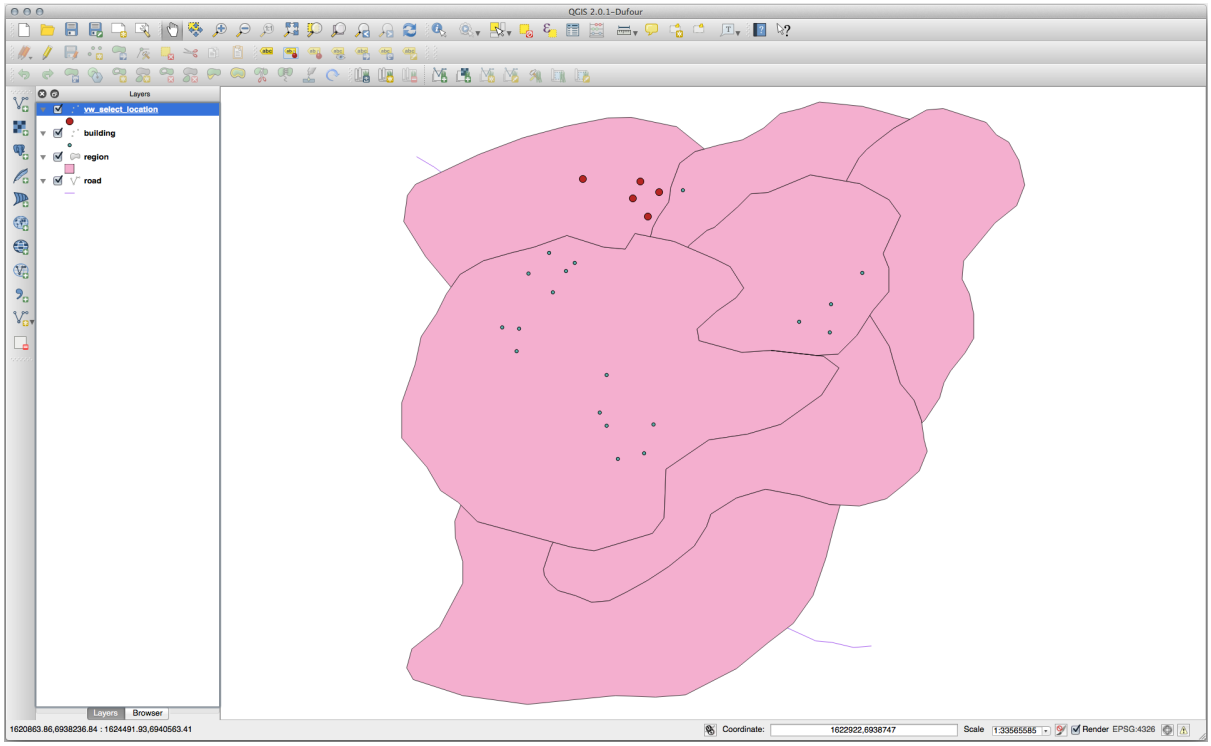
Результат:

```
 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)
```

Or, if we create a view from it:

```
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';
```

Add the view as a layer and view it in QGIS:



Select neighbors

Show a list of all the names of regions adjoining the Hokkaido region:

```
SELECT b.name
  FROM region a, region b
    WHERE st_touches(a.the_geom, b.the_geom)
    AND a.name = 'Hokkaido';
```

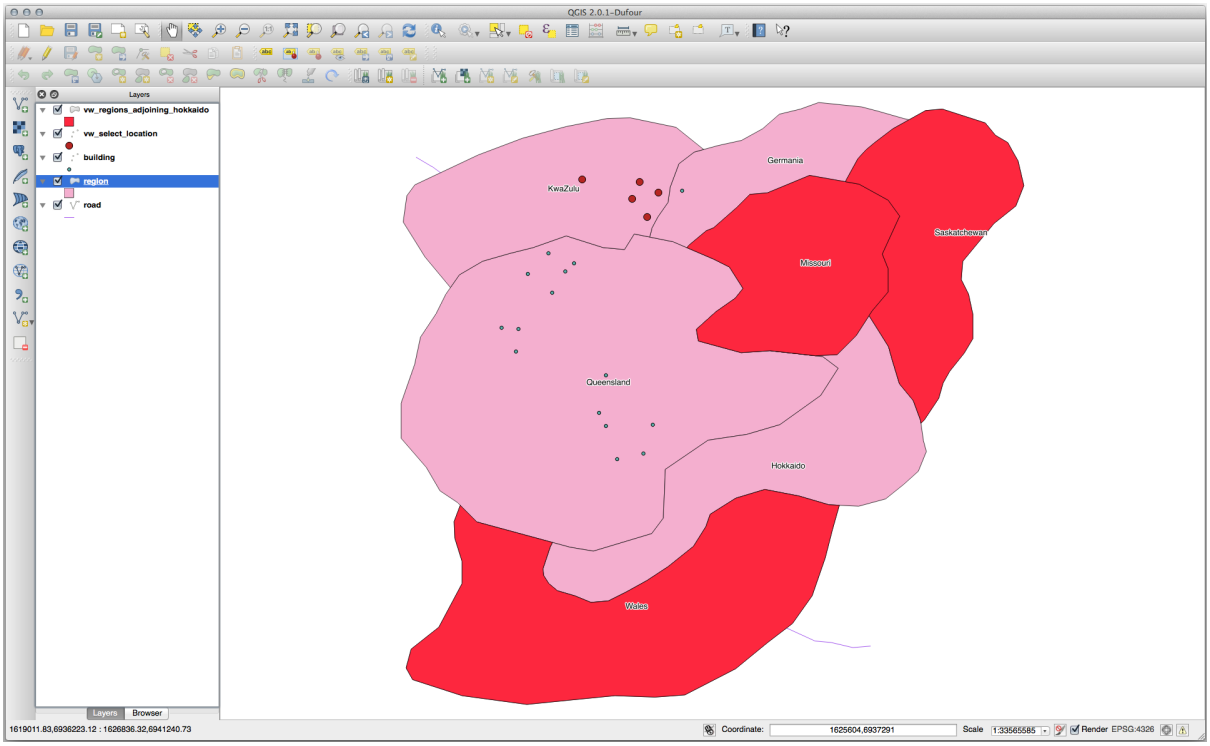
Результат:

```
    name
--------------
 Missouri
 Saskatchewan
 Wales
(3 rows)
```

As a view:

```
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';
```

In QGIS:



Note the missing region (Queensland). This may be due to a topology error. Artifacts such as this can alert us to potential problems in the data. To solve this enigma without getting caught up in the anomalies the data may have, we could use a buffer intersect instead:

```
CREATE VIEW vw_hokkaido_buffer AS
  SELECT gid, ST_BUFFER(the_geom, 100) as the_geom
```

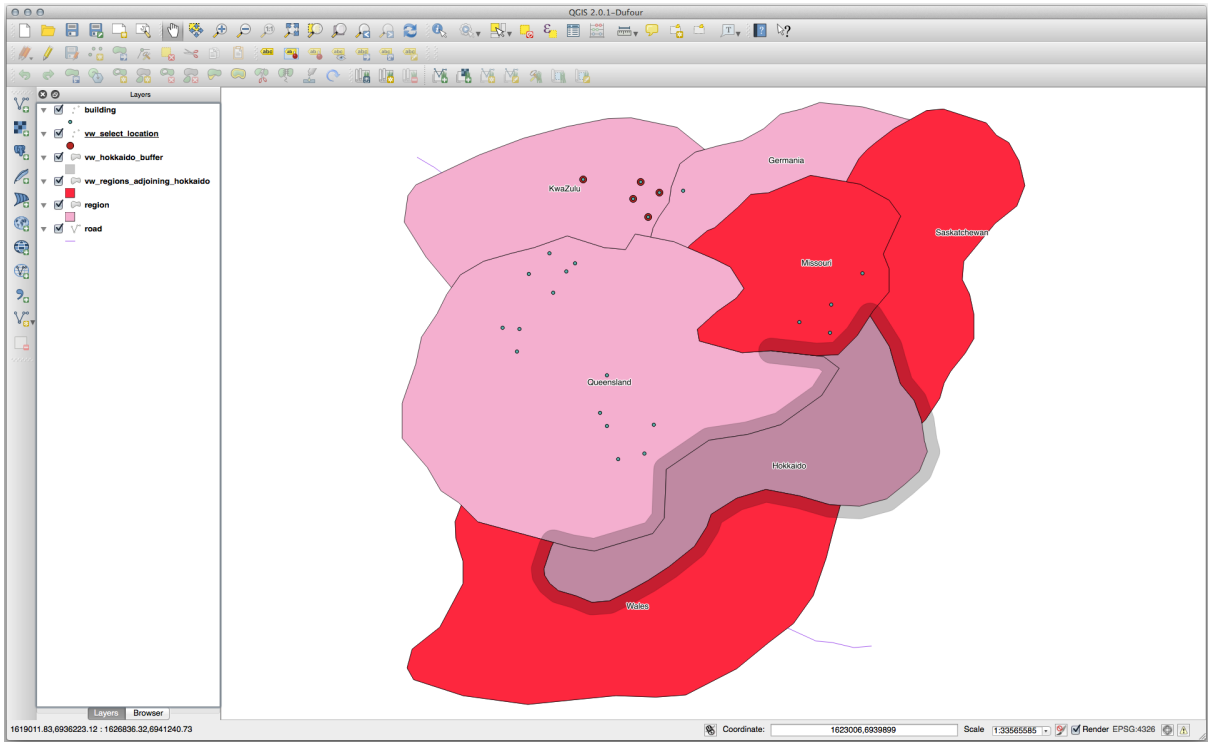
(continues on next page)

(продолжение с предыдущей страницы)

```
    FROM region
      WHERE name = 'Hokkaido';
```

This creates a buffer of 100 meters around the region Hokkaido.

The darker area is the buffer:

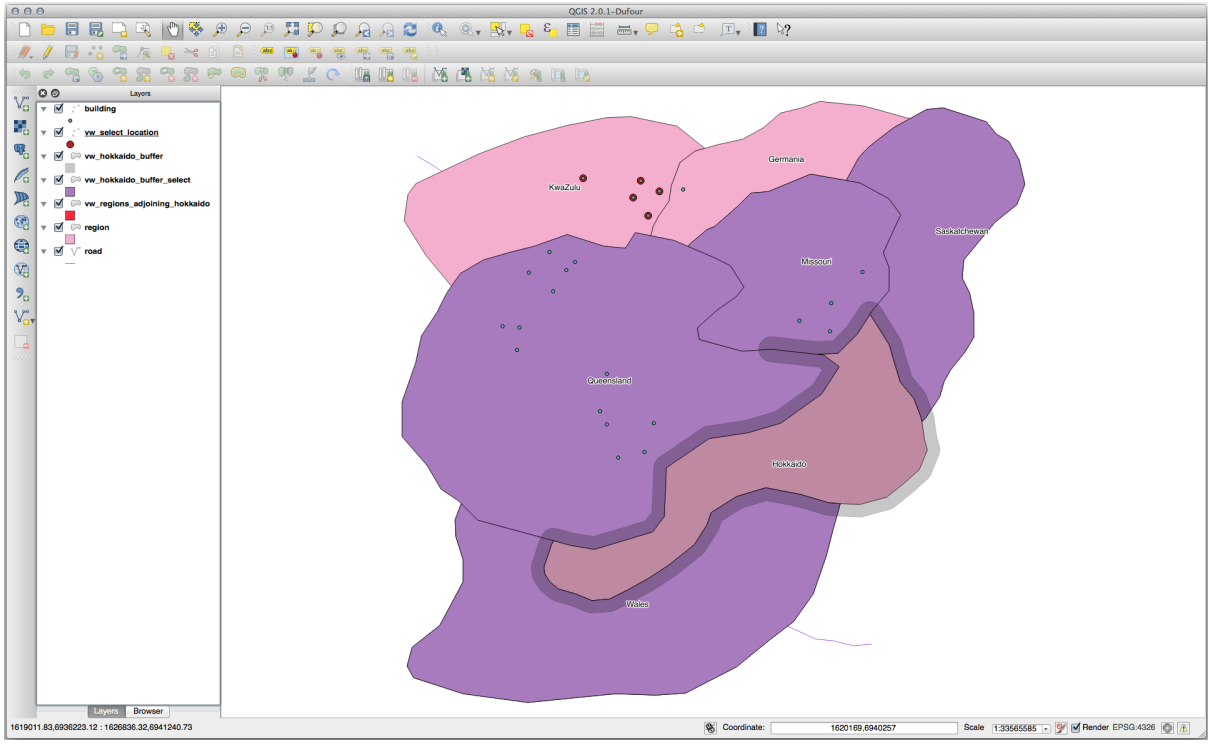


Select using the buffer:

```
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';
```

In this query, the original buffer view is used as any other table would be. It is given the alias `a`, and its geometry field, `a.the_geom`, is used to select any polygon in the `region` table (alias `b`) that intersects it. However, Hokkaido itself is excluded from this select statement, because we don't want it; we only want the regions adjoining it.

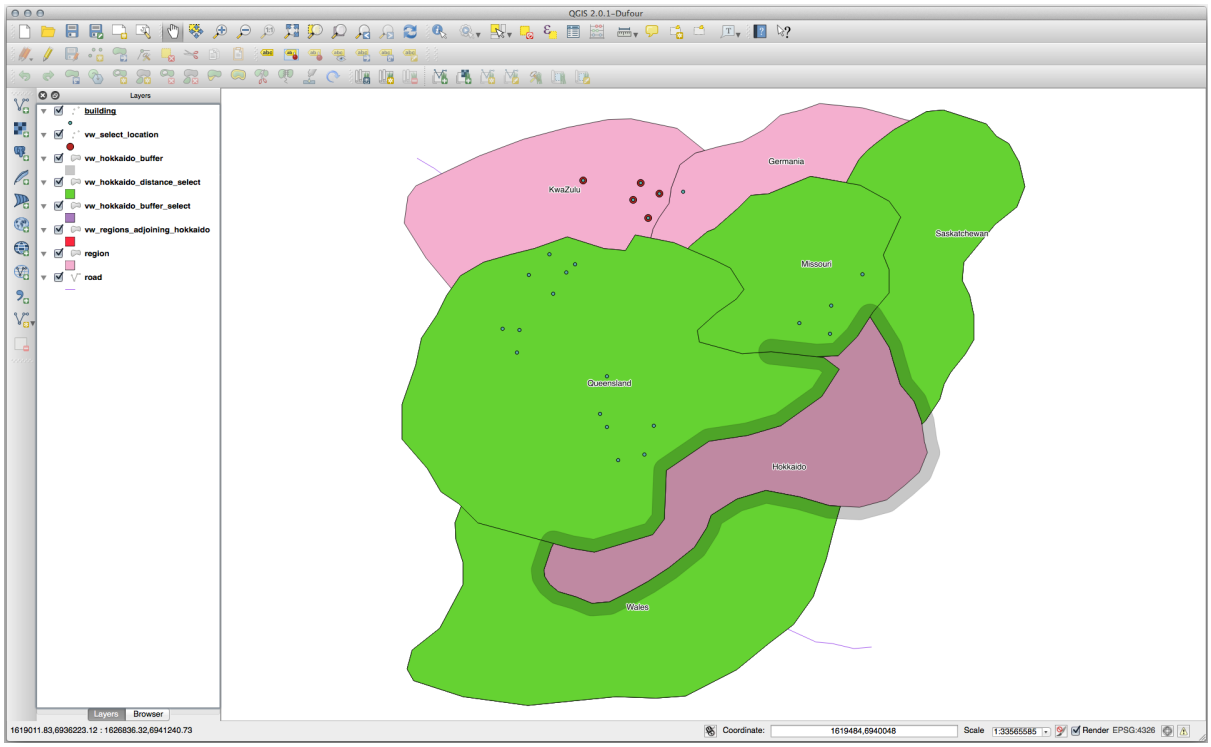
In QGIS:



It is also possible to select all objects within a given distance, without the extra step of creating a buffer:

```
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';
```

This achieves the same result, without need for the interim buffer step:



Select unique values

Show a list of unique town names for all buildings in the Queensland region:

```
SELECT DISTINCT a.name
  FROM building a, region b
    WHERE st_within(a.the_geom, b.the_geom)
    AND b.name = 'Queensland';
```

Результат:

```
  name
---------
 Beijing
 Berlin
 Atlanta
(3 rows)
```

Further examples ...

```
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

You have seen how to query spatial objects using the new database functions from PostGIS.

16.4.6 What's Next?

Next we're going to investigate the structures of more complex geometries and how to create them using PostGIS.

16.5 Lesson: Geometry Construction

In this section we are going to delve a little deeper into how simple geometries are constructed in SQL. In reality, you will probably use a GIS like QGIS to create complex geometries using their digitising tools; however, understanding how they are formulated can be handy for writing queries and understanding how the database is assembled.

The goal of this lesson: To better understand how to create spatial entities directly in PostgreSQL/PostGIS.

16.5.1 Creating Linestrings

Going back to our `address` database, let's get our streets table matching the others; i.e., having a constraint on the geometry, an index and an entry in the geometry_columns table.

16.5.2 Try Yourself 

- Modify the `streets` table so that it has a geometry column of type ST_LineString.
- Don't forget to do the accompanying update to the geometry columns table!
- Also add a constraint to prevent any geometries being added that are not LINESTRINGS or null.
- Create a spatial index on the new geometry column

Answer

```
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);
```

Now let's insert a linestring into our streets table. In this case we will update an existing street record:

```
update streets
set the_geom = 'SRID=4326;LINESTRING(20 -33, 21 -34, 24 -33)'
where streets.id=2;
```

Take a look at the results in QGIS. (You may need to right-click on the streets layer in the „Layers“ panel, and choose „Zoom to layer extent“.)

Now create some more streets entries - some in QGIS and some from the command line.

16.5.3 Creating Polygons

Creating polygons is just as easy. One thing to remember is that by definition, polygons have at least four vertices, with the last and first being co-located:

```
insert into cities (name, the_geom)
values ('Tokyo', 'SRID=4326;POLYGON((10 -10, 5 -32, 30 -27, 10 -10))');
```

Примечание: A polygon requires double brackets around its coordinate list; this is to allow you to add complex polygons with multiple unconnected areas. For instance

```
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))'
        );
```

If you followed this step, you can check what it did by loading the cities dataset into QGIS, opening its attribute table, and selecting the new entry. Note how the two new polygons behave like one polygon.

16.5.4 Exercise: Linking Cities to People

For this exercise you should do the following:

- Delete all data from your people table.
- Add a foreign key column to people that references the primary key of the cities table.
- Use QGIS to capture some cities.
- Use SQL to insert some new people records, ensuring that each has an associated street and city.

Your updated people schema should look something like this:

```
\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)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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)
```

Answer

```
delete from people;
alter table people add column city_id int not null references cities(id);
```

(capture cities in QGIS)

```
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)');
```

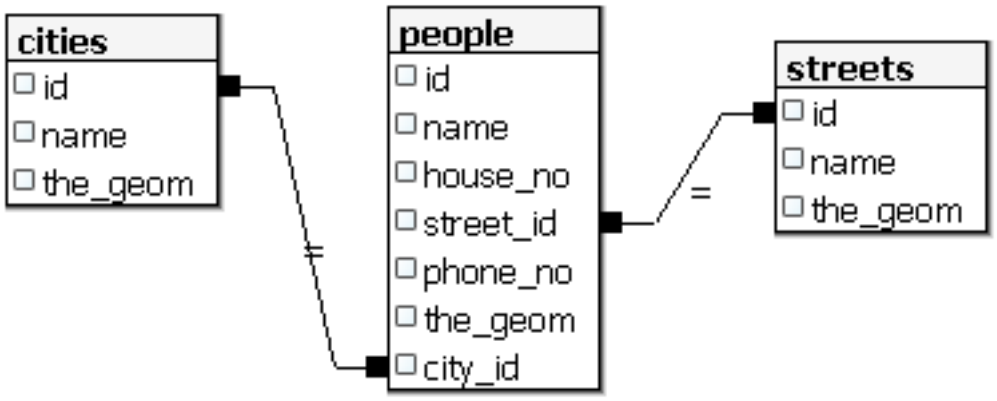
If you're getting the following error message:

```
ERROR:  insert or update on table "people" violates foreign key constraint
        "people_city_id_fkey"
DETAIL: Key (city_id)=(1) is not present in table "cities".
```

then it means that while experimenting with creating polygons for the cities table, you must have deleted some of them and started over. Just check the entries in your cities table and use any *id* which exists.

16.5.5 Looking at Our Schema

By now our schema should be looking like this:



16.5.6 Try Yourself

Create city boundaries by computing the minimum convex hull of all addresses for that city and computing a buffer around that area.

16.5.7 Access Sub-Objects

With the SFS-Model functions, you have a wide variety of options to access sub-objects of SFS Geometries. When you want to select the first vertex point of every polygon geometry in the table myPolygonTable, you have to do this in this way:

- Transform the polygon boundary to a linestring:

```
select st_boundary(geometry) from myPolygonTable;
```

- Select the first vertex point of the resultant linestring:

```
select st_startpoint(myGeometry)
from (
    select st_boundary(geometry) as myGeometry
    from myPolygonTable) as foo;
```

16.5.8 Data Processing

PostGIS supports all OGC SFS/MM standard conform functions. All these functions start with `ST_`.

16.5.9 Clipping

To clip a subpart of your data you can use the `ST_INTERSECT()` function. To avoid empty geometries, use:

```
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 Building Geometries from Other Geometries

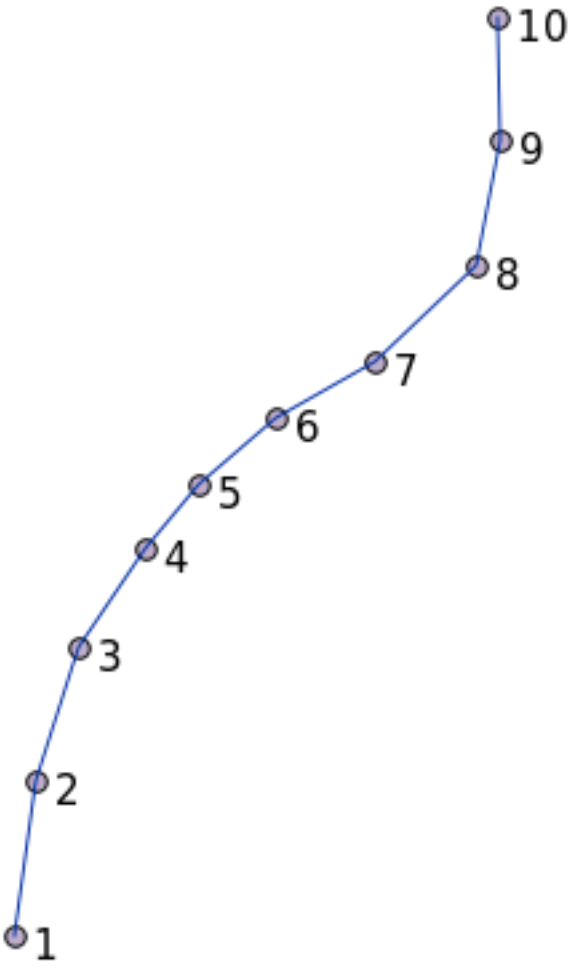
From a given point table, you want to generate a linestring. The order of the points is defined by their `id`. Another ordering method could be a timestamp, such as the one you get when you capture waypoints with a GPS receiver.



To create a linestring from a new point layer called „points“, you can run the following command:

```
select ST_LineFromMultiPoint(st_collect(the_geom)), 1 as id
from (
  select the_geom
  from points
  order by id
) as foo;
```

To see how it works without creating a new layer, you could also run this command on the „people“ layer, although of course it would make little real-world sense to do this.



16.5.11 Geometry Cleaning

You can get more information for this topic in this blog entry.

16.5.12 Differences between tables

To detect the difference between two tables with the same structure, you can use the PostgreSQL keyword `EXCEPT`:

```
select * from table_a
except
select * from table_b;
```

As the result, you will get all records from table_a which are not stored in table_b.

16.5.13 Tablespaces

You can define where postgres should store its data on disk by creating tablespaces:

```
CREATE TABLESPACE homespace LOCATION '/home/pg';
```

When you create a database, you can then specify which tablespace to use e.g.:

```
createdb --tablespace=homespace t4a
```

16.5.14 In Conclusion

You've learned how to create more complex geometries using PostGIS statements. Keep in mind that this is mostly to improve your tacit knowledge when working with geo-enabled databases through a GIS frontend. You usually won't need to actually enter these statements manually, but having a general idea of their structure will help you when using a GIS, especially if you encounter errors that would otherwise seem cryptic.

ГЛАВА 17

The QGIS processing guide

This module contributed by Victor Olaya and Paolo Cavallini.

Contents:

17.1 Введение

This guide describes how to use the QGIS processing framework. It assumes no previous knowledge of the Processing framework or any of the applications that it rely on. It assumes basic knowledge of QGIS. The chapters about scripting assume you have some basic knowledge of Python and maybe the QGIS Python API.

This guide is designed for self-study or to be used for running a processing workshop.

Examples in this guide use QGIS 3.4. They might not work or not be available in versions other than that one.

This guide is comprised of a set of small exercises of progressive complexity. If you have never used the processing framework, you should start from the very beginning. If you have some previous experience, feel free to skip lessons. They are more or less independent of each other, and each one introduces some new concept or some new element, as indicated in the chapter title and the short introduction at the beginning of each chapter. That should make it easy to locate lessons dealing with a particular topic.

For a more systematic description of all the framework components and their usage, it is recommended to check the corresponding chapter in the user manual. Use it as a support text along with this guide.

All the exercises in this guide use the same free dataset used throughout the training manual and referenced at section *Данные*. The zip file to download contains several folders corresponding to each one of the lessons in this guide. In each of them you will find a QGIS project file. Just open it and you will be ready to start the lesson.

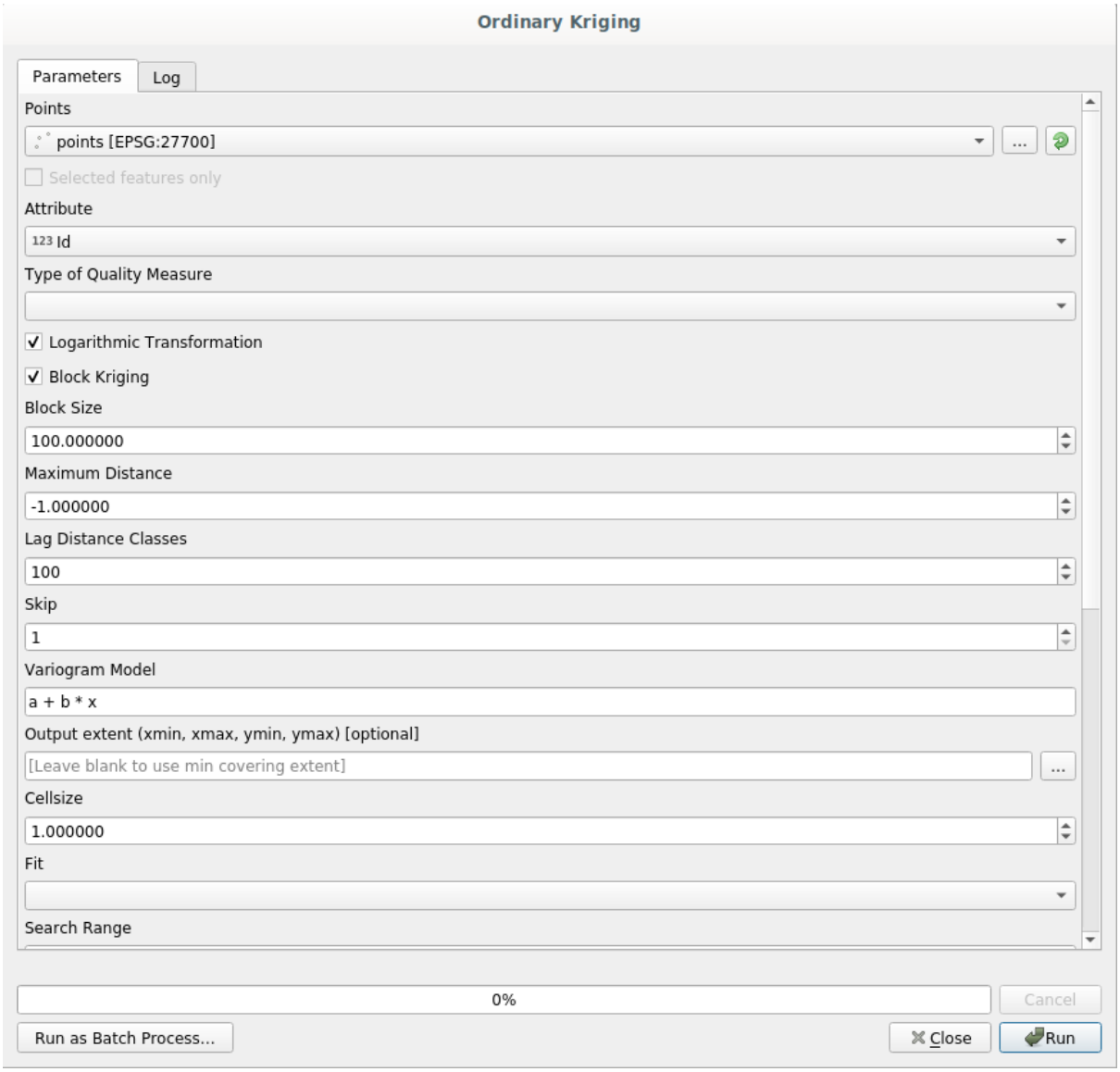
Enjoy!

17.2 An important warning before starting

Just like the manual of a word processor doesn't teach you how to write a novel or a poem, or a CAD tutorial doesn't show you how to calculate the size of a beam for a building, this guide will not teach you spatial analysis. Instead, it will show you how to use the QGIS Processing framework, a powerful tool for performing spatial analysis. It is up to you to learn the required concepts that are needed to understand that type of analysis. Without them, there is no point in using the framework and its algorithms, although you might be tempted to try.

Let's show this more clearly with an example.

Given a set of points and a value of a given variable value at each point, you can calculate a raster layer from them using the *Kriging* geoalgorithm. The parameters dialog for that module is like the following one.



It looks complex, right?

By reading this manual, you will learn things such as how to use that module, how to run it in a batch process to create raster layers from hundreds of points layers in a single run, or what happens if the input layer has some points selected. However, the parameters themselves are not explained. A seasoned analyst with a good knowledge of geostatistics will have no problem understanding those parameters. If you are not one of them and *sill*, *range*, or *nugget* are not familiar concepts to you, then you should not use the *Kriging* module. More than that, you are far from being ready to use the *Kriging* module, since it requires learning about concepts such as spatial autocorrelation or

semivariograms, which probably you also haven't heard before, or at least haven't studied long enough. You should first study and understand them, and then come back to QGIS to actually run it and perform the analysis. Ignoring this will result in wrong results and poor (and most likely useless) analysis.

Although not all algorithms are as complex as kriging (but some of them are even more complex!), almost all of them require understanding the fundamental analysis ideas that they are based on. Without that knowledge, using them will most likely lead to poor results.

Using geoalgorithms without having a good foundation of spatial analysis is like trying to write a novel without knowing anything about grammar or syntax, and having no knowledge about storytelling. You might get a result, but it is likely to have no value at all. Please, don't fool yourself and think that after reading this guide you are already capable of performing spatial analysis and get sound results. You need to study spatial analysis as well.

Here is a good reference that you can read to learn more about spatial data analysis.

Geospatial Analysis (3rd Edition): A Comprehensive Guide to Principles, Techniques and Software Tools Michael John De Smith, Michael F. Goodchild, Paul A. Longley

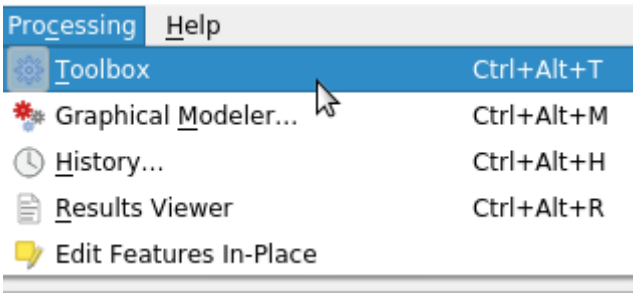
It is available online here

17.3 Setting-up the processing framework

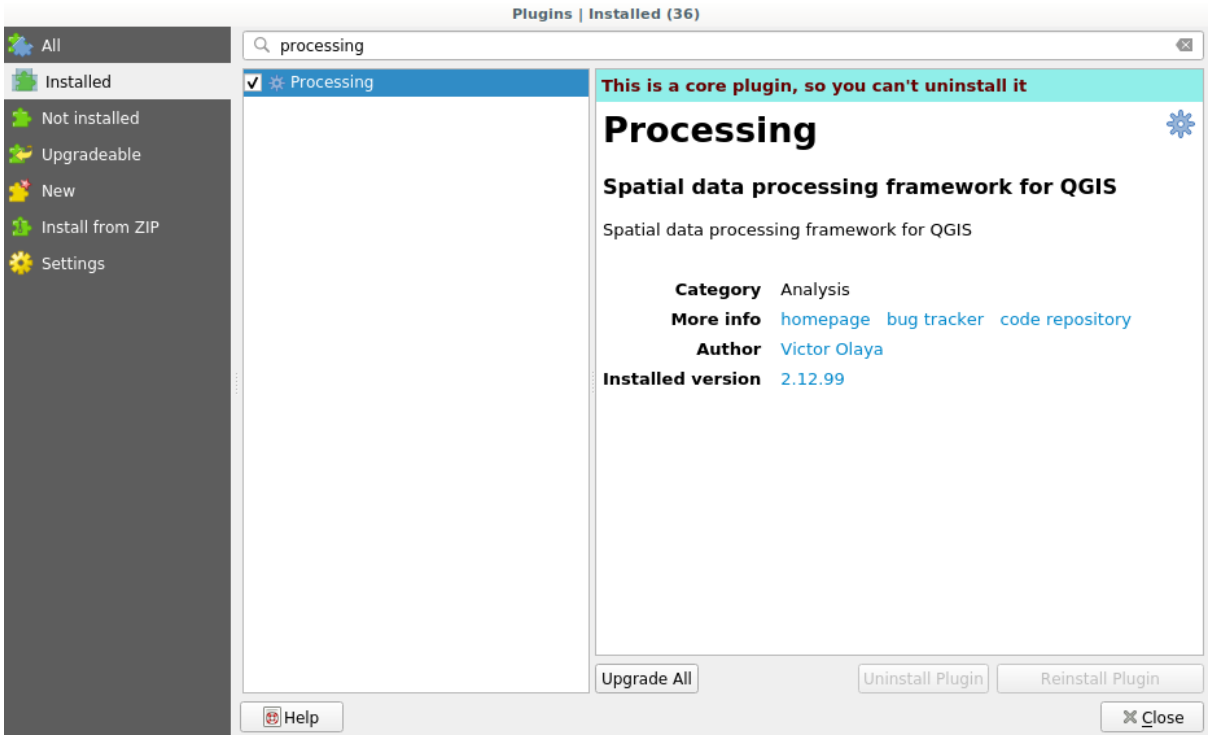
The first thing to do before using the processing framework is to configure it. There is not much to set-up, so this is an easy task.

Later on we will show how to configure the external applications that are used for extending the list of available algorithms, but for now we are just going to work with the framework itself.

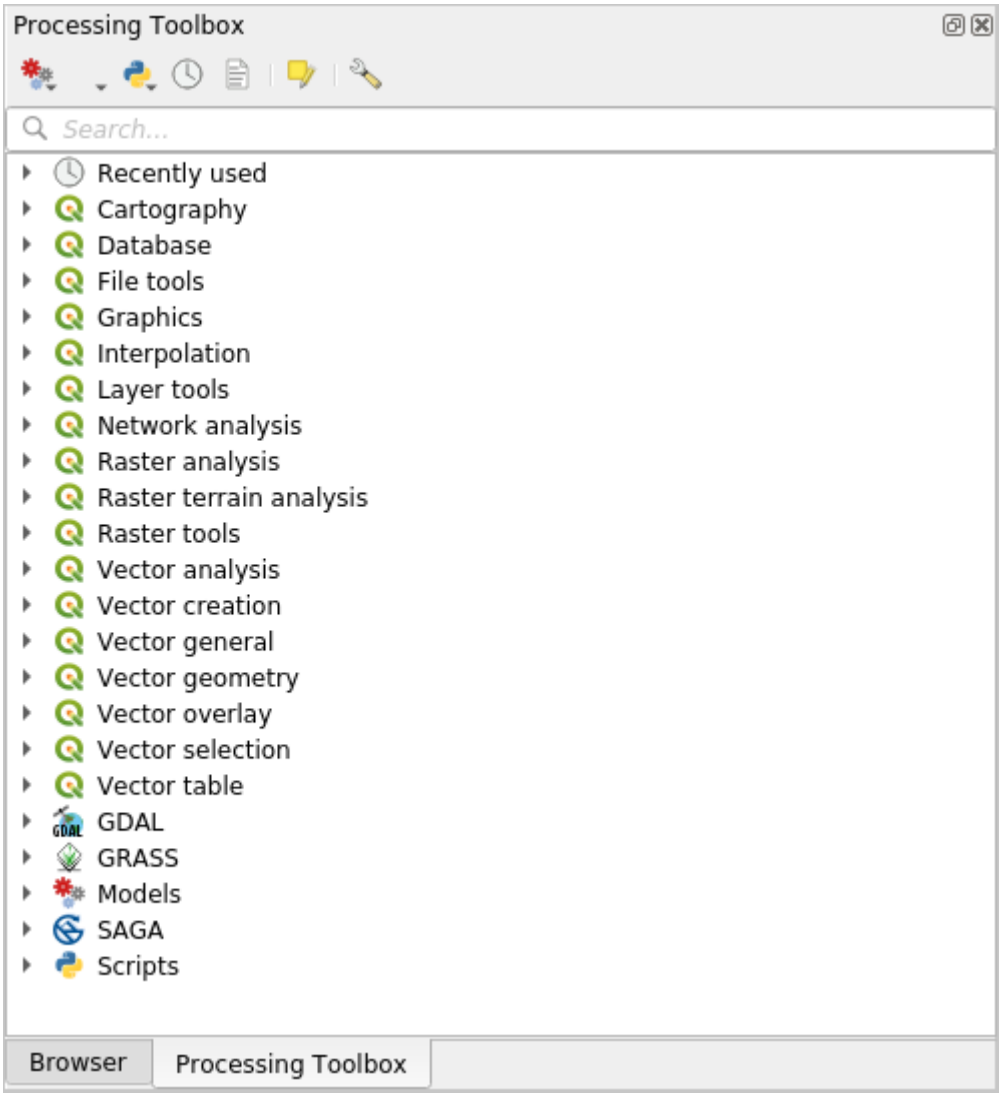
The processing framework is a core QGIS plugin, which means that it should already be installed in your system, since it is included with QGIS. In case it is active, you should see a menu called *Processing* in your menu bar. There you can access all the framework components.



If you cannot find that menu, you have to enable the plugin by going to the plugin manager and activating it.



The main element that we are going to work with is the toolbox. Click on the corresponding menu entry and you will see the toolbox docked at the right side of the QGIS window.



The toolbox contains a list of all the available algorithms, divided in groups called *Providers*. Providers can be (de)activated in the *Settings* ► *Options* ► *Processing*. We will discuss that dialog later in this manual.

By default, only providers that do not rely on third-party applications (that is, those that only require QGIS elements to be run) are active. Algorithms requiring external applications might need additional configuration. Configuring providers is explained in a later chapter in this manual.

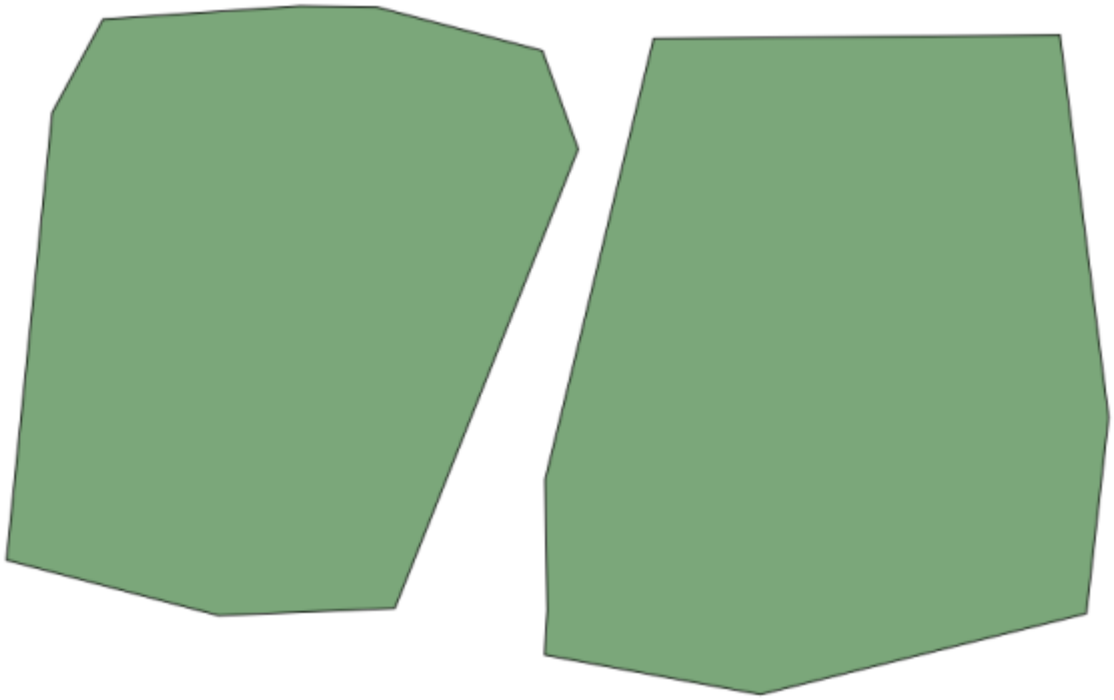
If you have reached this point, now you are ready to use geoalgorithms. There is no need to configure anything else by now. We can already run our first algorithm, which we will do in the next lesson.

17.4 Running our first algorithm. The toolbox

Примечание: In this lesson we will run our first algorithm, and get our first result from it.

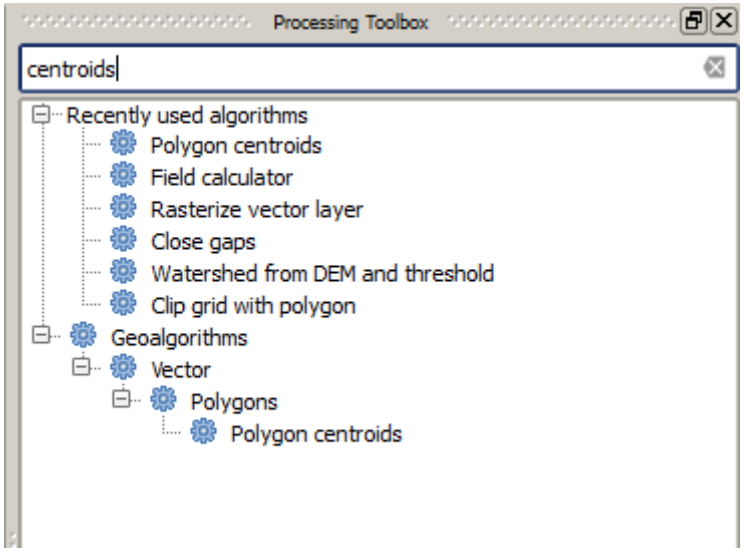
As we have already mentioned, the processing framework can run algorithms from other applications, but it also contains native algorithms that need no external software to be run. To start exploring the processing framework, we are going to run one of those native algorithms. In particular, we are going to calculate the centroids of set of polygons.

First, open the QGIS project corresponding to this lesson. It contains just a single layer with two polygons

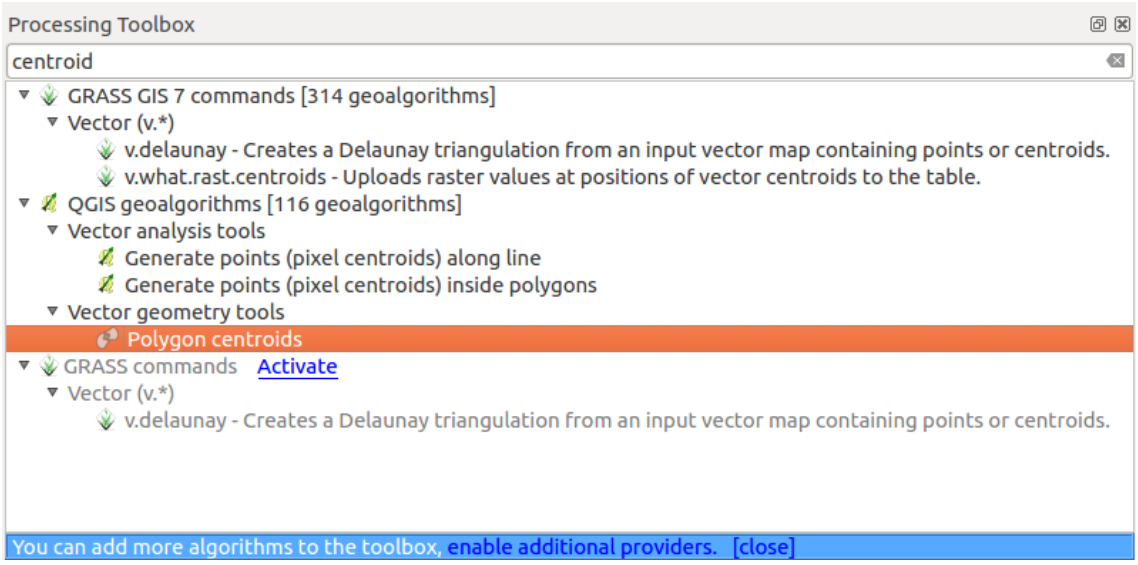


Now go to the text box at the top of the toolbox. That is the search box, and if you type text in it, it will filter the list of algorithms so just those ones containing the entered text are shown. If there are algorithms that match your search but belong to a provider that is not active, an additional label will be shown in the lower part of the toolbox.

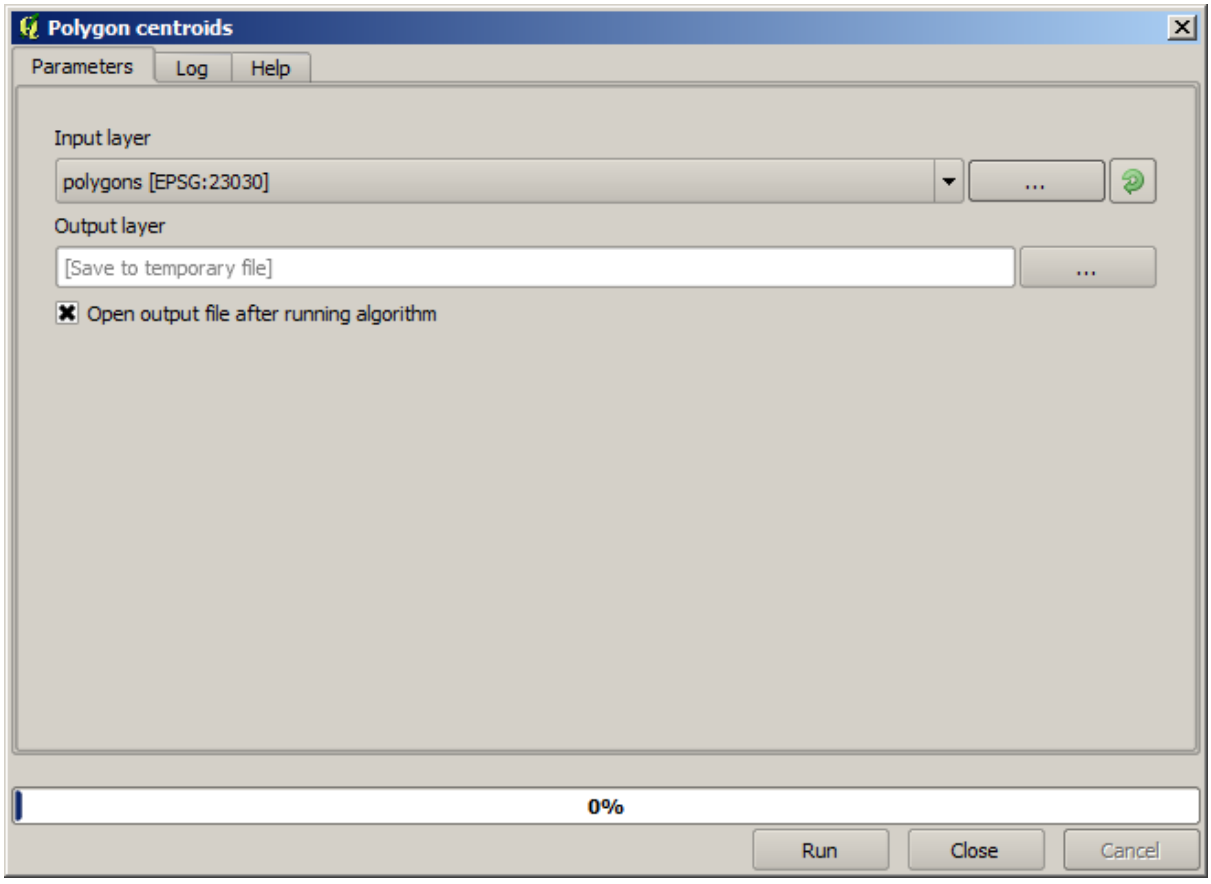
Type `centroids` and you should see something like this.



The search box is a very practical way of finding the algorithm you are looking for. At the bottom of the dialog, an additional label shows that there are algorithms that match your search but belong to a provider that is not active. If you click on the link in that label, the list of algorithms will also include results from those inactive providers, which will be shown in light gray. A link to activate each inactive provider is also shown. We'll see later how to activate other providers.



To execute an algorithm, you just have to double-click on its name in the toolbox. When you double-click on the *Polygon centroids* algorithm, you will see the following dialog.



All algorithms have a similar interface, which basically contains input parameters that you have to fill, and outputs that you have to select where to store. In this case, the only input we have is a vector layer with polygons.

Select the *Polygons* layer as input. The algorithm has a single output, which is the centroids layer. There are two options to define where a data output is saved: enter a filepath or save it to a temporary filename

In case you want to set a destination and not save the result in a temporary file, the format of the output is defined by the filename extension. To select a format, just select the corresponding file extension (or add it if you are directly

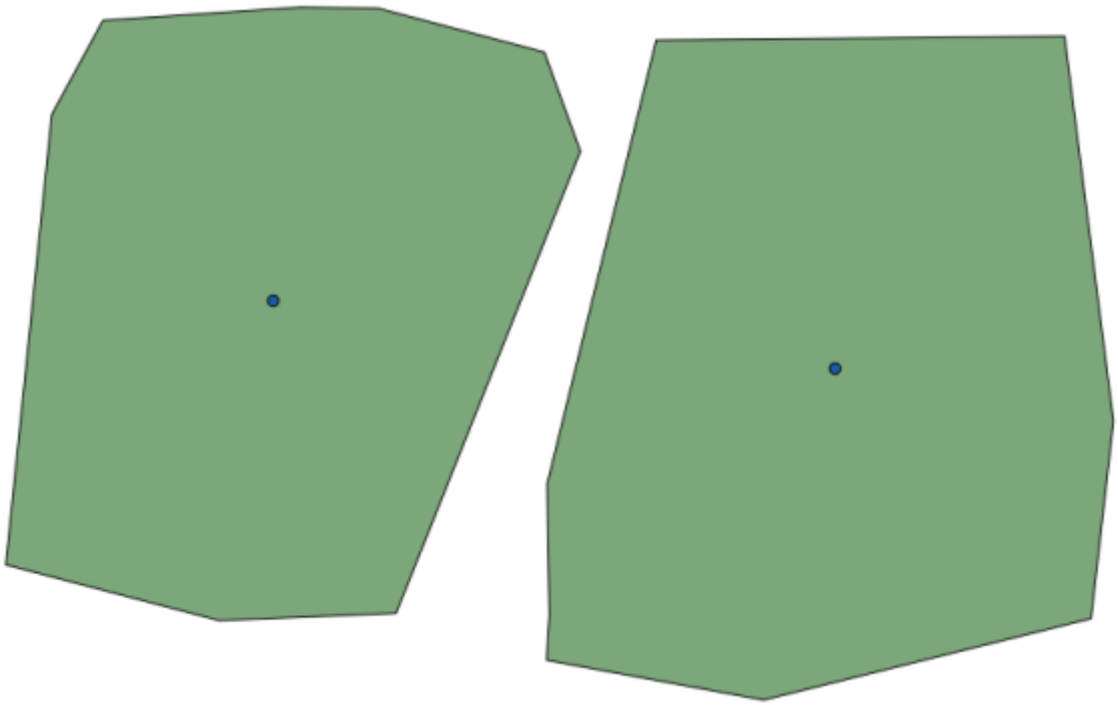
typing the filepath instead). If the extension of the filepath you entered does not match any of the supported ones, a default extension (usually `.dbf` for tables, `.tif` for raster layers and `.shp` for vector ones) will be appended to the filepath and the file format corresponding to that extension will be used to save the layer or table.

In all the exercises in this guide, we will be saving results to a temporary file, since there is no need to save them for a later use. Feel free to save them to a permanent location if you want to.

Предупреждение: Temporary files are deleted once you close QGIS. If you create a project with an output that was saved as a temporary output, QGIS will complain when you try to open back the project later, since that output file will not exist.

Once you have configured the algorithm dialog, press *Run* to run the algorithm.

You will get the following output.



The output has the same CRS as the input. Geoalgorithms assumes all input layers share the same CRS and do not perform any reprojection. Except in the case of some special algorithms (for instance, reprojection ones), the outputs will also have that same CRS. We will see more about this soon.

Try yourself saving it using different file formats (use, for instance, `shp` and `geojson` as extensions). Also, if you do not want the layer to be loaded in QGIS after it is generated, you can check off the checkbox that is found below the output path box.

17.5 More algorithms and data types

Примечание: In this lesson we will run three more algorithms, learn how to use other input types, and configure outputs to be saved to a given folder automatically.

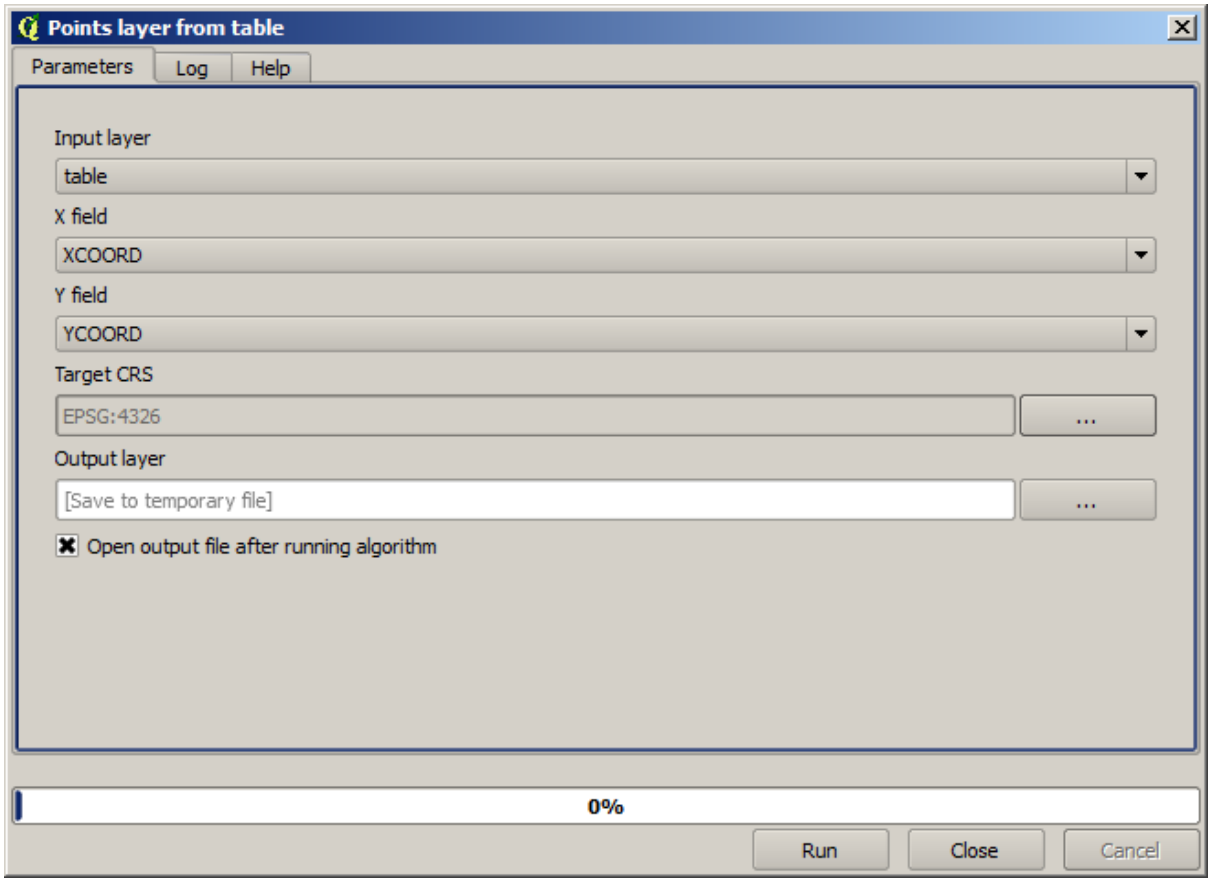
For this lessons we will need a table and a polygons layer. We are going to create a points layer based on coordinates in the table, and then count the number of points in each polygon. If you open the QGIS project corresponding to this lesson, you will find a table with X and Y coordinates, but you will find no polygons layer. Don't worry, we will create it using a processing geoalgorithm.

The first thing we are going to do is to create a points layer from the coordinates in the table, using the *Points layer from table* algorithm. You now know how to use the search box, so it should not be hard for you to find it. Double–click on it to run it and get to its following dialog.

This algorithm, like the one from the previous lesson, just generates a single output, and it has three inputs:

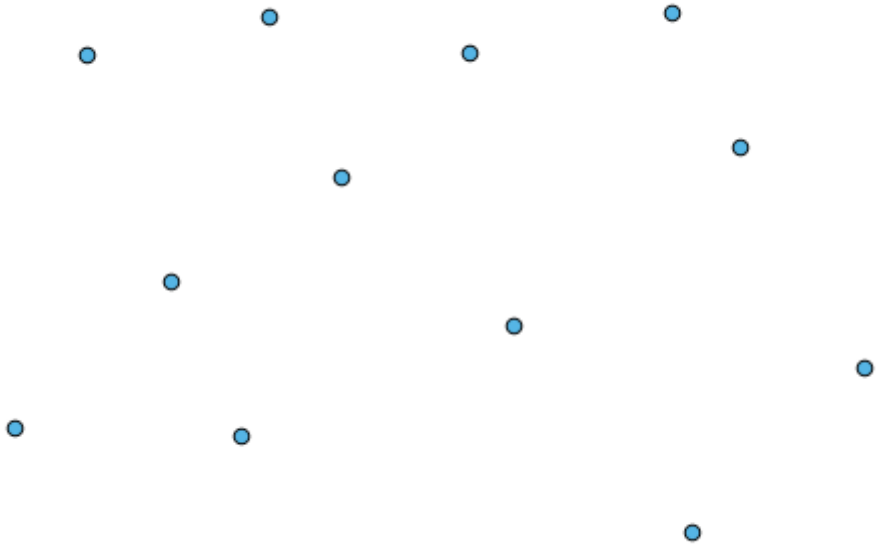
- *Table*: the table with the coordinates. You should select here the table from the lesson data.
- *X and Y fields*: these two parameters are linked to the first one. The corresponding selector will show the name of those fields that are available in the selected table. Select the *XCOORD* field for the *X* parameter, and the *YYCOORD* field for the *Y* parameter.
- *CRS*: Since this algorithm takes no input layers, it cannot assign a CRS to the output layer based on them. Instead, it asks you to manually select the CRS that the coordinates in the table use. Click on the button on the left–hand side to open the QGIS CRS selector, and select EPSG:4326 as the output CRS. We are using this CRS because the coordinates in the table are in that CRS.

Your dialog should look like this.

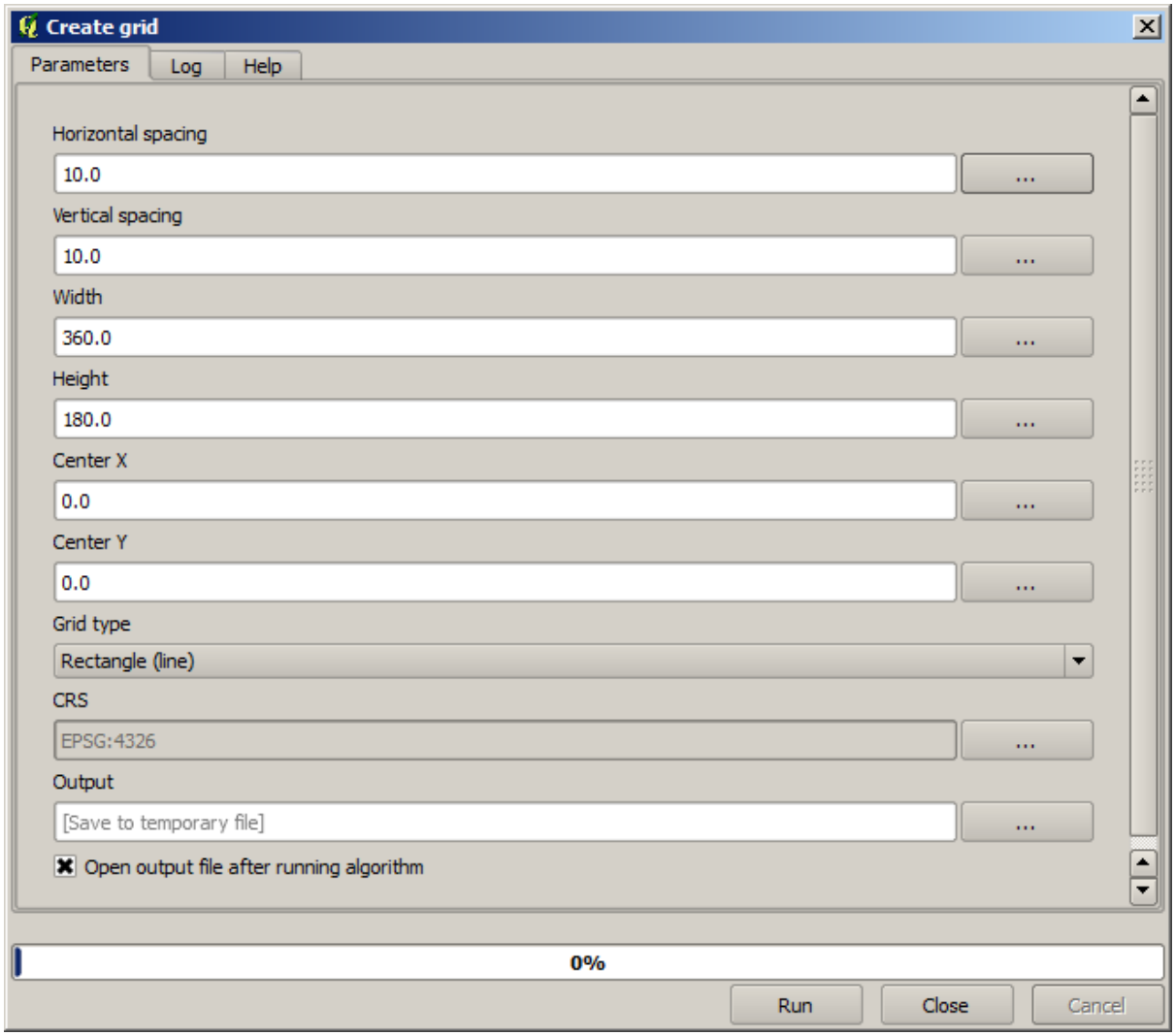


Now press the *Run* button to get the following layer (you may need to zoom full to reenter the map around the newly

created points):

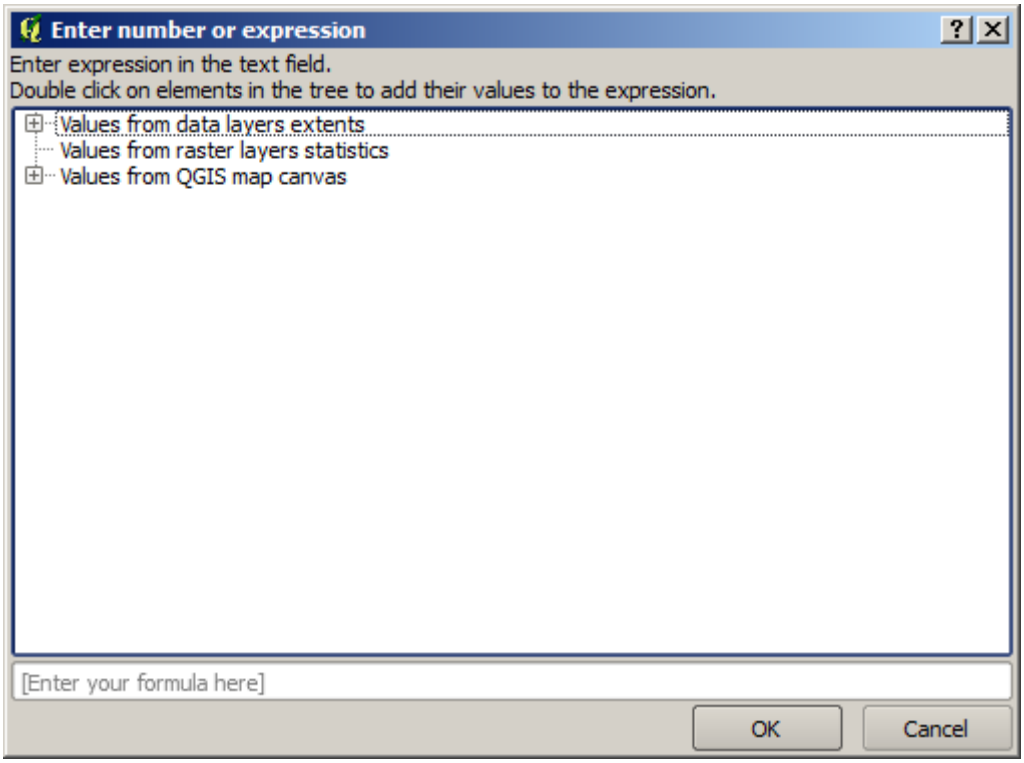


The next thing we need is the polygon layer. We are going to create a regular grid of polygons using the *Create grid* algorithm, which has the following parameters dialog.



Предупреждение: The options are simpler in recent versions of QGIS; you just need to enter min and max for X and Y (suggested values: -5.696226,-5.695122,40.24742,40.248171)

The inputs required to create the grid are all numbers. When you have to enter a numerical value, you have two options: typing it directly on the corresponding box or clicking the button on the right–hand side to get to a dialog like the one shown next.



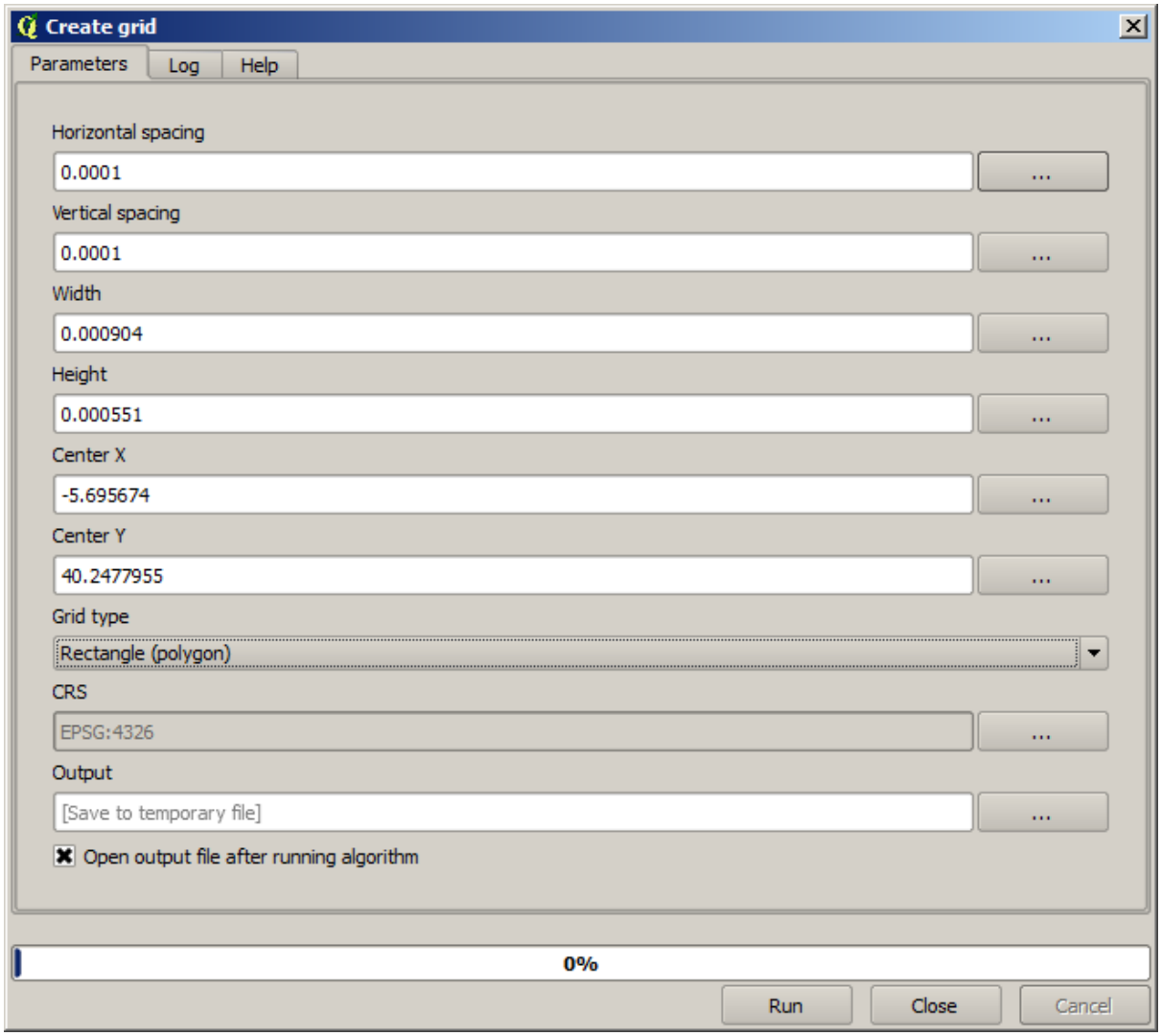
The dialog contains a simple calculator, so you can type expressions such as `11 * 34.7 + 4.6`, and the result will be computed and put in the corresponding text box in the parameters dialog. Also, it contains constants that you can use, and values from other layers available.

In this case, we want to create a grid that covers the extent of the input points layer, so we should use its coordinates to calculate the center coordinate of the grid and its width and height, since those are the parameters that the algorithm takes to create the grid. With a little bit of math, try to do that yourself using the calculator dialog and the constants from the input points layer.

Select *Rectangles (polygons)* in the *Type* field.

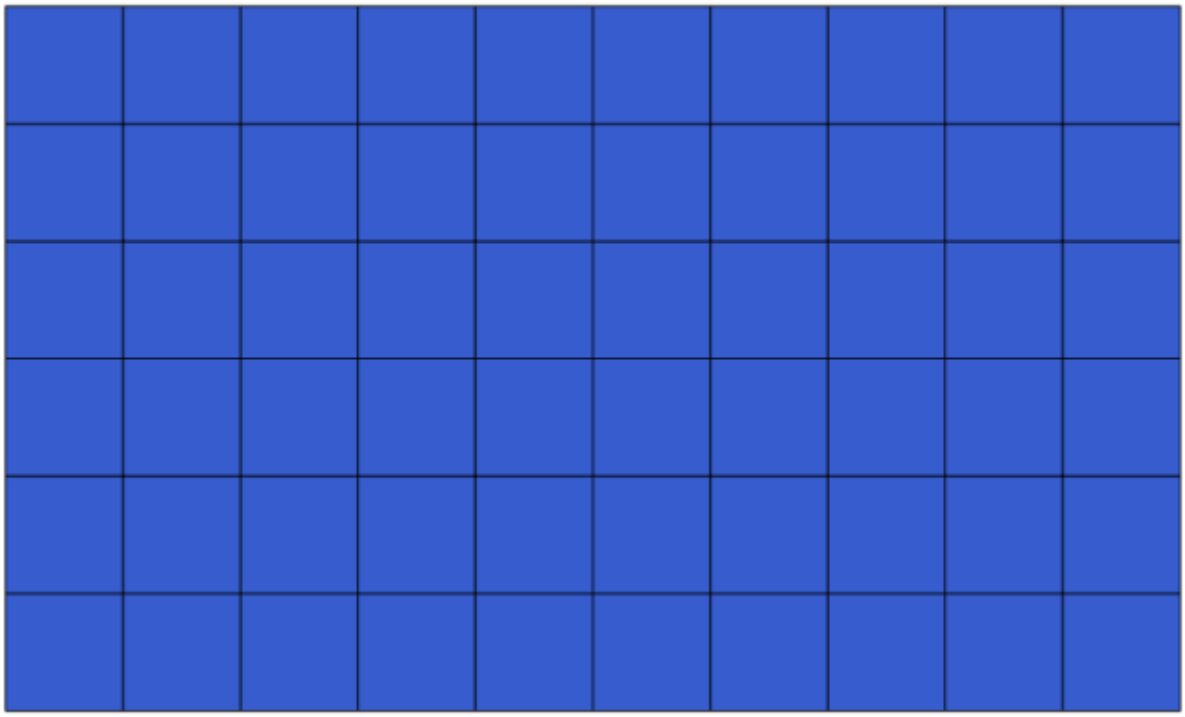
As in the case of the last algorithm, we have to enter the CRS here as well. Select EPSG:4326 as the target CRS, as we did before.

In the end, you should have a parameters dialog like this:

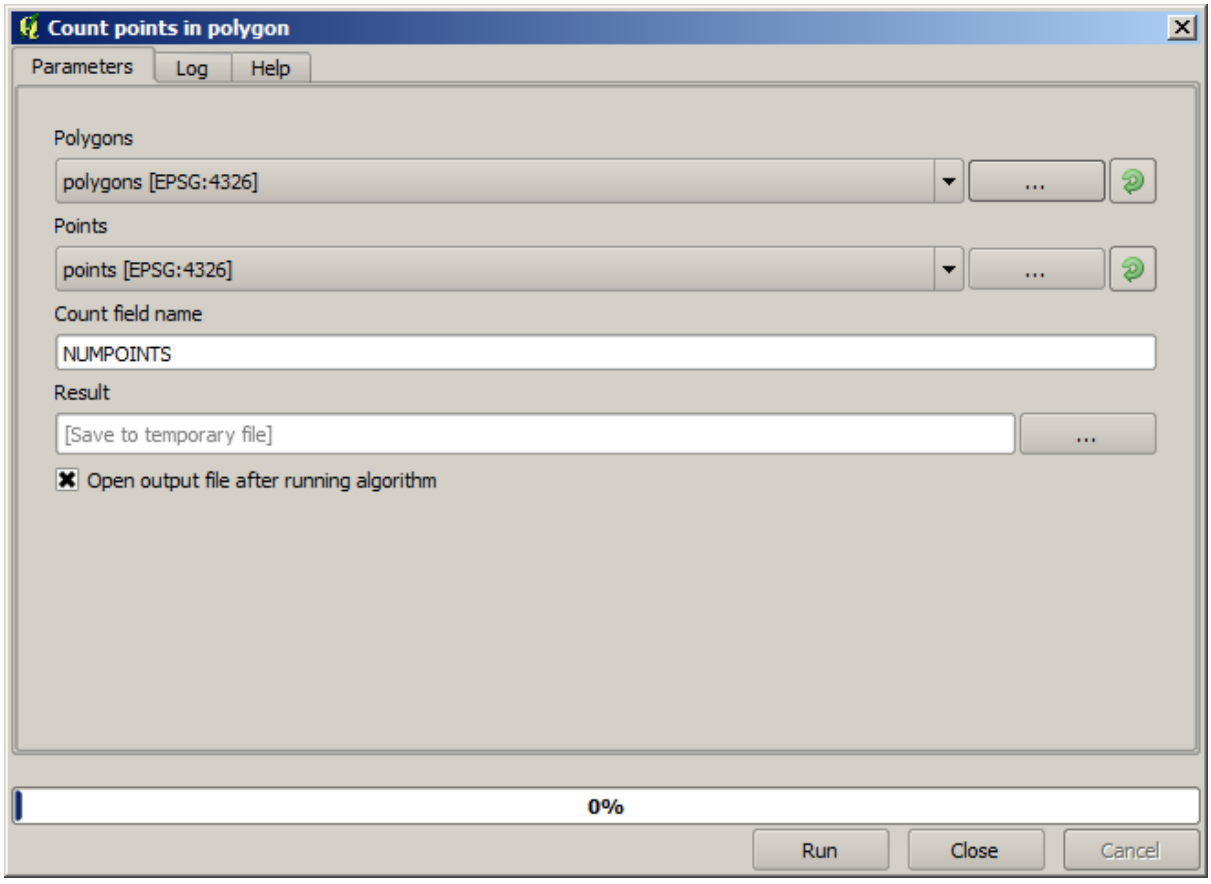


(Better add one spacing on the width and height: Horizontal spacing: 0.0001, Vertical spacing: 0.0001, Width: 0.001004, Height: 0.000651, Center X: -5.695674, Center Y: 40.2477955) The case of X center is a bit tricky, see: -5.696126+((-5.695222+ 5.696126)/2)

Press *Run* and you will get the graticule layer.

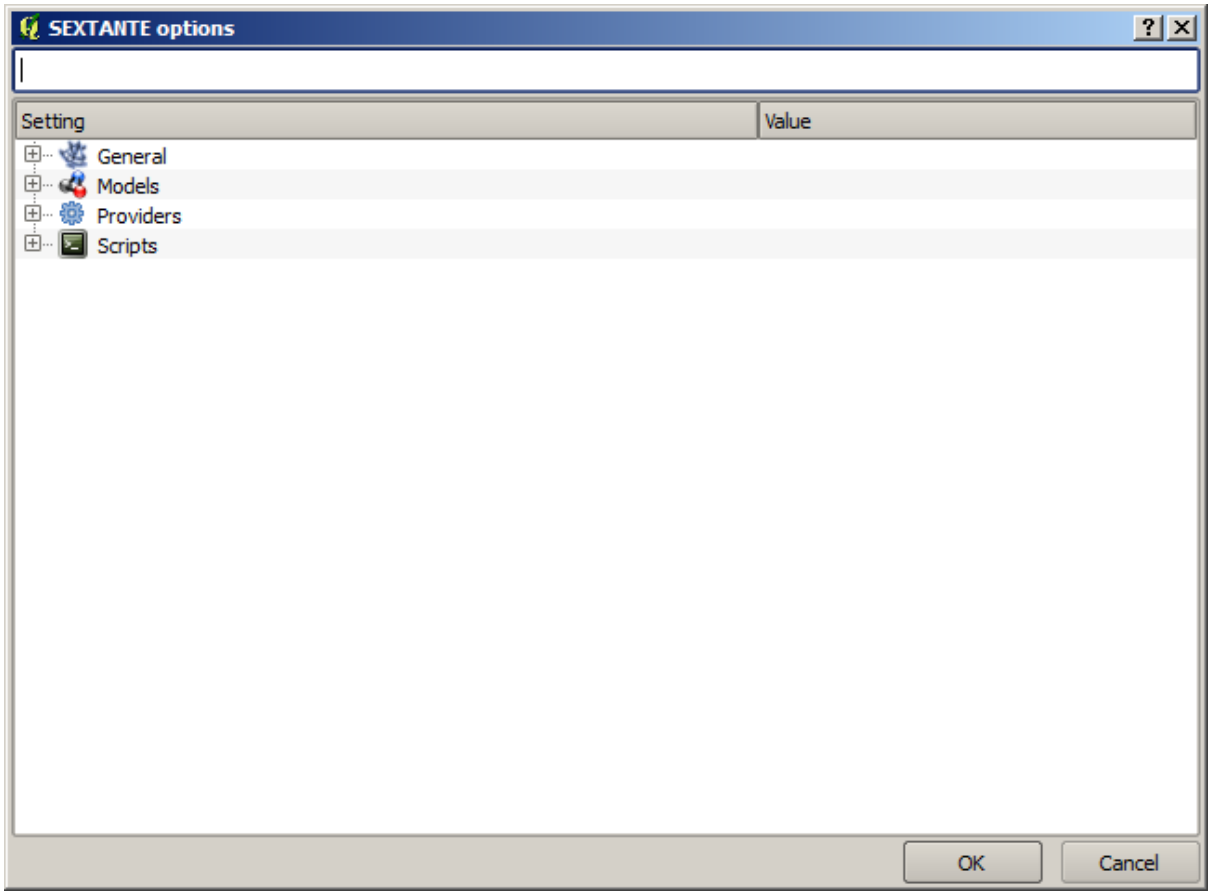


The last step is to count the points in each one of the rectangles of that graticule. We will use the *Count points in polygons* algorithm.

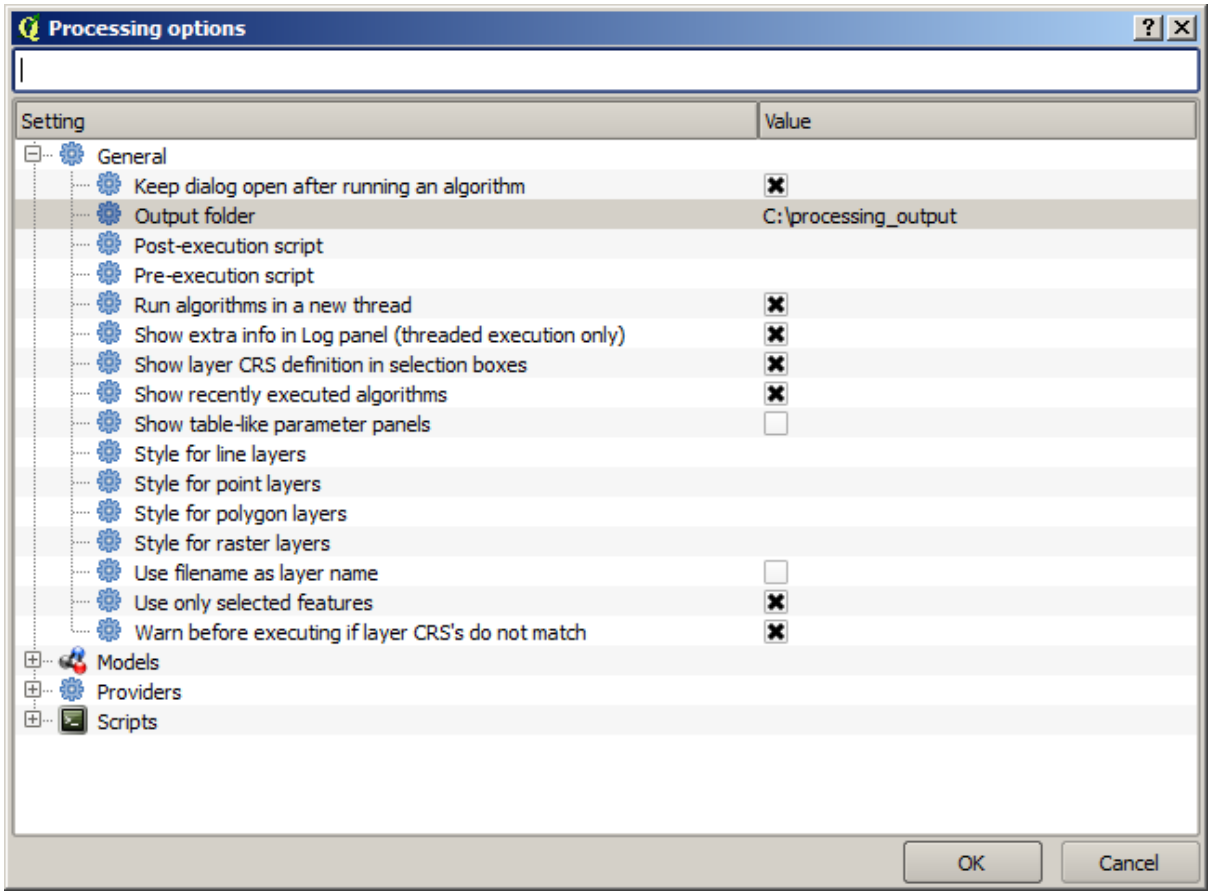


Now we have the result we were looking for.

Before finishing this lesson, here is a quick tip to make your life easier in case you want to persistently save your data. If you want all your output files to be saved in a given folder, you do not have to type the folder name each time. Instead, go to the processing menu and select the *Options and configuration* item. It will open the configuration dialog.



In the *Output folder* entry that you will find in the *General* group, type the path to your destination folder.



Now when you run an algorithm, just use the filename instead of the full path. For instance, with the configuration shown above, if you enter `graticule.shp` as the output path for the algorithm that we have just used, the result will be saved in `D:\processing_output\graticule.shp`. You can still enter a full path in case you want a result to be saved in a different folder.

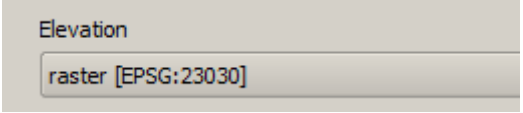
Try yourself the *Create grid* algorithm with different grid sizes, and also with different types of grids.

17.6 CRSs. Reprojecting

Примечание: In this lesson we will discuss how Processing uses CRSs. We will also see a very useful algorithm: reprojecting.

CRS's are a great source of confusion for QGIS Processing users, so here are some general rules about how they are handled by geoalgorithms when creating a new layer.

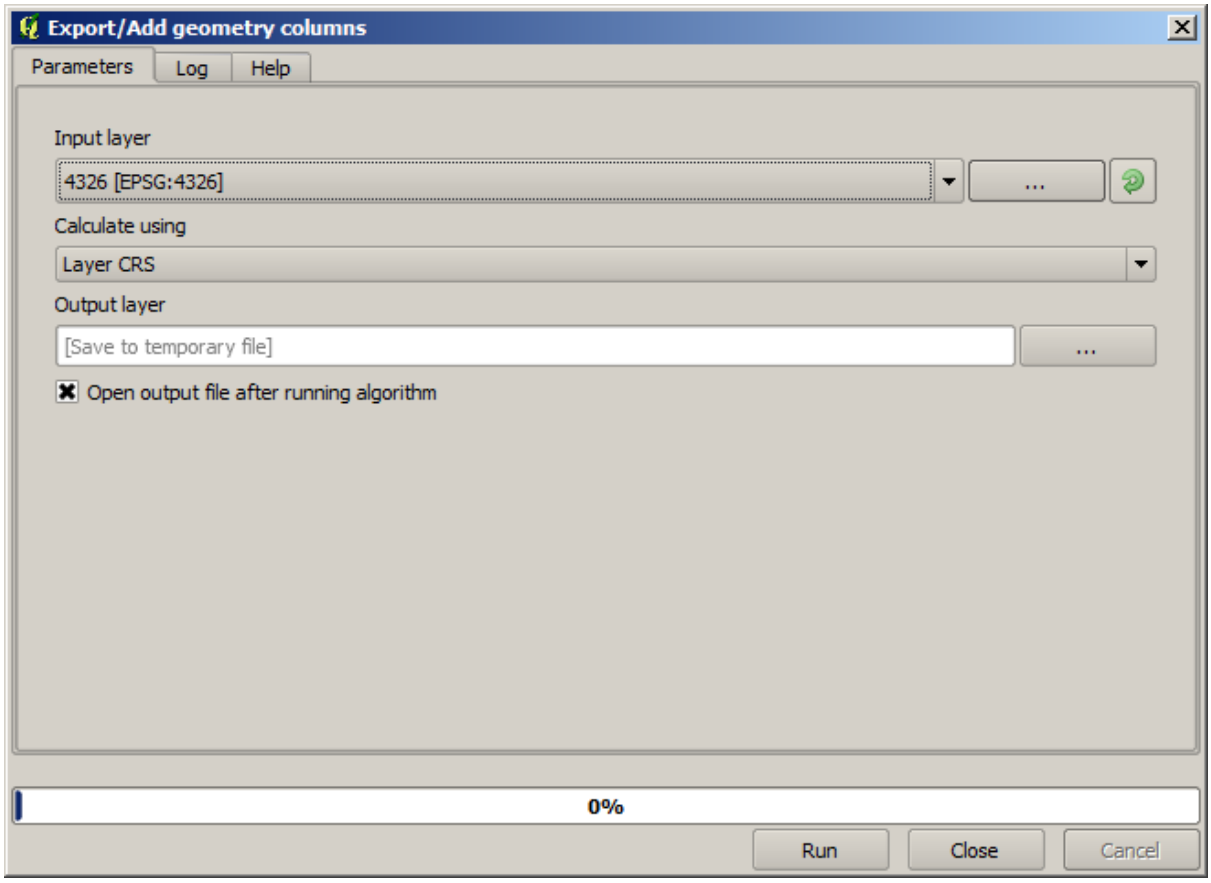
- If there are input layers, it will use the CRS of the first layer. This is assumed to be the CRS of all input layers, since they should have the same one. If you use layers with unmatching CRS's, QGIS will warn you about it. Notice that the CRS of input layers is shown along with its name in the parameters dialog.



- If there are no input layer, it will use the project CRS, unless the algorithm contains a specific CRS field (as it happened in the last lesson with the graticule algorithm)

Open the project corresponding to this lesson and you will see two layers named 23030 and 4326. They both contain the same points, but in different CRSs (EPSG:23030 and EPSG:4326). They appear in the same place because QGIS is reprojecting on the fly to the project CRS (EPSG:4326), but they are not actually the same layer.

Open the *Export/Add geometry columns* algorithm.



This algorithm add new columns to the attributes table of a vector layer. The content of the columns depend on the type of geometry of the layer. In the case of points, it adds new columns with the X and Y coordinates of each point.

In the list of available layers that you will find in the input layer field, you will see each one with its corresponding CRS. That means that, although they appear in the same place in your canvas, they will be treated differently. Select the 4326 layer.

The other parameter of the algorithm allows to set how the algorithm uses coordinates to calculate the new value that it will add to the resulting layers. Most algorithms do not have an option like that, and just use the coordinates directly. Select the *Layer CRS* option to just use coordinates as they are. This is how almost all geoalgorithms work.

You should get a new layer with exactly the same points as the other two layers. If you right click on the name of the layer and open its properties, you will see that it shares the same CRS of the input layer, that is, EPSG:4326. When the layer is loaded into QGIS, you will not be asked to enter the CRS of the layer, since QGIS already knows about it.

If you open the attributes table of the new layer you will see that it contains two new fields with the X and Y coordinates of each point.

	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

Those coordinate values are given in the layer CRS, since we chose that option. However, even if you choose another option, the output CRS of the layer would have been the same, since the input CRS is used to set the CRS of the output layer. Choosing another option will cause the values to be different, but not the resulting point to change or the CRS of the output layer to be different to the CRS of the input one.

Now do the same calculation using the other layer. You should find the resulting layer rendered exactly in the same place as the other ones, and it will have the EPSG:23030 CRS, since that was the one of the input layer.

If you go to its attribute table, you will see values that are different to the ones in the first layer that we created.

	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

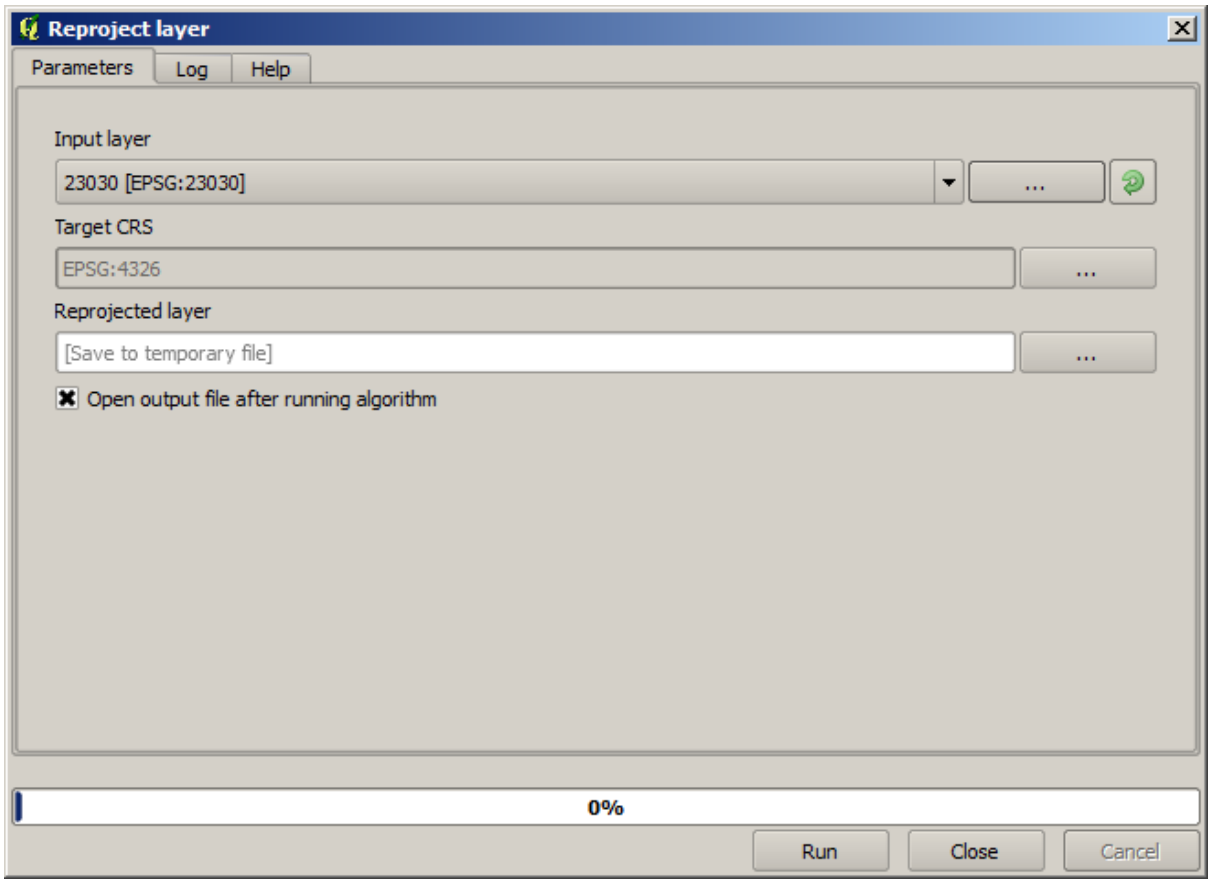
This is because the original data is different (it uses a different CRS), and those coordinates are taken from it.

What should you learn from this? The main idea behind these examples is that geoalgorithms use the layer as it is in its original data source, and completely ignore the reprojections that QGIS might be doing before rendering. In other words, do not trust what you see in the canvas, but always have in mind that the original data will be used. That is not so important in this case, since we are just using one single layer at a time, but in an algorithm that needs several of them (such as a clip algorithm), layers that appear to match or overlay might be very far one from each other, since they might have different CRSs.

Algorithms performs no reprojection (except in the reprojection algorithm that we will see next), so it is up to you to make sure that layers have matching CRS's.

An interesting module that deals with CRS's is the reprojection one. It represents a particular case, since it has an input layer (the one to reproject), but it will not use its CRS for the output one.

Open the *Reproject layer* algorithm.



Select any of the layers as input, and select EPSG:23029 as the destination CRS. Run the algorithm and you will get a new layer, identical to the input one, but with a different CRS. It will appear on the same region of the canvas, like the other ones, since QGIS will reproject it on the fly, but its original coordinates are different. You can see that by running the *Export/Add geometry columns* algorithm using this new layer as input, and veryfing that the added coordinates are different to the ones in the attribute tables of both of the two layers that we had computed before.

17.7 Selection

Примечание: In this lesson we will see how processing algorithms handle selections in vector layers that are used as inputs, and how to create a selection using a particular type of algorithm.

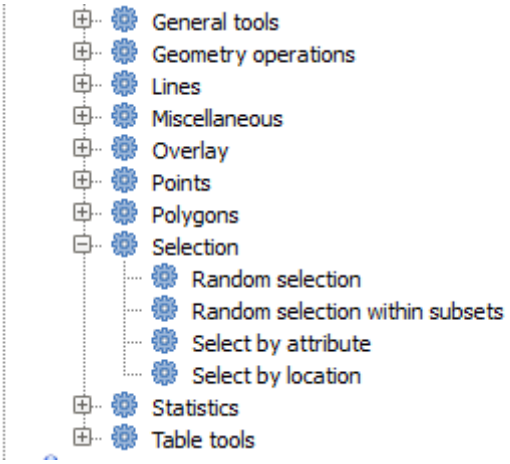
Unlike other analysis plugins in QGIS, you will not find in processing geoalgorithms any «Use only selected features» checkbox or similar. The behaviour regarding selection is set for the whole plugin and all its algorithms, and not for each algorithm execution. Algorithms follow the following simple rules when using a vector layer.

- If the layer has a selection, only selected features are used.
- If there is no selection, all features are used.

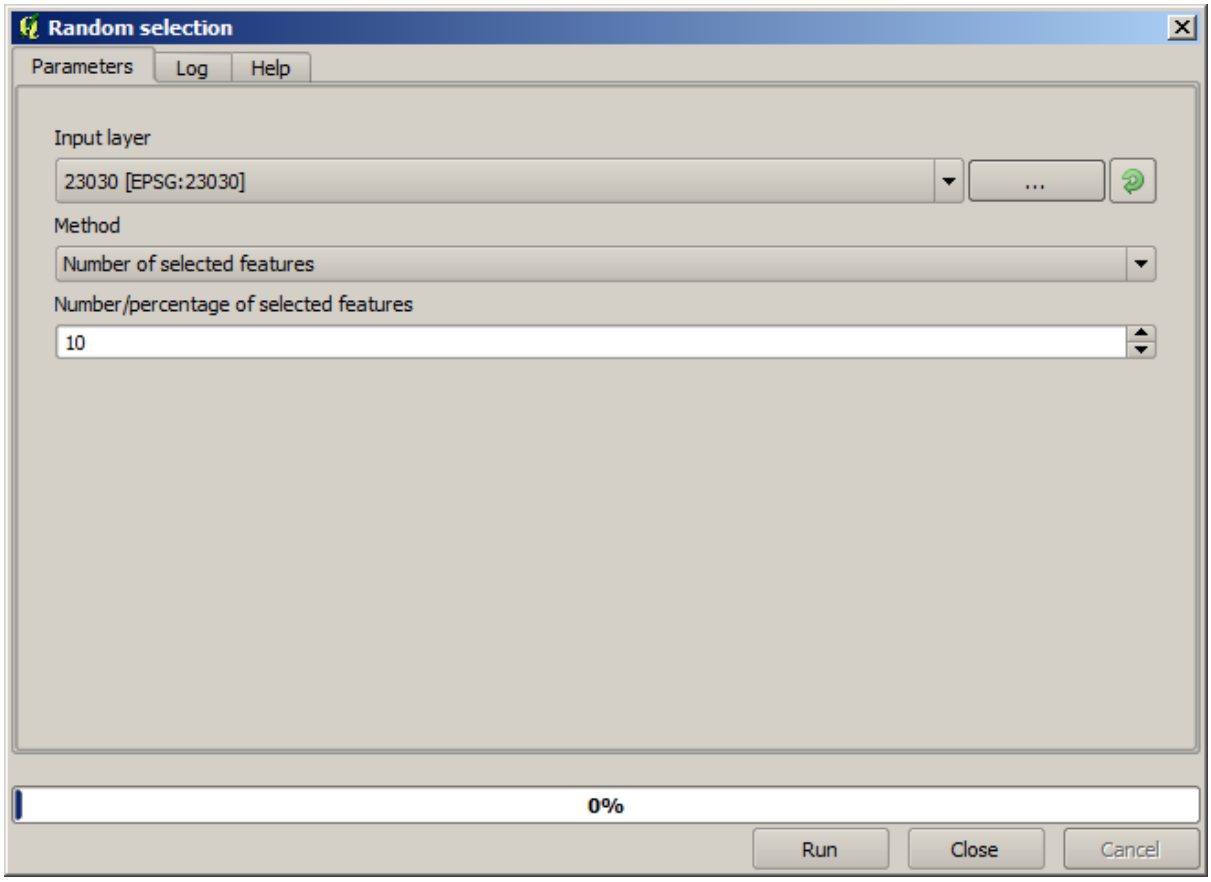
Please note that you can change this behaviour by unselecting the relevant option in the *Processing* ► *Options* ► *General* menu.

You can test that yourself by selecting a few points in any of the layers that we used in the last chapter, and running the reprojection algorithm on them. The reprojected layer that you will obtain will contain only those points that were selected, unless there was no selection, which will cause the resulting layer to contain all points from the original layer.

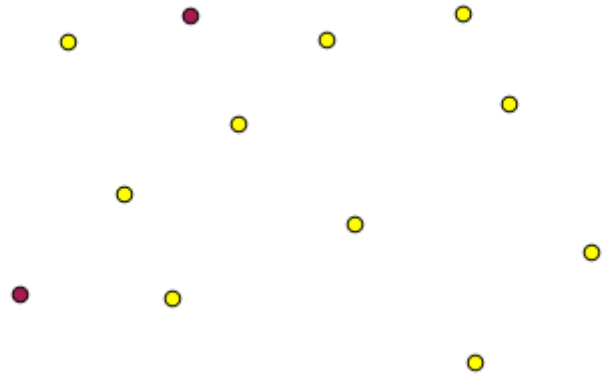
To make a selection, you can use any of the available methods and tools in QGIS. However, you can also use a geoalgorithm to do so. Algorithms for creating a selection are found in the toolbox under *Vector/Selection*



Open the *Random selection* algorithm.



Leaving the default values, it will select 10 points from the current layer.



You will notice that this algorithm does not produce any output, but modifies the input layer (not the layer itself, but its selection). This is an uncommon behaviour, since all the other algorithms will produce new layers and not alter the input layers.

Since the selection is not part of the data itself, but something that only exist within QGIS, these selection algorithms only must be used selecting a layer that is open in QGIS, and not with the file selection option that you can find in the corresponding parameter value box.

The selection we have just made, like most of the ones created by the rest of the selection algorithms, can also be done manually from QGIS, so you might be wondering what is the point on using an algorithm for that. Although now this might not make much sense to you, we will later see how to create models and scripts. If you want to make a selection in the middle of a model (which defines a processing workflow), only a geoalgorithm can be added to a model, and other QGIS elements and operations cannot be added. That is the reason why some processing algorithms duplicate functionality that is also available in other QGIS elements.

By now, just remember that selections can be made using processing geoalgorithms, and that algorithms will only use the selected features if a selection exists, or all features otherwise.

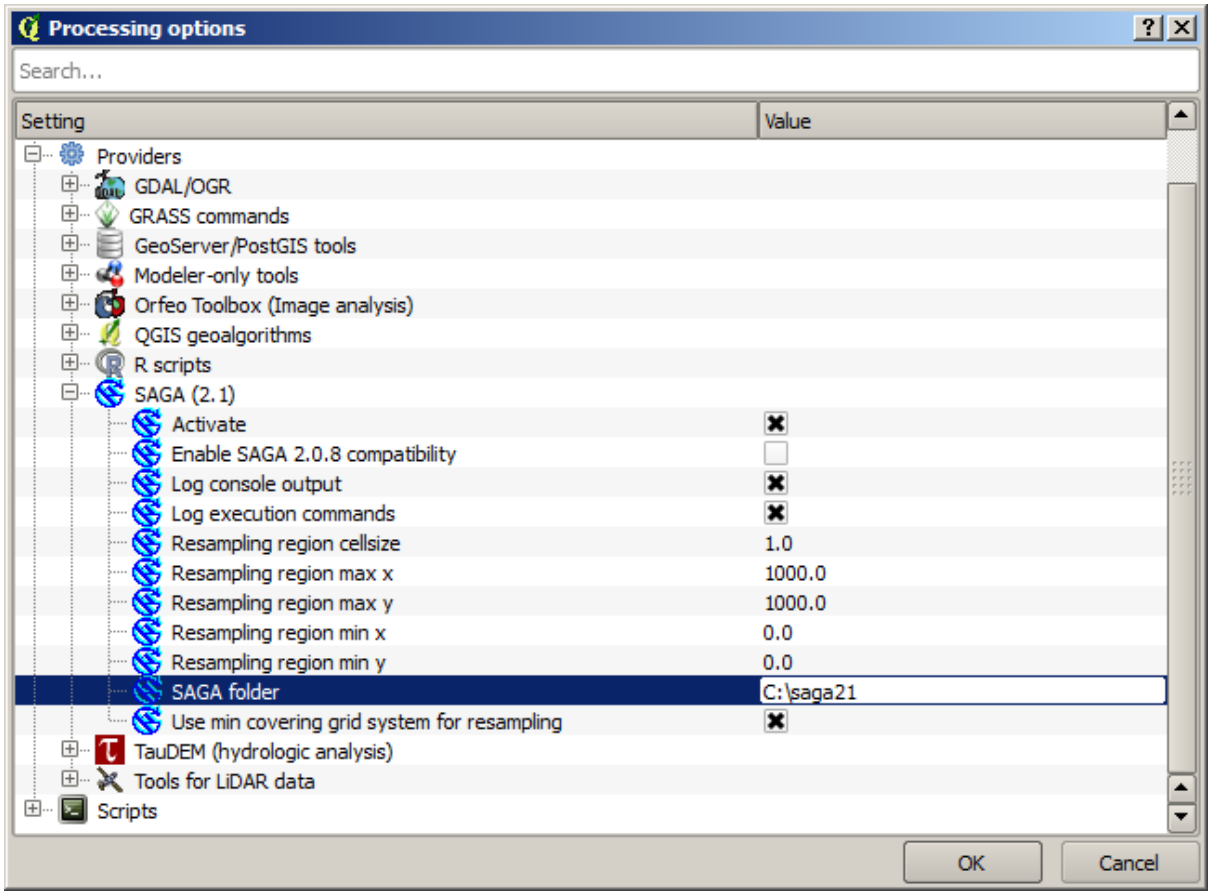
17.8 Running an external algorithm

Примечание: In this lesson we will see how to use algorithms that depend on a third-party application, particularly SAGA, which is one of the main algorithm providers.

All the algorithms that we have run so far are part of processing framework. That is, they are *native* algorithms implemented in the plugin and run by QGIS just like the plugin itself is run. However, one of the greatest features of the processing framework is that it can use algorithms from external applications and extend the possibilities of those applications. Such algorithms are wrapped and included in the toolbox, so you can easily use them from QGIS, and use QGIS data to run them.

Some of the algorithms that you see in the simplified view require third party applications to be installed in your system. One algorithm provider of special interest is SAGA (System for Automated Geospatial Analysis). First, we need to configure everything so QGIS can correctly call SAGA. This is not difficult, but it's important to understand how it works. Each external application has its own configuration, and later in this same manual we will talk about some of the other ones, but SAGA is going to be our main backend, so we will discuss it here.

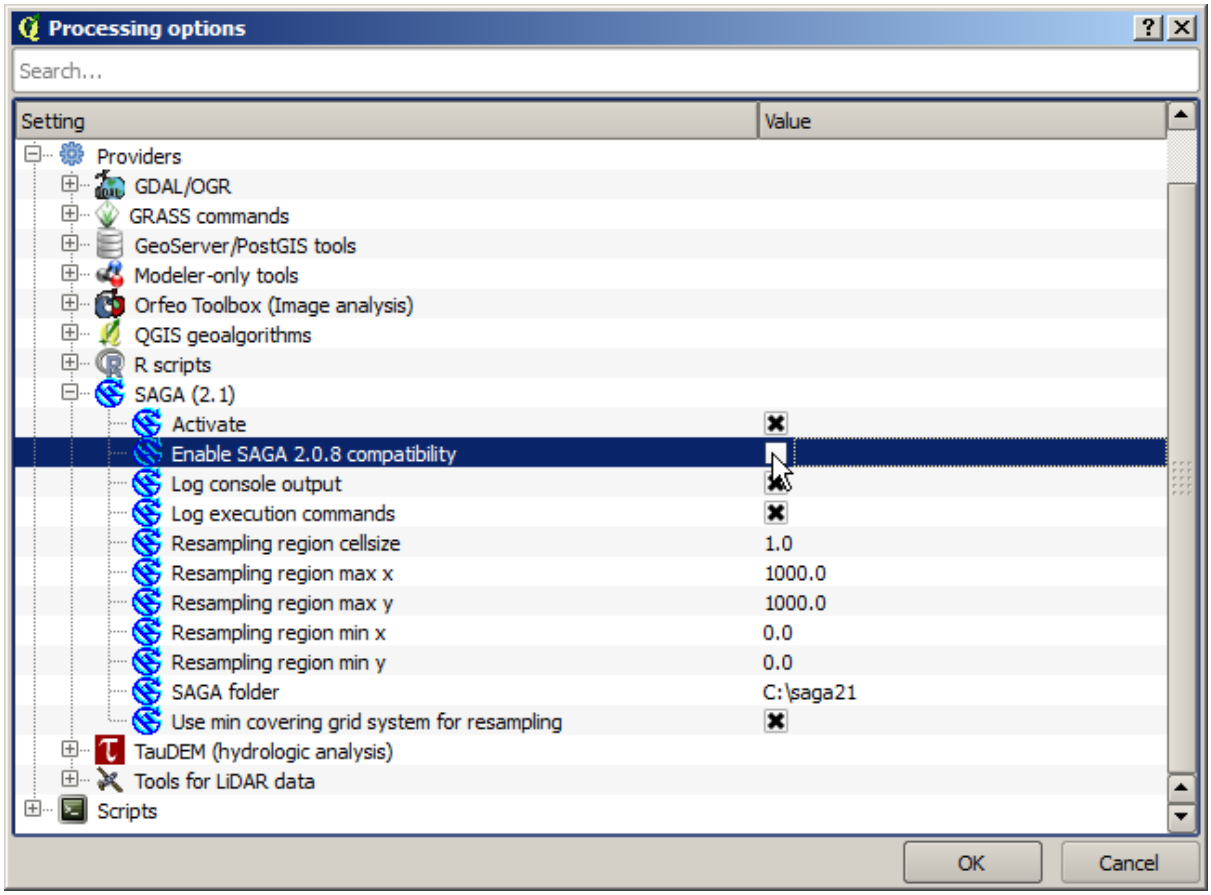
If you are on Windows, the best way to work with external algorithms is to install QGIS using the standalone installer. It will take care of installing all the needed dependencies, including SAGA, so if you have used it, there is nothing else to do. You can open the settings dialog and go to the *Providers/SAGA* group.



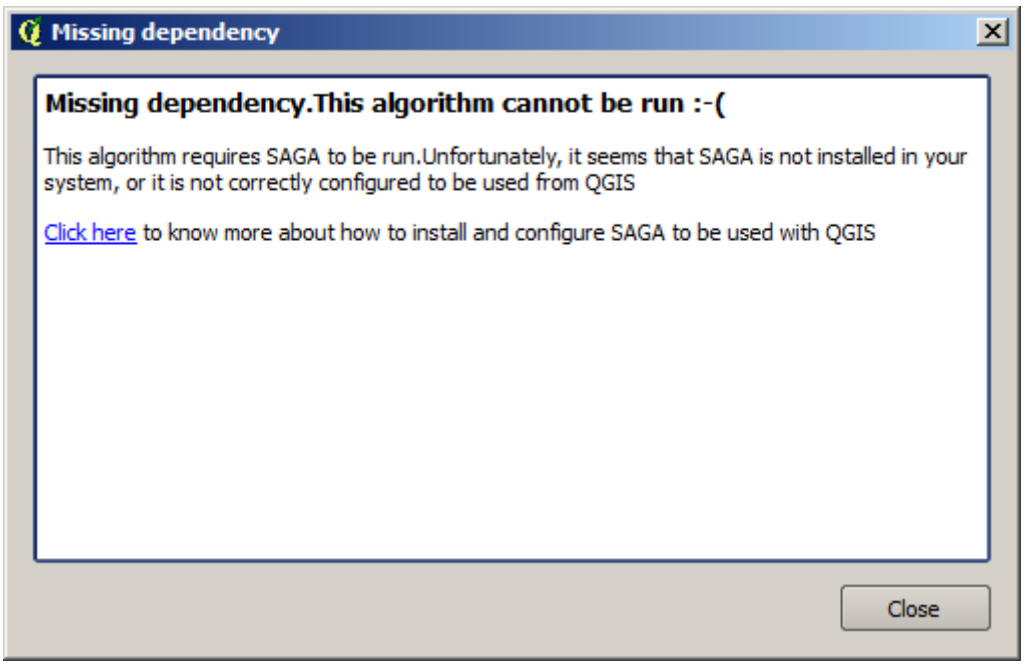
The SAGA path should already be configured and pointing to the folder where SAGA is installed.

If you have installed QGIS not using the standalone installer, then you must enter the path to your SAGA installation (which you must have installed separately) there. The required version is SAGA 2.1 [this is changing according to the releases of SAGA].

In case you are using Linux, you do not have to set the path to your SAGA installation in the processing configuration. Instead, you must install SAGA and make sure that the SAGA folder is in PATH, so it can be called from the console (just open a console and type `saga_cmd` to check it). Under Linux, the target version for SAGA is also 2.1, but in some installations (such as the OSGeo Live DVD) you might have just 2.0.8 available. There are some 2.1 packages available, but they are not commonly installed and might have some issues, so if you prefer to use the more common and stable 2.0.8, you can do it by enabling 2.0.8 compatibility in the configuration dialog, under the *SAGA* group.

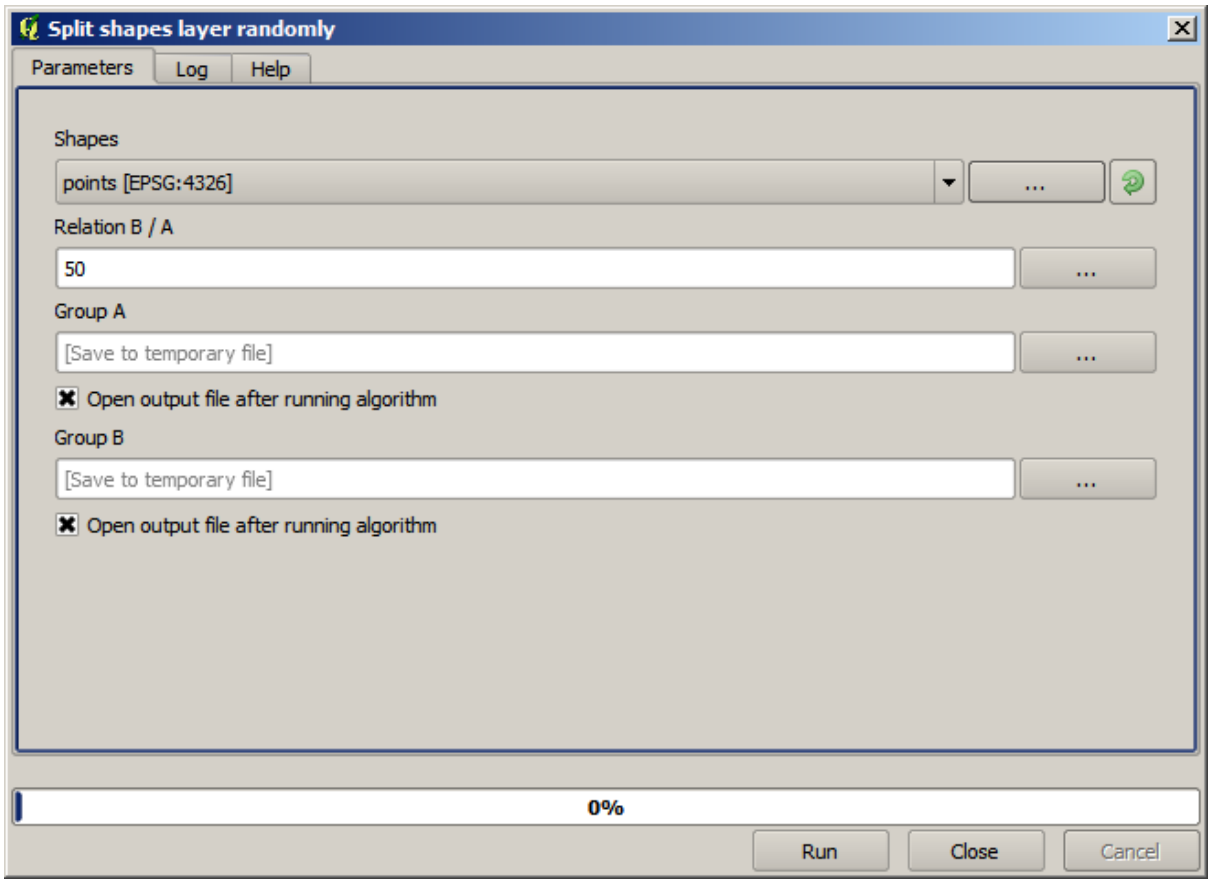


Once SAGA is installed, you can launch a SAGA algorithm double clicking on its name, as with any other algorithm. Since we are using the simplified interface, you do not know which algorithms are based on SAGA or in another external application, but if you happen to double–click on one of them and the corresponding application is not installed, you will see something like this.

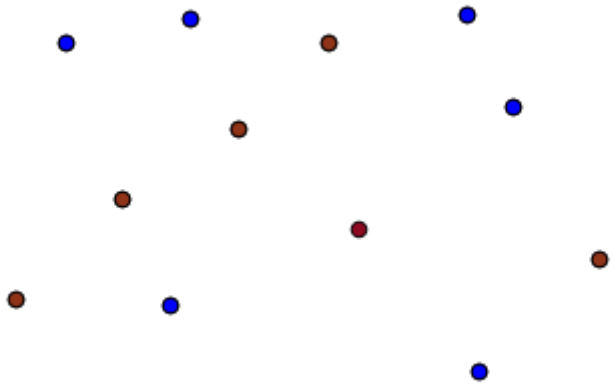


In our case, and assuming that SAGA is correctly installed and configured, you should not see this window, and you will get to the parameters dialog instead.

Let's try with a SAGA–based algorithm, the one called *Split shapes layer randomly*.



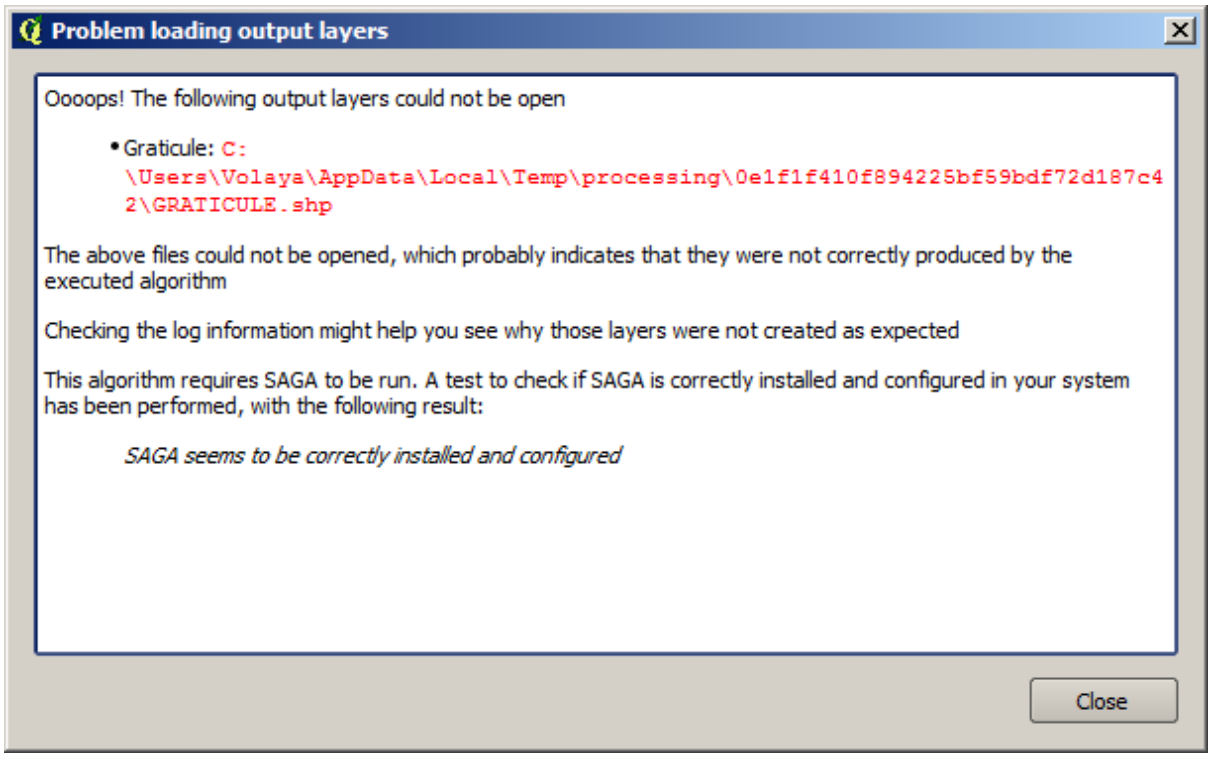
Use the points layer in the project corresponding to this lesson as input, and the default parameter values, and you will get something like this (the split is random, so your result might be different).



The input layer has been split in two layers, each one with the same number of points. This result has been computed by SAGA, and later taken by QGIS and added to the QGIS project.

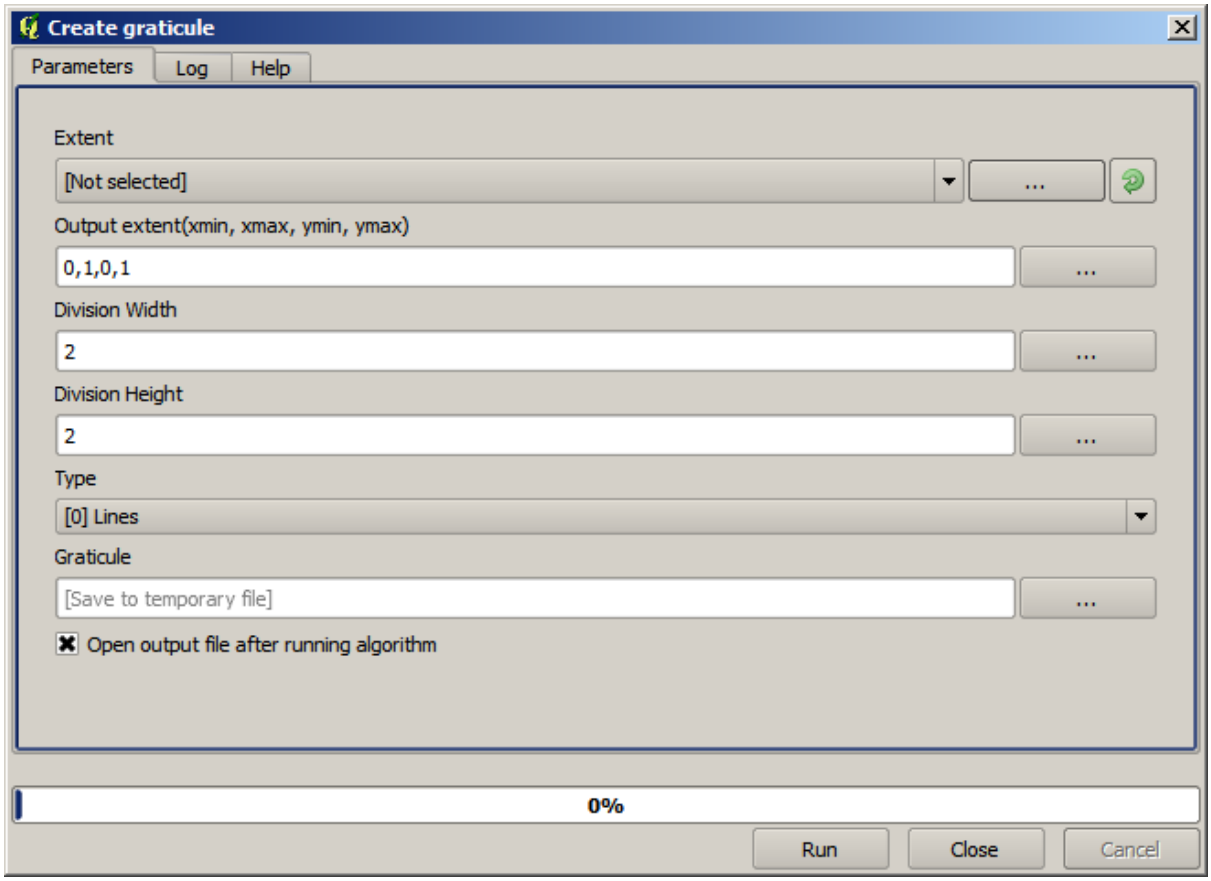
If all goes fine, you will not notice any difference between this SAGA–based algorithm and one of the others that we have previously run. However, SAGA might, for some reason, not be able to produce a result and not generate

the file that QGIS is expecting. In that case, there will be problems adding the result to the QGIS project, and an error message like this will be shown.



This kind of problems might happen, even if SAGA (or any other application that we are calling from the processing framework) is correctly installed, and it is important to know how to deal with them. Let's produce one of those error messages.

Open the *Create graticule* algorithm and use the following values.



We are using width and height values that is larger than the specified extent, so SAGA cannot produce any output. In other words, the parameter values are wrong, but they are not checked until SAGA gets them and tries to create the graticule. Since it cannot create it, it will not produce the expected layer, and you will see the error message shown above.

Примечание: In SAGA >= 2.2.3, the command will adjust automatically wrong input data, so you'll not get an error. To provoke an error, use negative values for division.

Understanding this kind of problems will help you solve them and find an explanation to what is happening. As you can see in the error message, a test is performed to check that the connection with SAGA is working correctly, indicating you that there might be a problem in how the algorithm was executed. This applies not only to SAGA, but also to other external applications as well.

In the next lesson we will introduce the processing log, where information about commands run by geoalgorithms is kept, and you will see how to get more detail when issues like this appear.

17.9 The processing log

Примечание: This lesson describes the processing log.

All the analysis performed with the processing framework is logged in QGIS logging system. This allows you to know more about what has been done with the processing tools, to solve problems when they happen, and also to re–run previous operations, since the logging system also implements some interactivity.

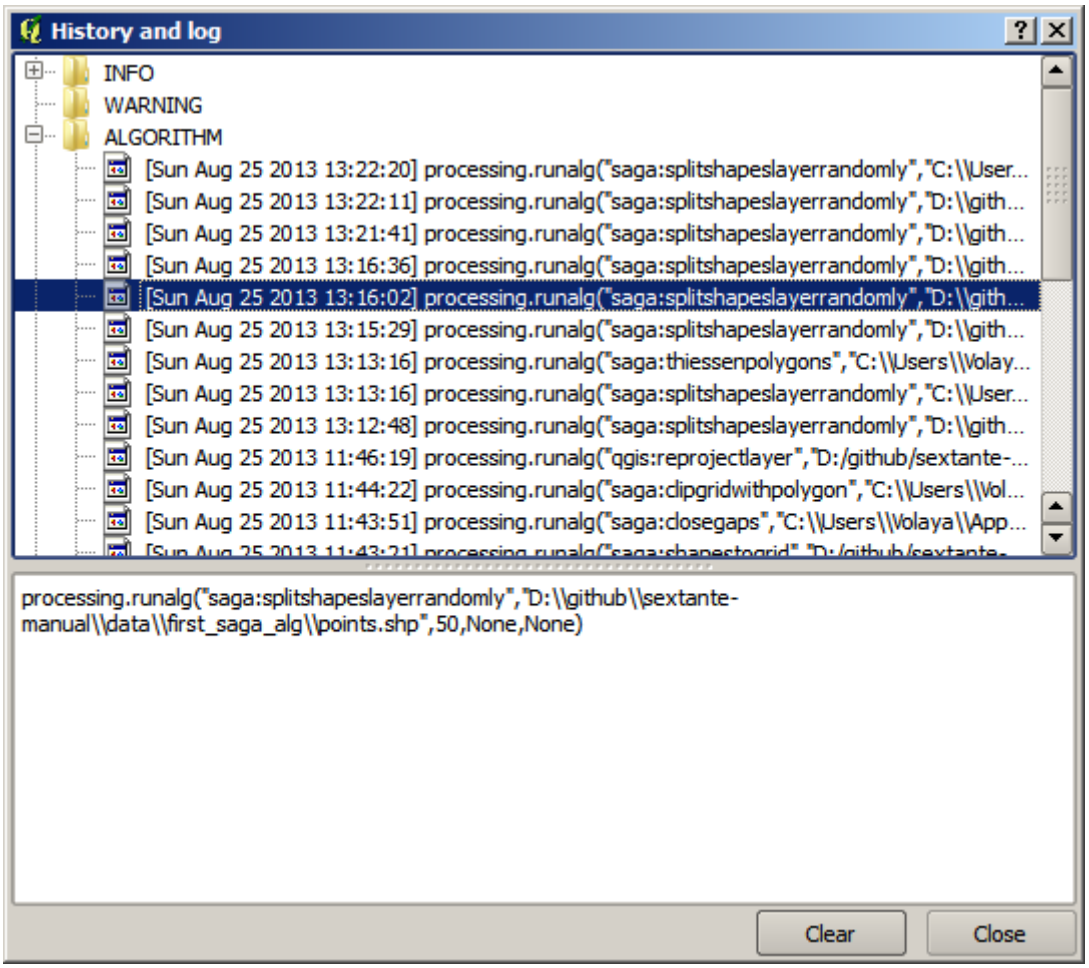
To open the log, click on the balloon at the bottom right, on the QGIS status bar. Some algorithms might leave here information about their execution. For instance, those algorithms that call an external application usually log the

console output of that application to this entry. If you have a look at it, you will see that the output of the SAGA algorithm that we just run (and that fail to execute because input data was not correct) is stored here.

This is helpful to understand what is going on. Advanced users will be able to analyze that output to find out why the algorithm failed. If you are not an advanced user, this will be useful for others to help you diagnose the problem you are having, which might be a problem in the installation of the external software or an issue with the data you provided.

Even if the algorithm could be executed, some algorithms might leave warnings in case the result might not be right. For instance, when executing an interpolation algorithm with a very small amount of points, the algorithm can run and will produce a result, but it is likely that it will not be correct, since more points should be used. It's a good idea to regularly check for this type of warnings if you are not sure about some aspect of a given algorithm.

From the *Processing* menu, under the *History* section, you'll find *Algorithms*. All algorithms that are executed, even if they are executed from the GUI and not from the console (which will be explained later in this manual) are stored in this section as a console call. That means that everytime you run an algorithm, a console command is added to the log, and you have the full history of your working session. Here is how that history looks like:



This can be very useful when starting working with the console, to learn about the syntax of algorithms. We will use it when we discuss how to run analysis commands from the console.

The history is also interactive, and you can re–run any previous algorithm just by double–clicking on its entry. This is an easy way of replicating the work we already did before.

For instance, try the following. Open the data corresponding to the first chapter of this manual and run the algorithm explained there. Now go to the log dialog and locate the last algorithm in the list, which corresponds to the algorithm you have just run. Double–click on it an a new result should be produced, just like when you run it using the normal dialog and calling it from the toolbox.

17.9.1  Advanced

You can also modify the algorithm. Just copy it, open the *Plugins* ► *Python console*, click on *Import class* ► *Import Processing class*, then paste it to re-run the analysis; change the text at will. To display the resulting file, type `iface.addVectorLayer('/path/filename.shp', 'Layer name in legend', 'ogr')`. Otherwise, you can use `processing.runandload`.

17.10 The raster calculator. No-data values

Примечание: In this lesson we will see how to use the raster calculator to perform some operations on raster layers. We will also explain what are no–data values and how the calculator and other algorithms deal with them

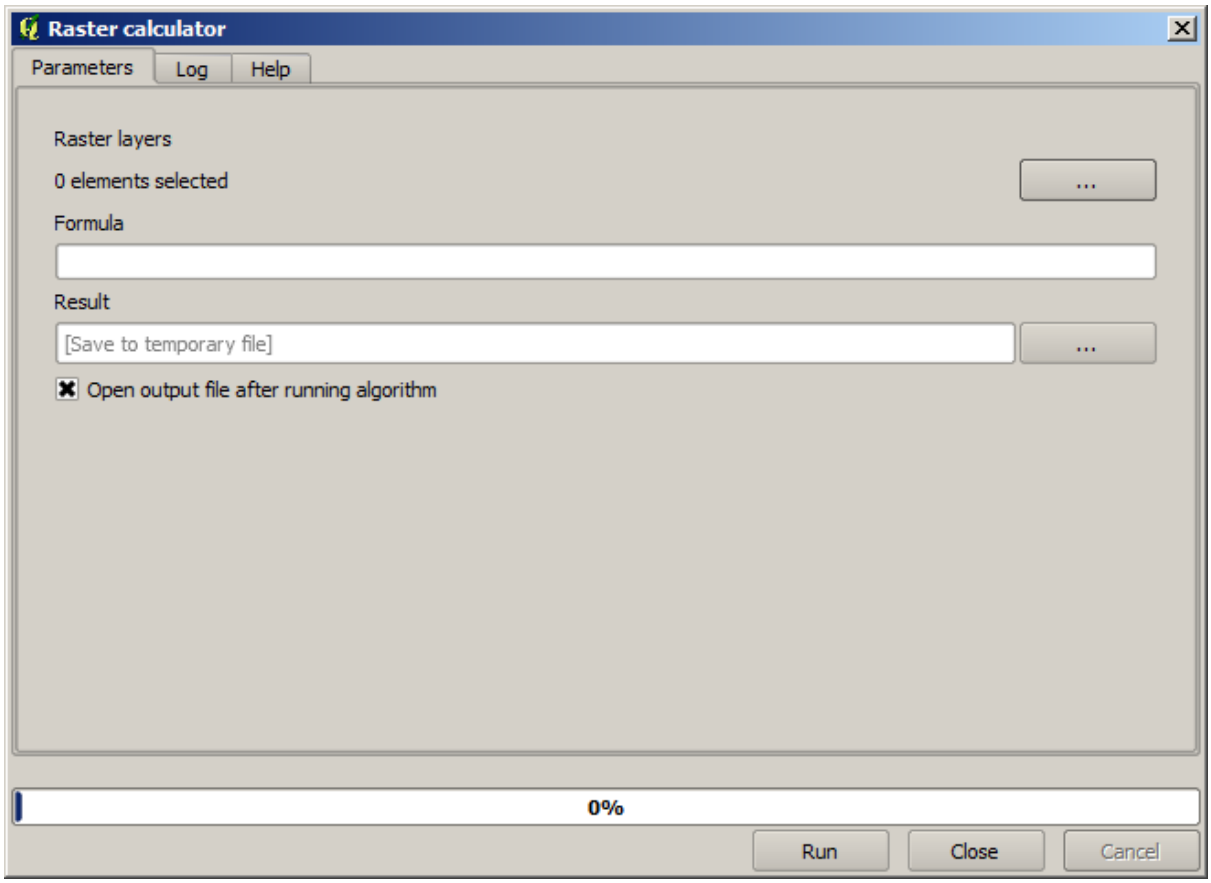
---

The raster calculator is one of the most powerful algorithms that you will find. It's a very flexible and versatile algorithm that can be used for many different calculations, and one that will soon become an important part of your toolbox.

In this lesson we will be performing some calculation with the raster calculator, most of them rather simple. This will let us see how it is used and how it deals with some particular situations that it might find. Understanding that is important to later get the expected results when using the calculator, and also to understand certain techniques that are commonly applied with it.

Open the QGIS project corresponding to this lesson and you will see that it contains several raster layers.

Now open the toolbox and open the dialog corresponding to the raster calculator.



Примечание: The interface is different in recent versions.

The dialog contains 2 parameters.

- The layers to use for the analysis. This is a multiple input, that meaning that you can select as many layers as you want. Click on the button on the right–hand side and then select the layers that you want to use in the dialog that will appear.
- The formula to apply. The formula uses the layers selected in the above parameter, which are named using alphabet letters (`a, b, c...`) or `g1, g2, g3...` as variable names. That is, the formula `a + 2 * b` is the same as `g1 + 2 * g2` and will compute the sum of the value in the first layer plus two times the value in the second layer. The ordering of the layers is the same ordering that you see in the selection dialog.

Предупреждение: The calculator is case sensitive.

To start with, we will change the units of the DEM from meters to feet. The formula we need is the following one:

```
h' = h * 3.28084
```

Select the DEM in the layers field and type `a * 3.28084` in the formula field.

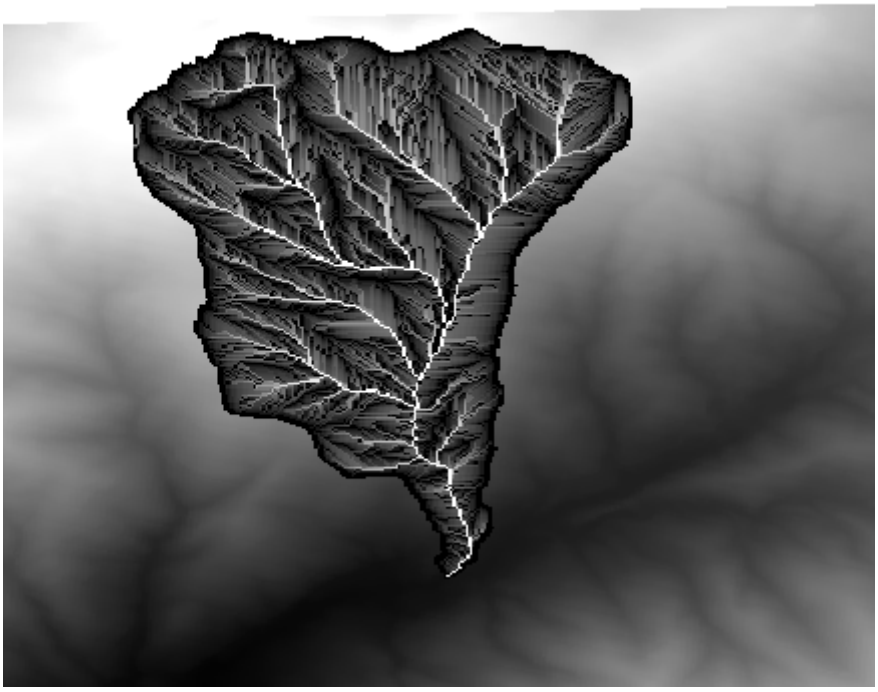
Предупреждение: For non English users: use always «.», not «,».

Click *Run* to run the algorithm. You will get a layer that has the same appearance of the input layer, but with different values. The input layer that we used has valid values in all its cells, so the last parameter has no effect at all.

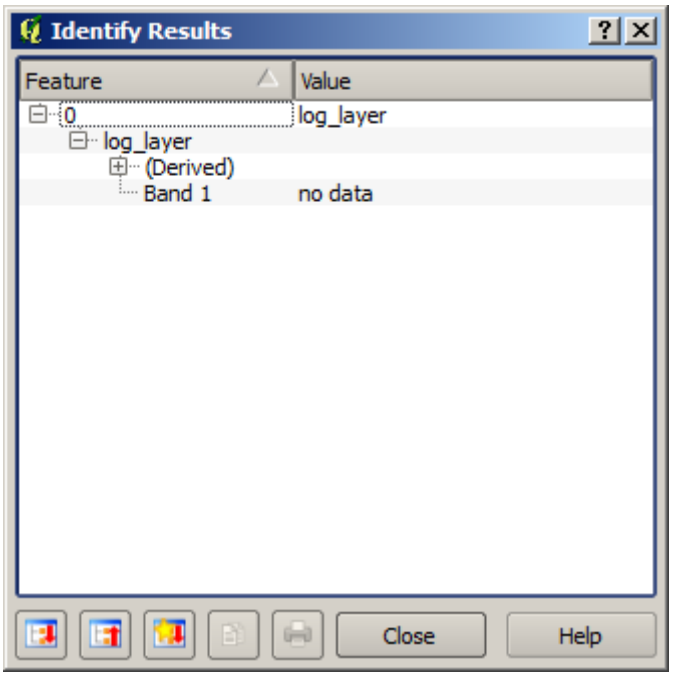
Let's now perform another calculation, this time on the *accflow* layer. This layer contains values of accumulated flow, a hydrological parameter. It contains those values only within the area of a given watershed, with no–data values outside of it. As you can see, the rendering is not very informative, due to the way values are distributed. Using the logarithm of that flow accumulation will yield a much more informative representation. We can calculate that using the raster calculator.

Open the algorithm dialog again, select the *accflow* layer as the only input layer, and enter the following formula: `log(a)`.

Here is the layer that you will get.



If you select the *Identify* tool to know the value of a layer at a given point, select the layer that we have just created, and click on a point outside of the basin, you will see that it contains a no–data value.



For the next exercise we are going to use two layers instead of one, and we are going to get a DEM with valid elevation values only within the basin defined in the second layer. Open the calculator dialog and select both layers of the project in the input layers field. Enter the following formula in the corresponding field:

```
a/a * b
```

`a` refers to the accumulated flow layer (since it is the first one to appear in the list) and `b` refers to the DEM. What

we are doing in the first part of the formula here is to divide the accumulated flow layer by itself, which will result in a value of 1 inside the basin, and a no–data value outside. Then we multiply by the DEM, to get the elevation value in those cells inside the basin (`DEM * 1 = DEM`) and the no–data value outside (`DEM * no_data = no_data`)

Here is the resulting layer.



This technique is used frequently to *mask* values in a raster layer, and is useful whenever you want to perform calculations for a region other that the arbitrary rectangular region that is used by raster layer. For instance, an elevation histogram of a raster layer doesn't have much meaning. If it is instead computed using only values corresponding to a basin (as in he case above), the result that we obtain is a meaningful one that actually gives information about the configuration of the basin.

There are other interesting things about this algorithm that we have just run, apart from the no–data values and how they are handled. If you have a look at the extents of the layers that we have multiplied (you can do it double–clicking on their names of the layer in the table of contents and looking at their properties), you will see that they are not the same, since the extent covered by the flow accumulation layer is smaller that the extent of the full DEM.

That means that those layers do not match, and that they cannot be multiplied directly without homogenizing those sizes and extents by resampling one or both layers. However, we did not do anything. QGIS takes care of this situation and automatically resamples input layers when needed. The output extent is the minimum covering extent calculated from the input layers, and the minimum cell size of their cellsizes.

In this case (and in most cases), this produces the desired results, but you should always be aware of the additional operations that are taking place, since they might affect the result. In cases when this behaviour might not be the desired, manual resampling should be applied in advance. In later chapters, we will see more about the behaviour of algorithms when using multiple raster layers.

Let's finish this lesson with another masking exercise. We are going to calculate the slope in all areas with an elevation between 1000 and 1500 meters.

In this case, we do not have a layer to use as a mask, but we can create it using the calculator.

Run the calculator using the DEM as only input layer and the following formula

```
ifelse(abs(a-1250) < 250, 1, 0/0)
```

As you can see, we can use the calculator not only to do simple algebraic operations, but also to run more complex calculation involving conditional sentences, like the one above.

The result has a value of 1 inside the range we want to work with, and no-data in cells outside of it.



The no-data value comes from the 0/0 expression. Since that is an undetermined value, SAGA will add a NaN (Not a Number) value, which is actually handled as a no-data value. With this little trick you can set a no-data value without needing to know what the no–data value of the cell is.

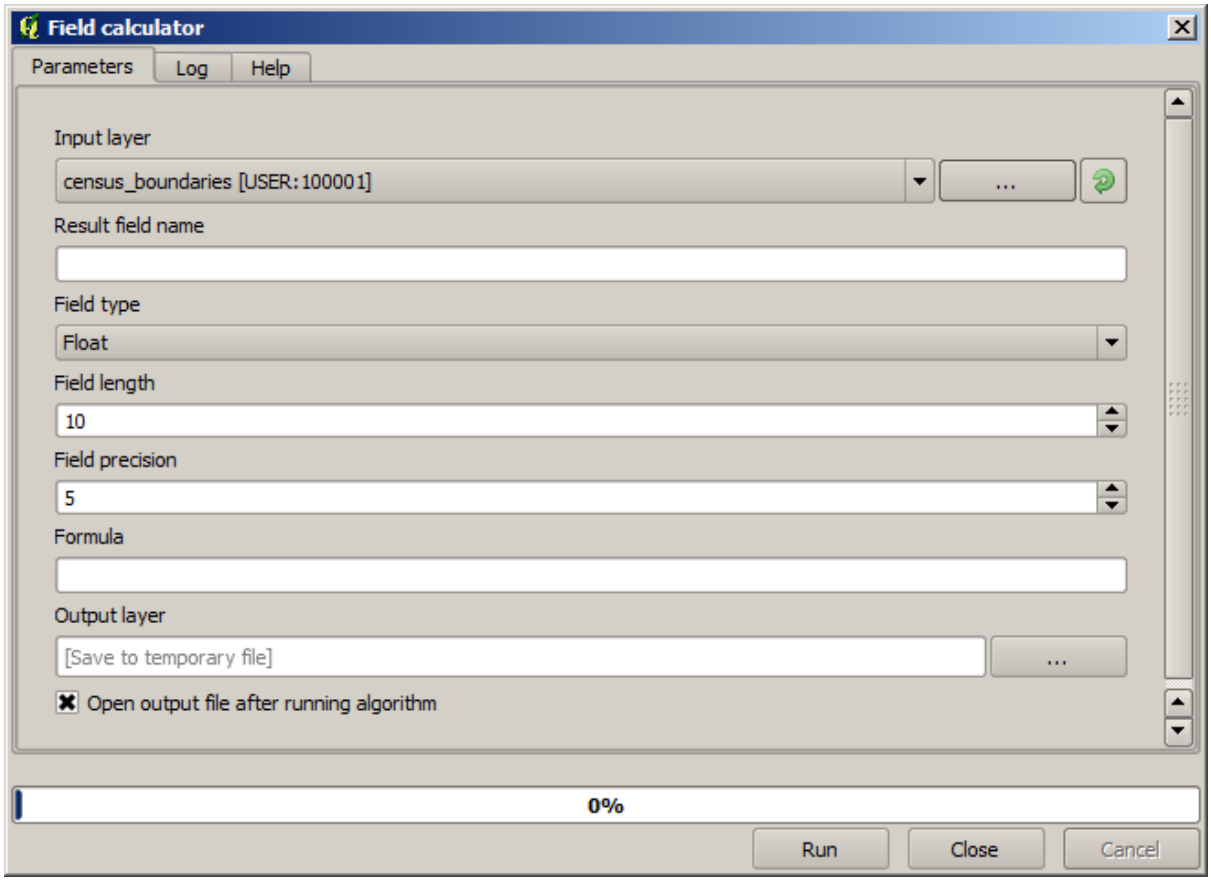
Now you just have to multiply it by the slope layer included in the project, and you will get the desired result.

All that can be done in a single operation with the calculator. We leave that as an exercise for the reader.

17.11 Калькулятор для векторов

Примечание: В этом уроке мы добавим новые атрибуты в векторный слой, используя математические выражения Векторного калькулятора.

Мы уже знакомы с процессом использования калькулятора растров для того, чтобы создать растровый слой на основе математических выражений. Подобный алгоритм существут для вектроных слов. Он создает новый слой с такими же атрибутами, как и у входного слоя, добавляя новое поле с результатами вычислений математического выражения. Этот алогормит называется «Калькулятор полей» со следующим набором параметров.



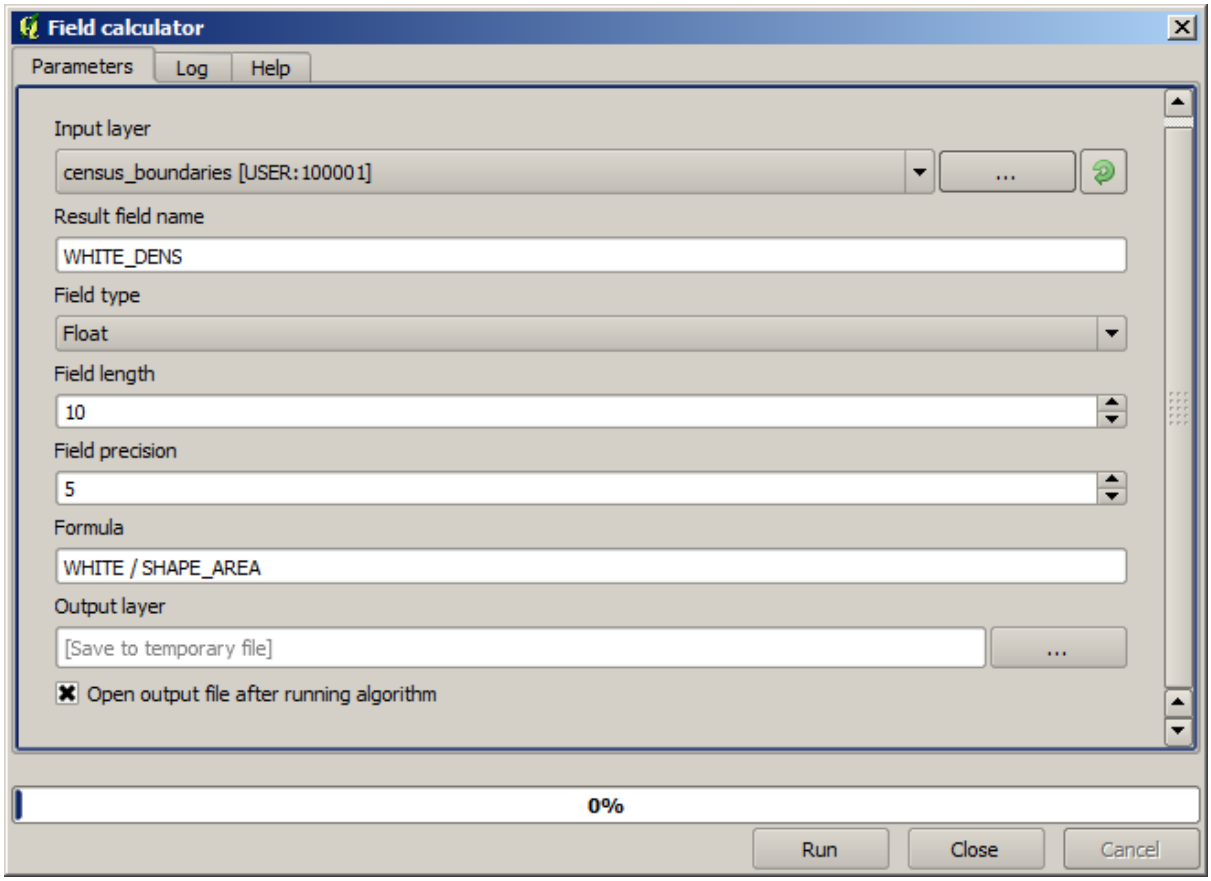
Примечание: В более новых версиях интерфейс панели иструментов Обработки значительно изменился - он стал более понятным и легким в использовании.

Здесь приведены примеры того, как используется алгоритм.

Сначала давайте посчитайм плотность белого населения в каждом полигоне, который представляет перепись населения. В таблице атрибутов есть два поля, которые будут использованы для вычислений *WHITE*` и `SHAPE_AREA`. Нам нужно лишь разделить одно на другое и умножить на один миллион (чтобы получить плотность населения на километр), для этого воспользуемся следующей формулой

```
( "WHITE" / "SHAPE_AREA" ) * 1000000
```

Диалоговое окно должно быть заполнено следующими параметрами.



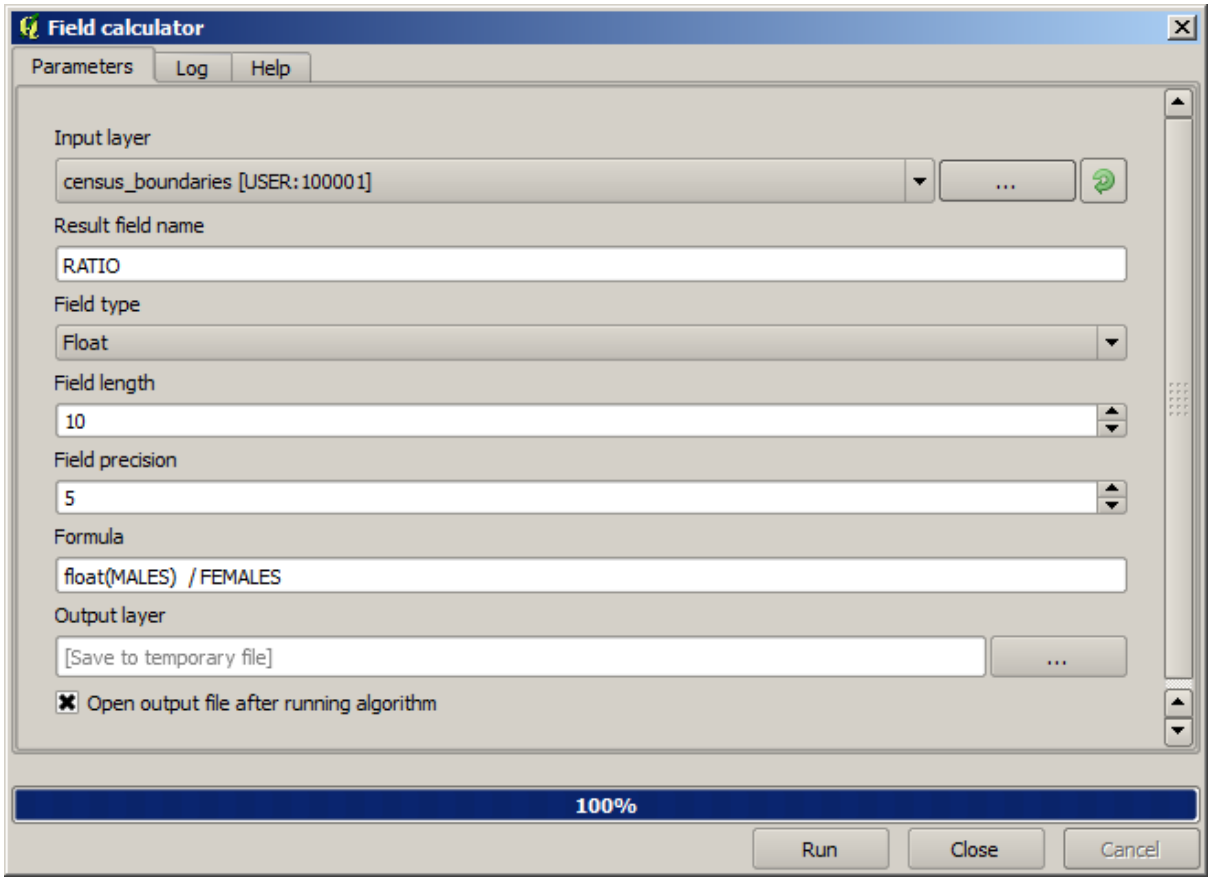
после чего будет создано новое поле `WHITE_DENS`

Далее давайте посчитаем отношение мужчин и женщин , чтобы создать новое поле, которое будет отображать информацию о том доминирует ли численность мужского населения над женским.

Введите формулу

```
"MALES" / "FEMALES"
```

Перед тем , как нажать кнопку ОК убедитесь. что окно параметров выглядит следуюищм образом

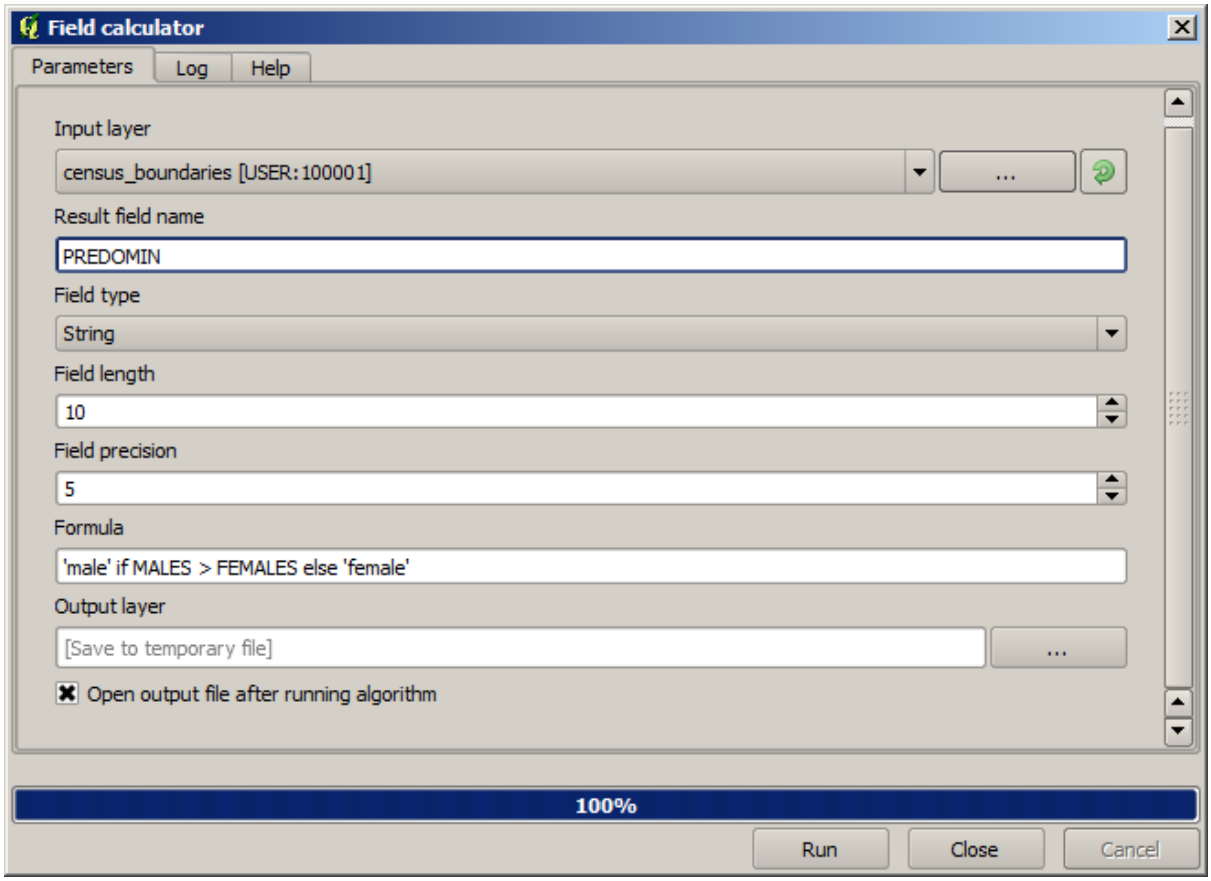


В ранних версиях , если оба поля были типа integer, то результаты вычисления будут сокращены до типа integer. В данном случае формула должна выглядеть: `1.0 * "MALES" / "FEMALES"` , чтобы указать, что результат должен содержать значения с плавающей десятичной запятой.

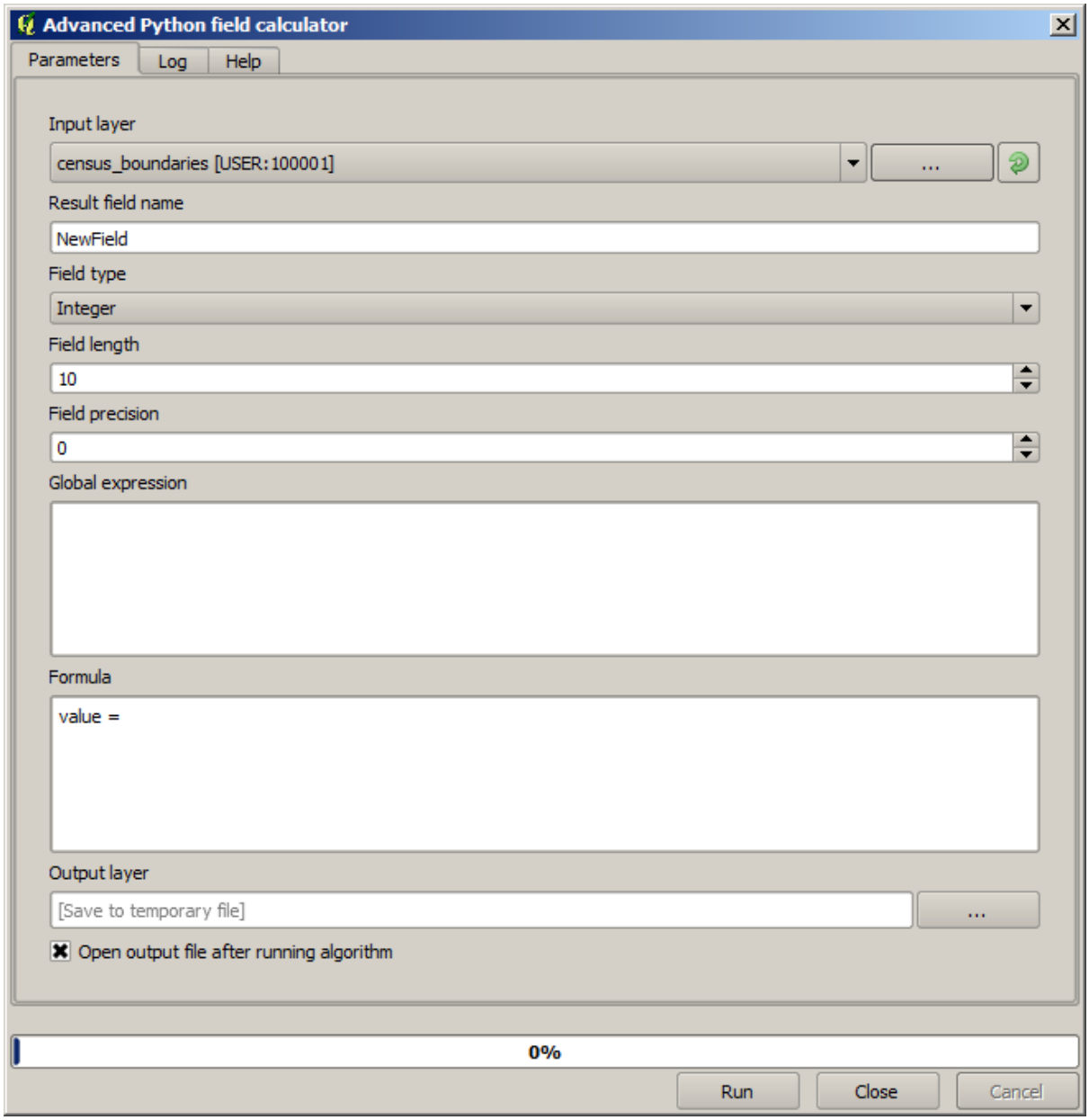
Мы можем воспользоваться фукциями с условиями, чтобы создать новое поле с текстовыми значениями `male` (мужчина) или `female` (женщина) вместо значения соотношения количества мужчин и женщин.

```
CASE WHEN  "MALES" > "FEMALES"  THEN 'male' ELSE 'female' END
```

Окно параметров должно выглядеть следующим образом.



Калькулятор полей, исплозующий язык программирования питон, доступен в *Advanced Python field calculator*. В этом уроке он не будет освещен.



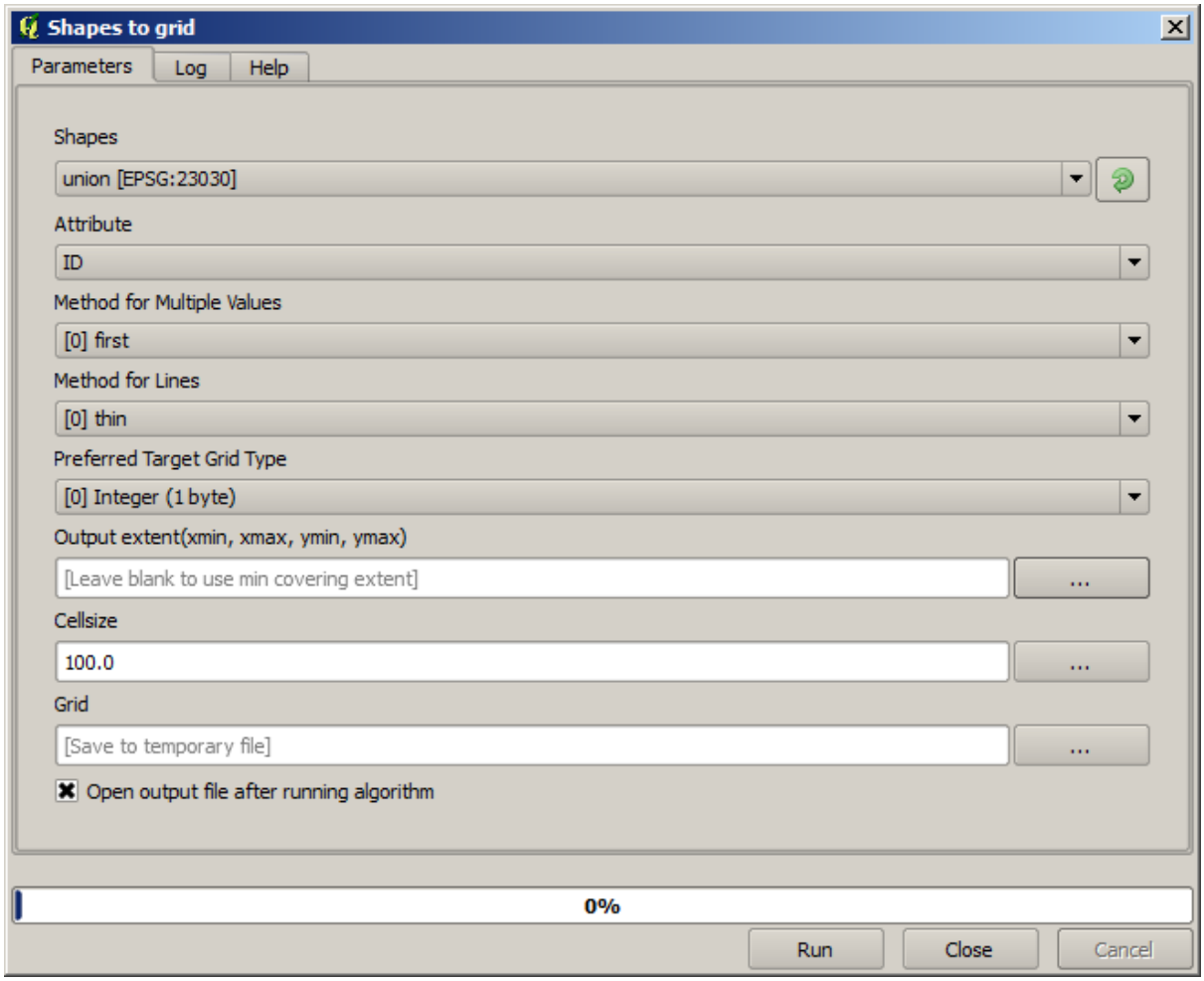
17.12 Defining extents

Примечание: In this lesson we will see how to define extents, which are needed by some algorithms, especially raster ones.

Some algorithms require an extent to define the area to be covered by the analysis they perform, and usually to define the extent of the resulting layer.

When an extent is required, it can be defined manually by entering the four values that define it (min X, min Y, max X, max Y), but there are other more practical and more interesting ways of doing it as well. We will see all of them in this lesson.

First, let's open an algorithm that requires an extent to be defined. Open the *Rasterize* algorithm, which creates a raster layer from a vector layer.

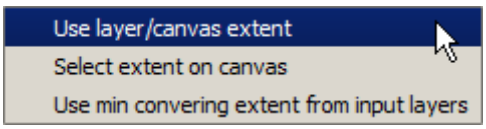


All the parameters, except for the last two ones, are used to define which layer is to be rasterized, and configure how the rasterization process should work. The two last parameters, on the other hand, define the characteristics of the output layer. That means that they define the area that is covered (which is not necessarily the same area covered by the input vector layer), and the resolution/cellsize (which cannot be inferred from the vector layer, since vector layers do not have a cellsize).

The first thing you can do is to type the 4 defining values explained before, separated by commas.

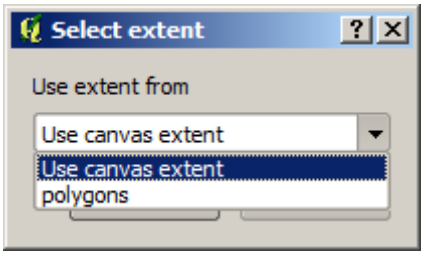


That doesn't need any extra explanation. While this is the most flexible option, it is also the less practical in some cases, and that's why other options are implemented. To access them, you have to click on the button on the right–hand side of the extent text box.



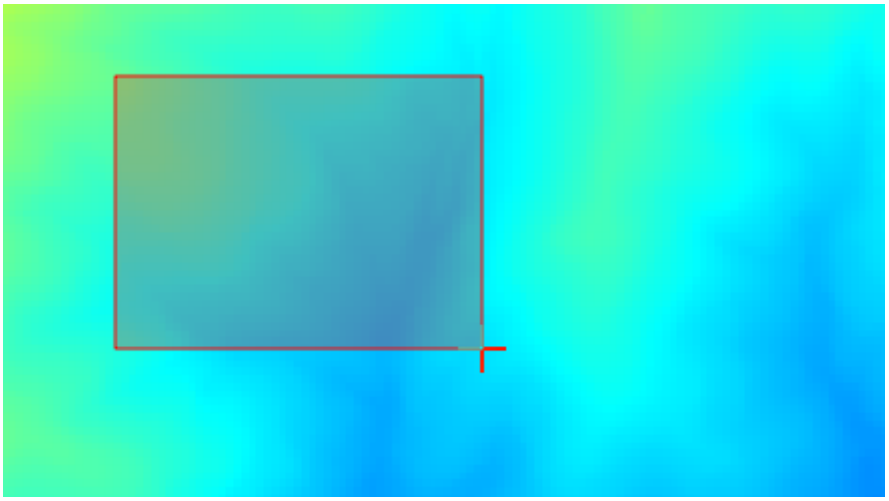
Let's see what each one of them does.

The first option is *Use layer/canvas extent*, which will show the selection dialog shown below.



Here you can select the extent of the canvas (the extent covered by the current zoom), or the extension any of the available layers. Select it and click on *OK*, and the text box will be automatically filled with the corresponding values.

The second option is *Select extent on canvas*. In this case, the algorithm dialog disappears and you can click and drag on the QGIS canvas to define the desired extent.

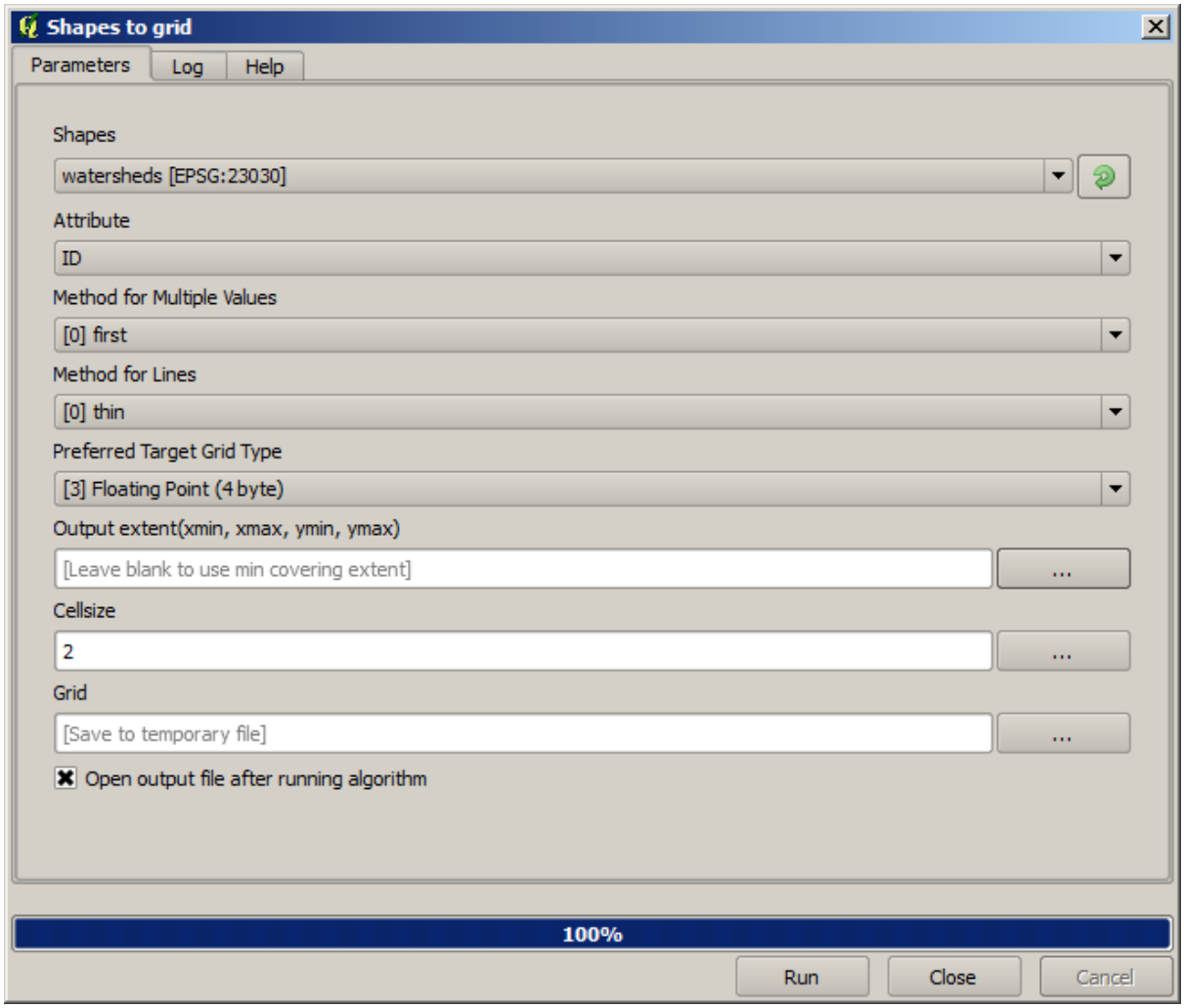


Once you release the mouse button, the dialog will reappear and the text box will already have the values corresponding to the defined extent.

The last option is *Use min covering extent from input layers*, which is the default option. This will compute the min covering extent of all layers used to run the algorithm, and there is no need to enter any value in the text box. In the case of a single input layer, as in the algorithm we are running, the same extent can be obtained by selecting that same input layer in the *Use layer/canvas extent* that we already saw. However, when there are several input layers, the min covering extent does not correspond to any of the input layer extent, since it is computed from all of them together.

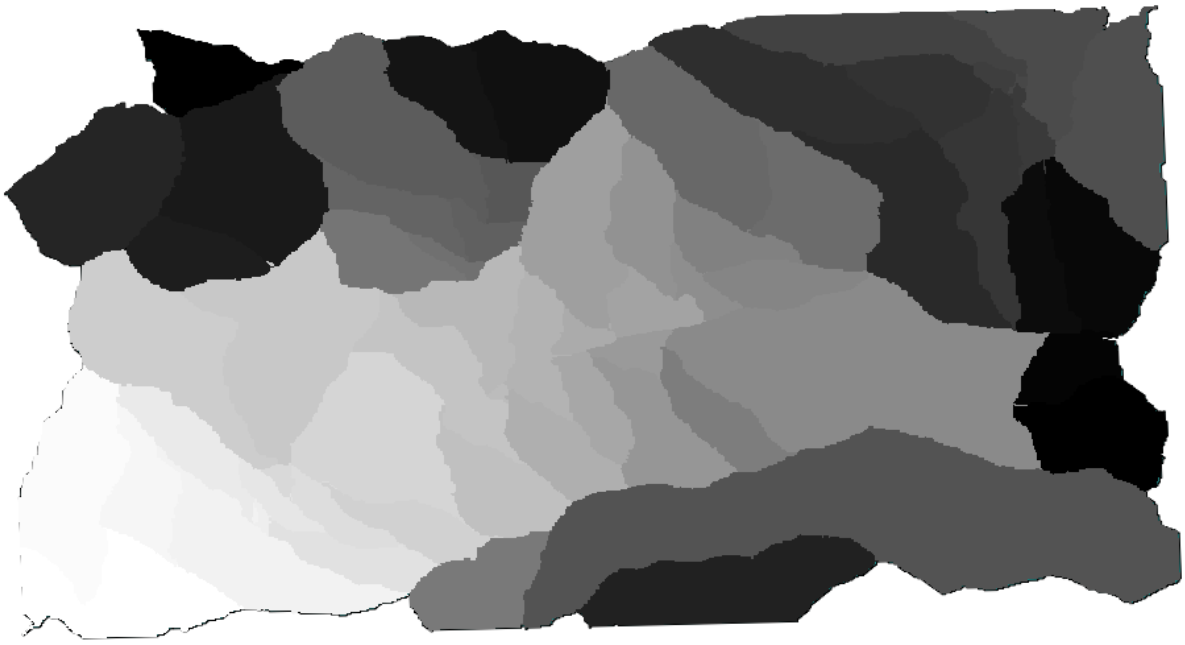
We will use this last method to execute our rasterization algorithm.

Fill the parameters dialog as shown next, and press *OK*.



Примечание: In this case, better use an *Integer (1 byte)* instead of a *Floating point (4 byte)*, since the *NAME* is an integer with maximum value=64. This will result in a smaller file size and faster computations.

You will get a rasterized layer that covers exactly the area covered by the original vector layer.



In some cases, the last option, *Use min covering extent from input layers*, might not be available. This will happen in those algorithm that do not have input layers, but just parameters of other types. In that case, you will have to enter the value manually or use any of the other options.

Notice that, when a selection exist, the extent of the layer is that of the whole set of features, and the selection is not used to compute the extent, even though the rasterization is executed on the selected items only. In that case, you might want to actually create a new layer from the selection, and then use it as input.

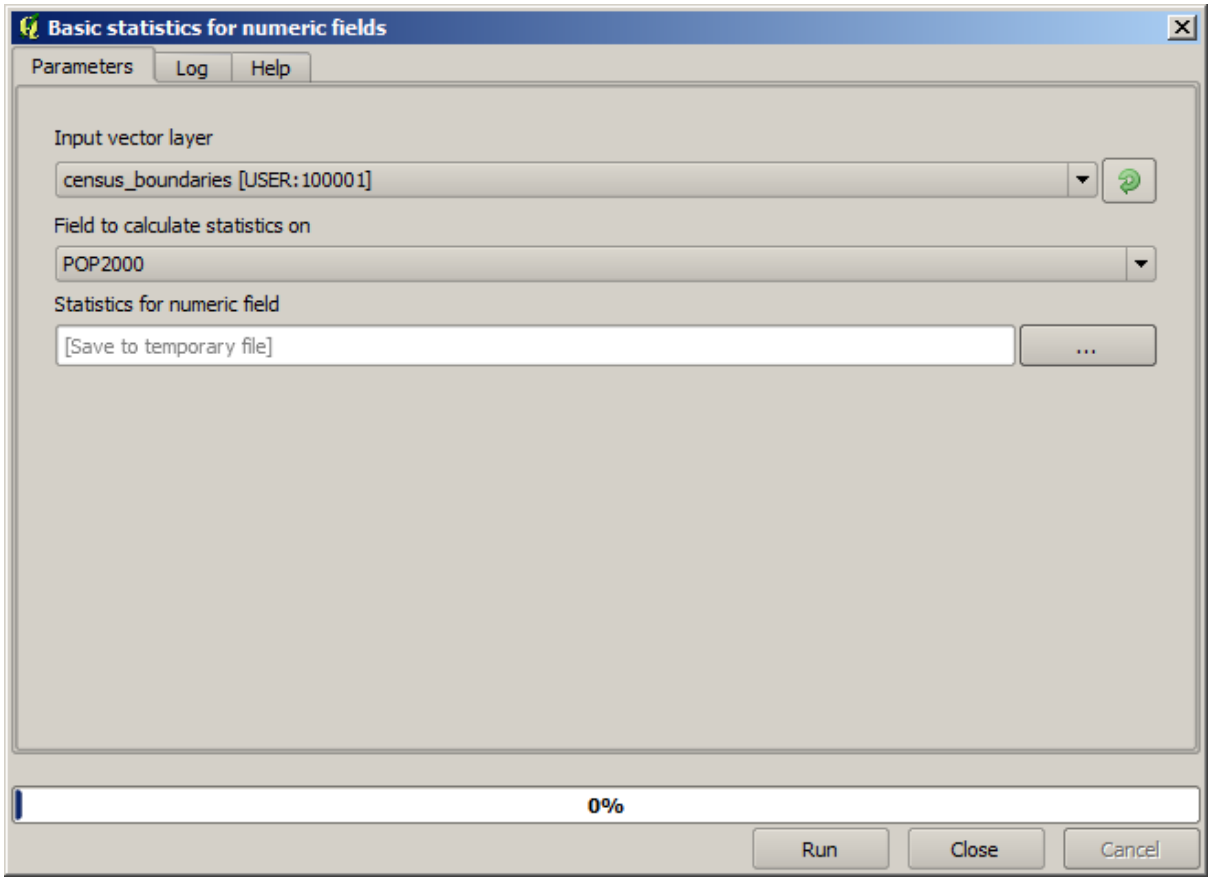
17.13 HTML outputs

Примечание: In this lesson we learn how QGIS handles outputs in HTML format, which are used to produce text outputs and graphs.

All the outputs we have produced so far were layers (whether raster or vector). However, some algorithms generate outputs in the form of text and graphics. All this outputs are wrapped in HTML files and displayed in the so–called *Results viewer*, which is another element of the processing framework.

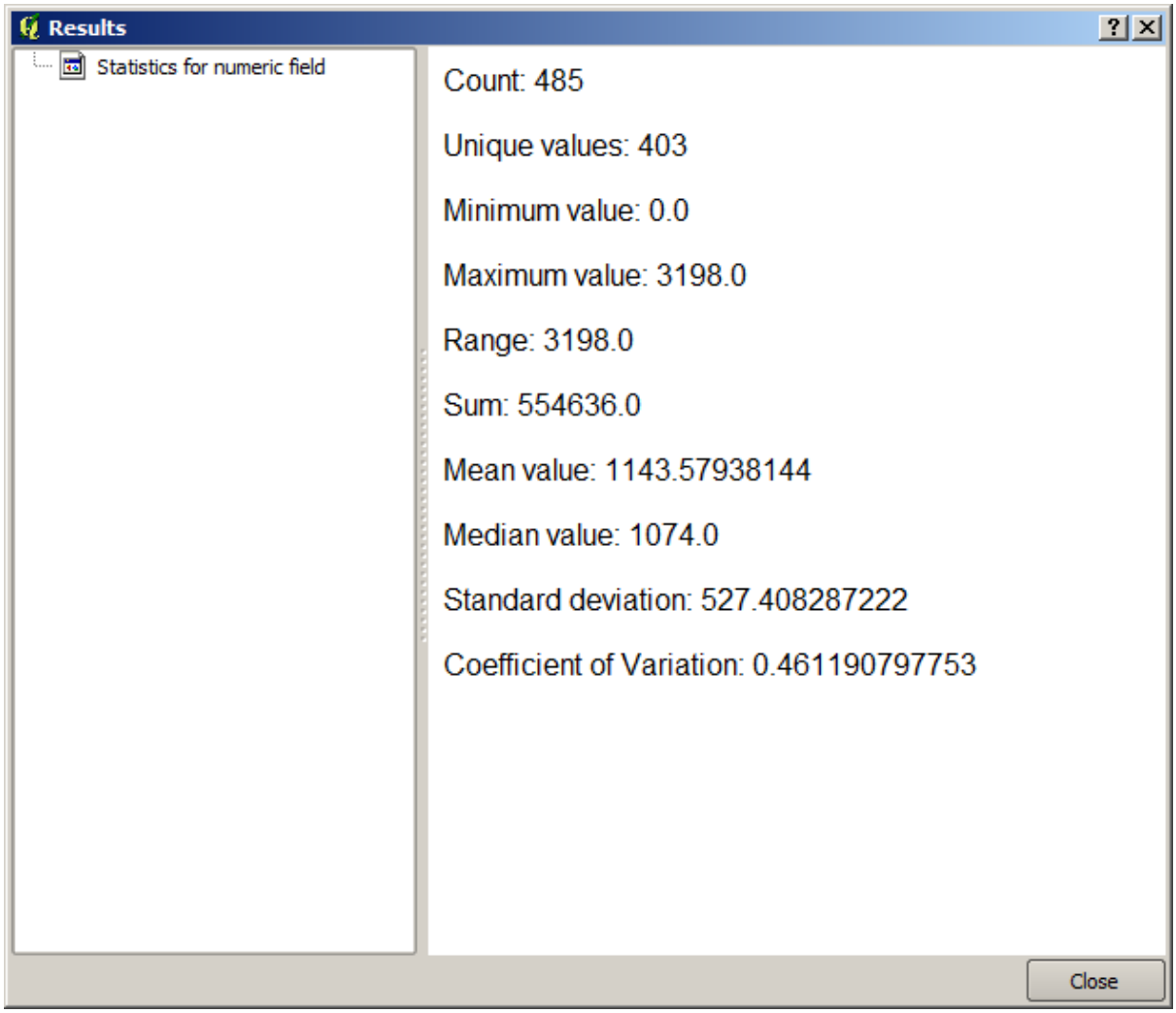
Let's see one of those algorithms to understand how they work.

Open the project with the data to be used in this lesson and then open the *Basic statistics for numeric fields* algorithm.



The algorithm is rather simple, and you just have to select the layer to use and one of its field (a numeric one). The output is of type HTML, but the corresponding box works exactly like the one that you can find in the case of a raster or vector output. You can enter a filepath or leave it blank to save to a temporary file. In this case, however, only the `html` and `htm` extensions are allowed, so there is no way of altering the output format by using a different one.

Run the algorithm selecting the only layer in the project as input, and the *POP2000* field, and a new dialog like the one shown next will appear once the algorithm is executed and the parameters dialog is closed.



This is the *Results viewer*. It keeps all the HTML result generated during the current session, easily accessible, so you can check them quickly whenever you need it. As it happens with layers, if you have saved the output to a temporary file, it will be deleted once you close QGIS. If you have saved to a non-temporary path, the file will remain, but it will not appear in the *Results viewer* the next time you open QGIS.

Some algorithms generate text that cannot be divided into other more detailed outputs. That is the case if, for instance, the algorithm captures the text output from an external process. In other cases, the output is presented as text, but internally is divided into several smaller outputs, usually in the form of numeric values. The algorithm that we have just executed is one of them. Each one of those values is handled as a single output, and stored in a variable. This has no importance at all now, but once we move to the graphical modeler, you will see that it will allow us to use those values as numeric inputs for other algorithms.

17.14 First analysis example

Примечание: In this lesson we will perform some real analysis using just the toolbox, so you can get more familiar with the processing framework elements.

Now that everything is configured and we can use external algorithms, we have a very powerful tool to perform spatial analysis. It is time to work out a larger exercise with some real–world data.

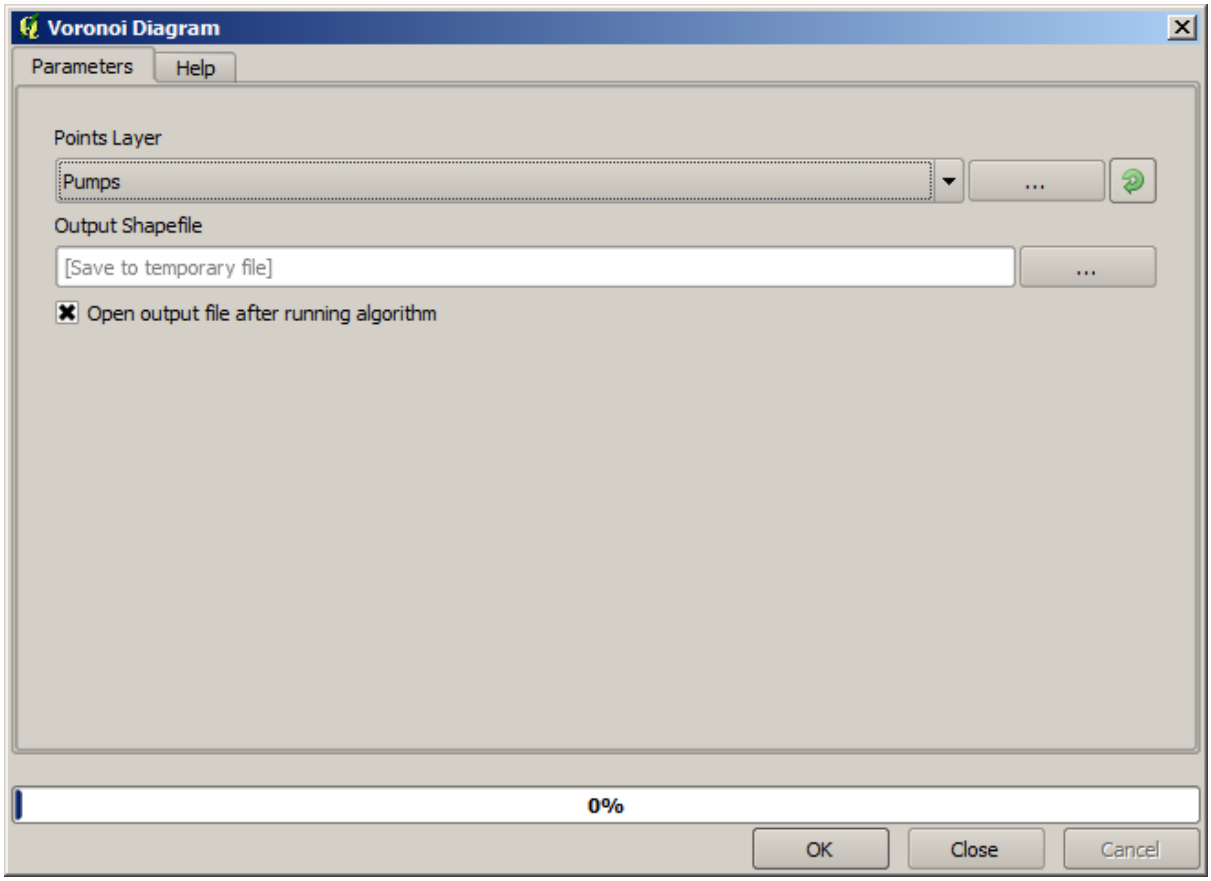
We will be using the well-known dataset that John Snow used in 1854, in his groundbreaking work (https://en.wikipedia.org/wiki/John_Snow_%28physician%29), and we will get some interesting results. The analysis of this

dataset is pretty obvious and there is no need for sophisticated GIS techniques to end up with good results and conclusions, but it is a good way of showing how these spatial problems can be analyzed and solved by using different processing tools.

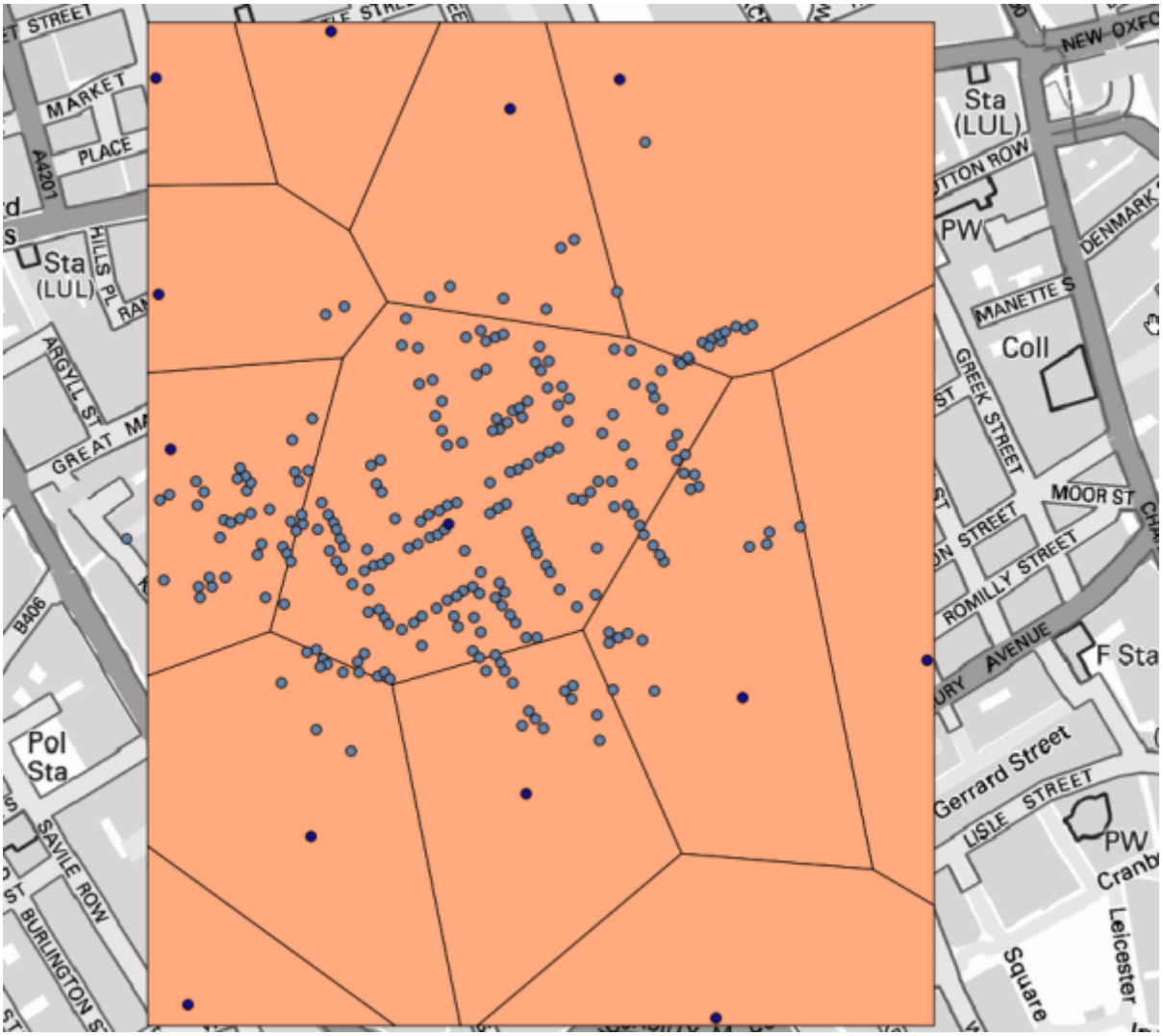
The dataset contains shapefiles with cholera deaths and pump locations, and an OSM rendered map in TIFF format. Open the corresponding QGIS project for this lesson.



The first thing to do is to calculate the Voronoi diagram (a.k.a. Thiessen polygons) of the pumps layer, to get the influence zone of each pump. The *Voronoi Diagram* algorithm can be used for that.

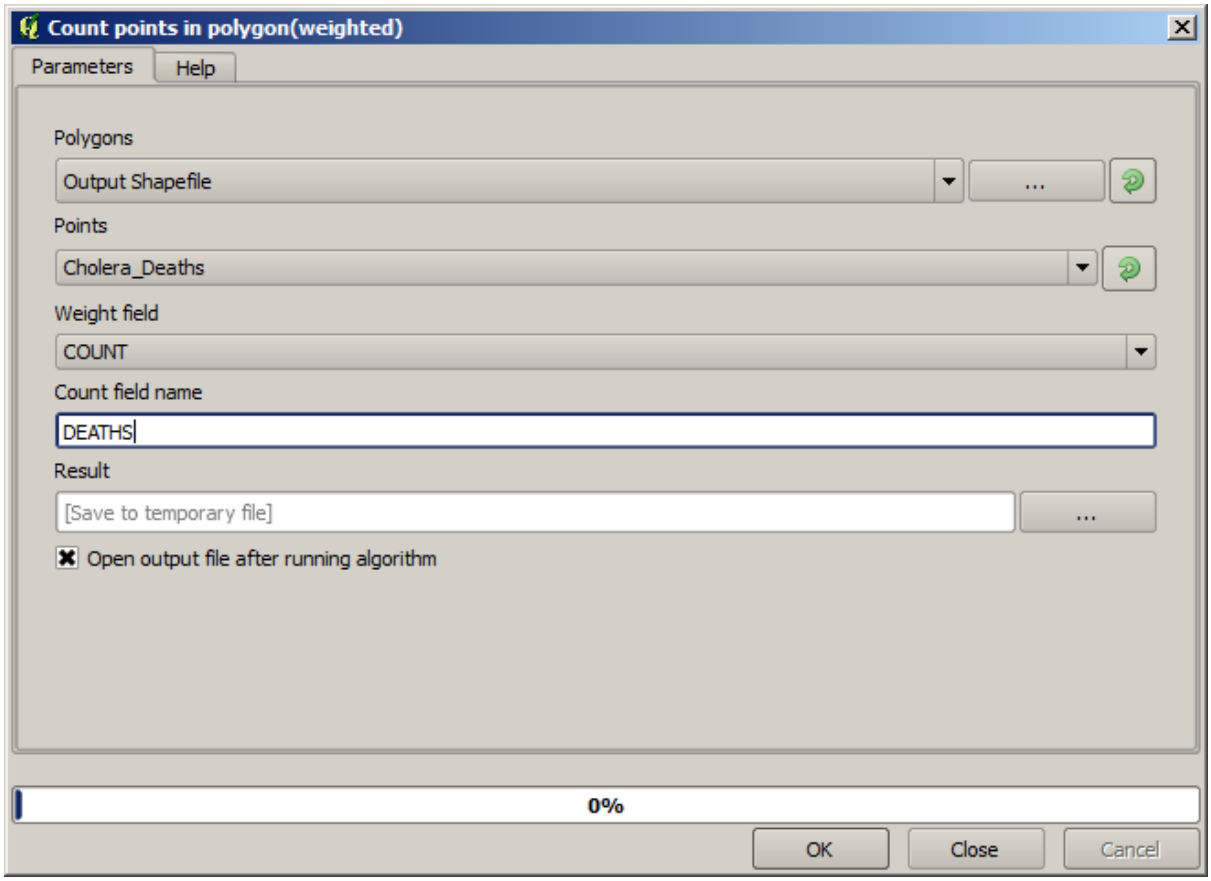


Pretty easy, but it will already give us interesting information.

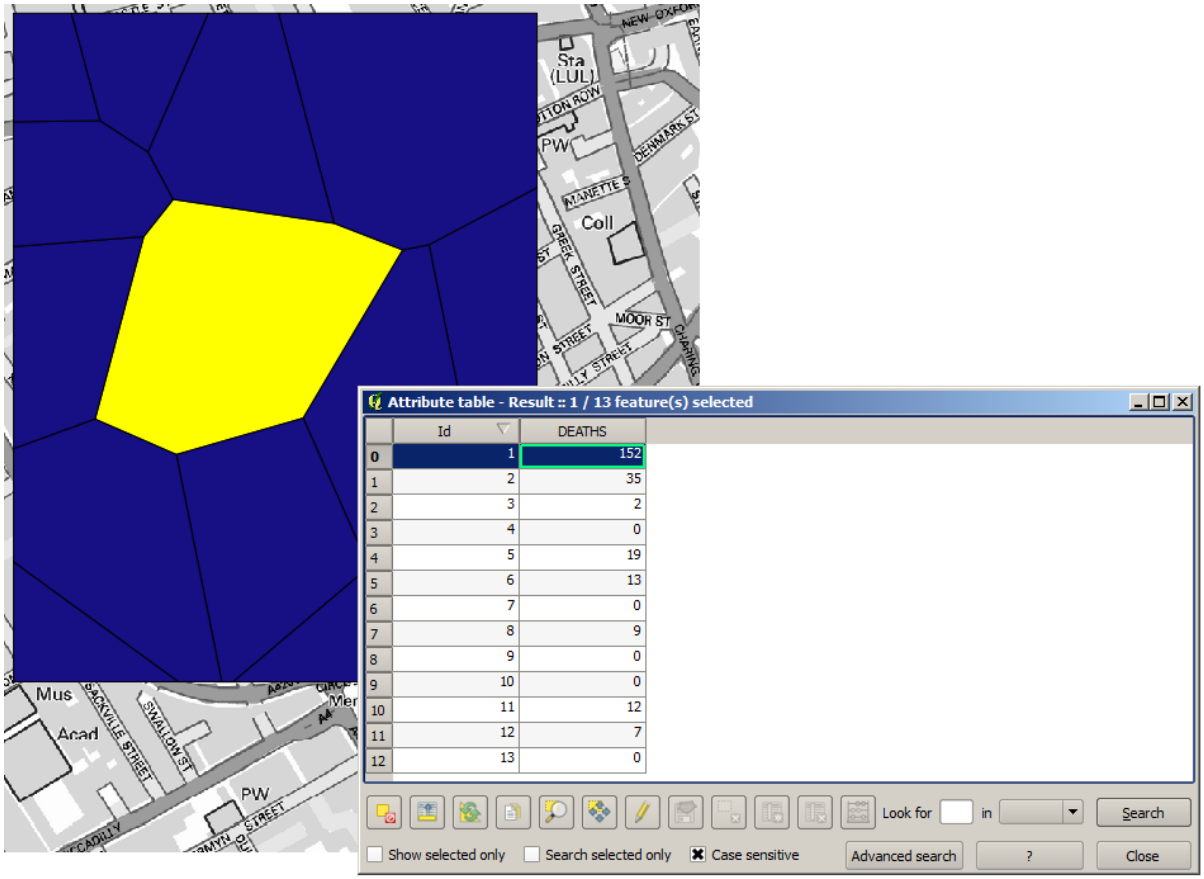


Clearly, most cases are within one of the polygons

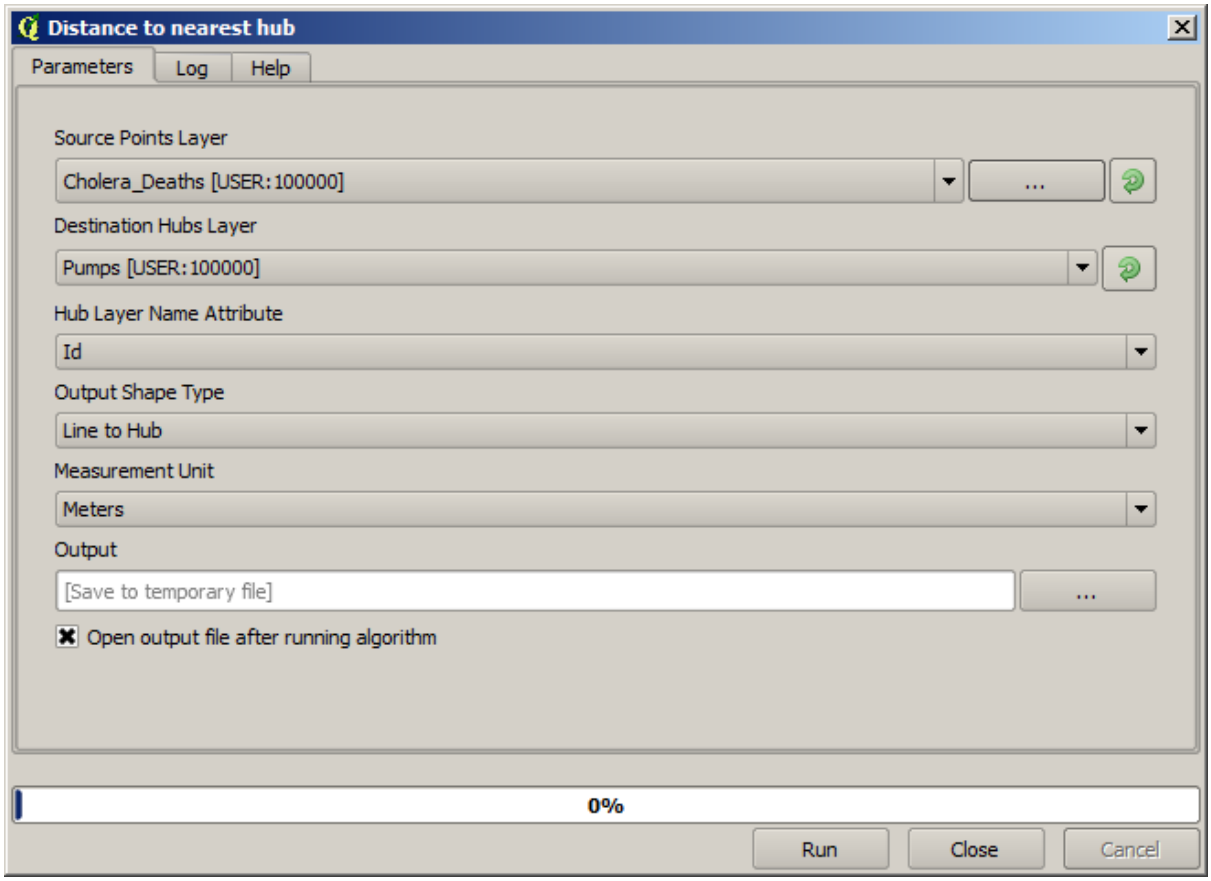
To get a more quantitative result, we can count the number of deaths in each polygon. Since each point represents a building where deaths occured, and the number of deaths is stored in an attribute, we cannot just count the points. We need a weighted count, so we will use the *Count points in polygon (weighted)* tool.



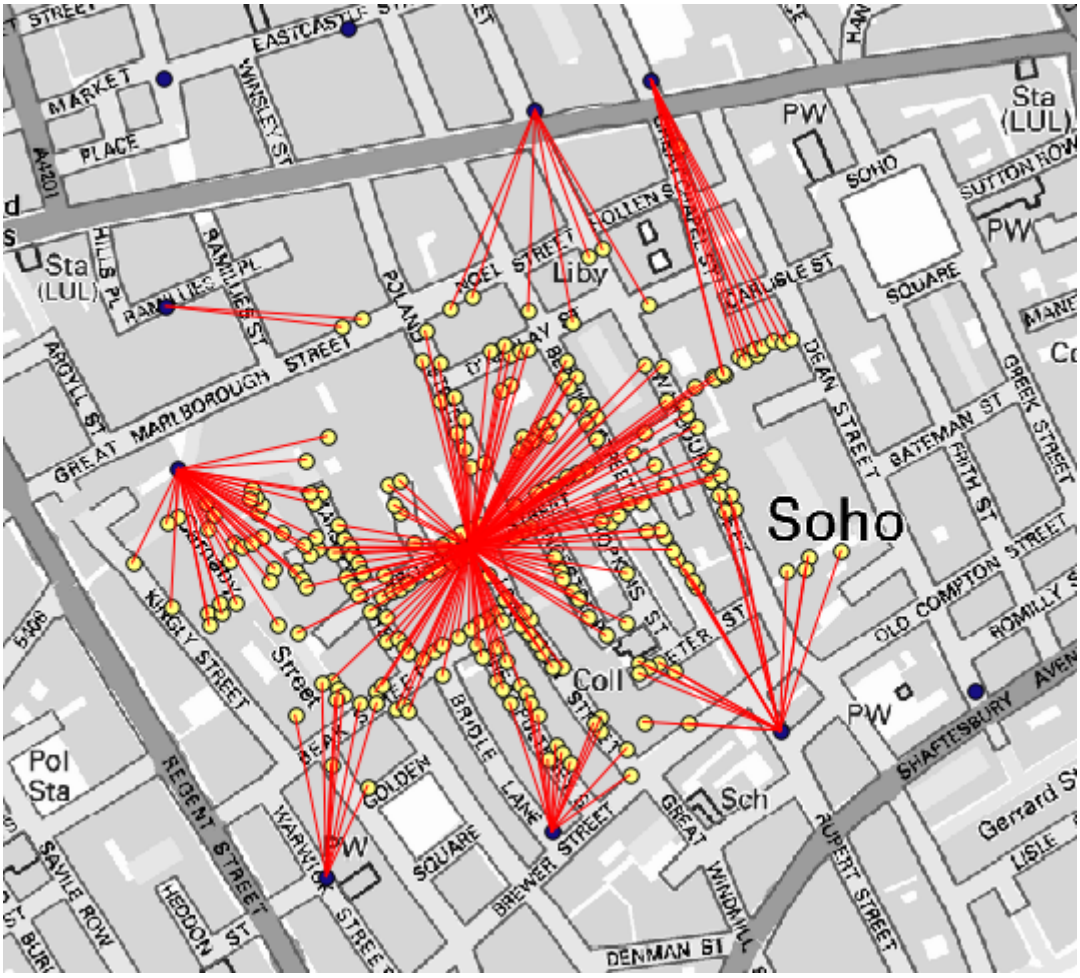
The new field will be called *DEATHS*, and we use the *COUNT* field as weighting field. The resulting table clearly reflects that the number of deaths in the polygon corresponding to the first pump is much larger than the other ones.



Another good way of visualizing the dependence of each point in the `Cholera_deaths` layer with a point in the `Pumps` layer is to draw a line to the closest one. This can be done with the *Distance to nearest hub* tool, and using the configuration shown next.

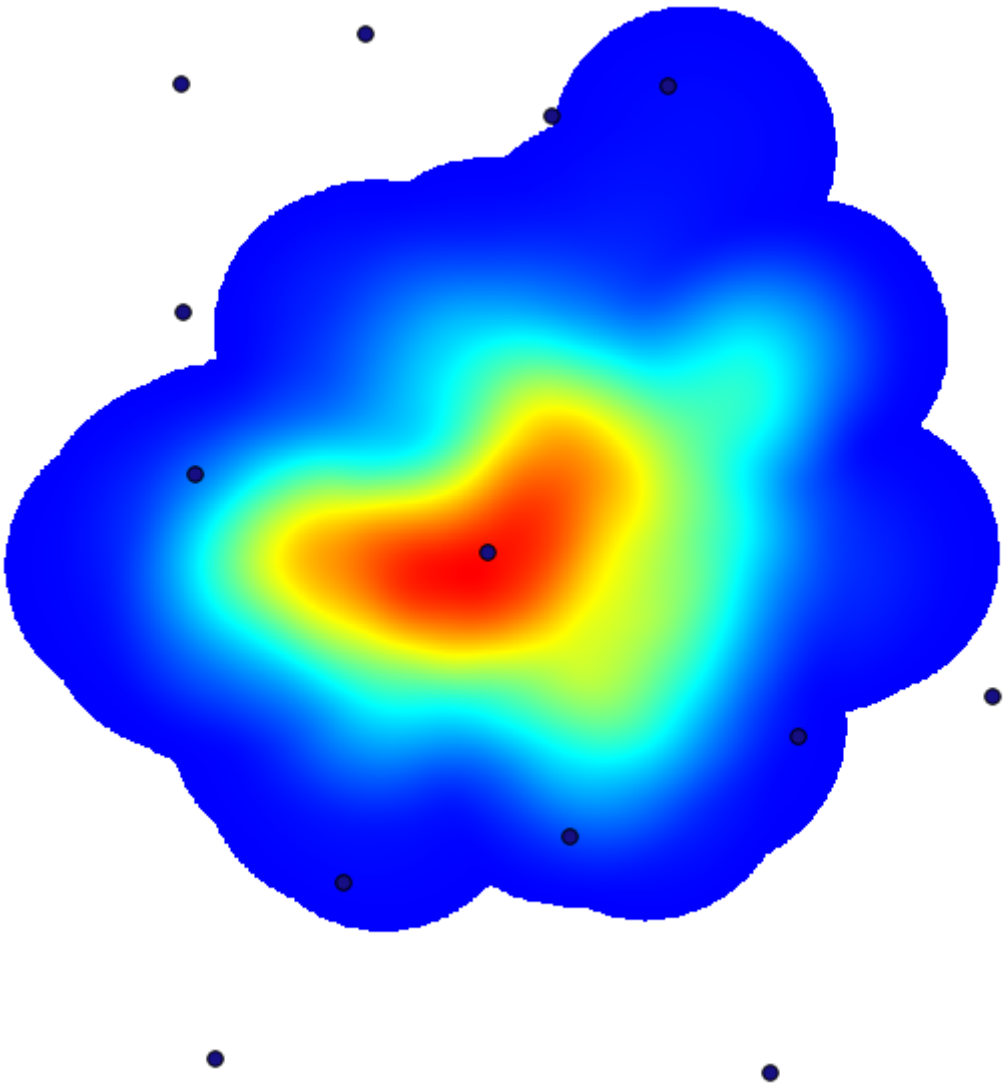


The result looks like this:

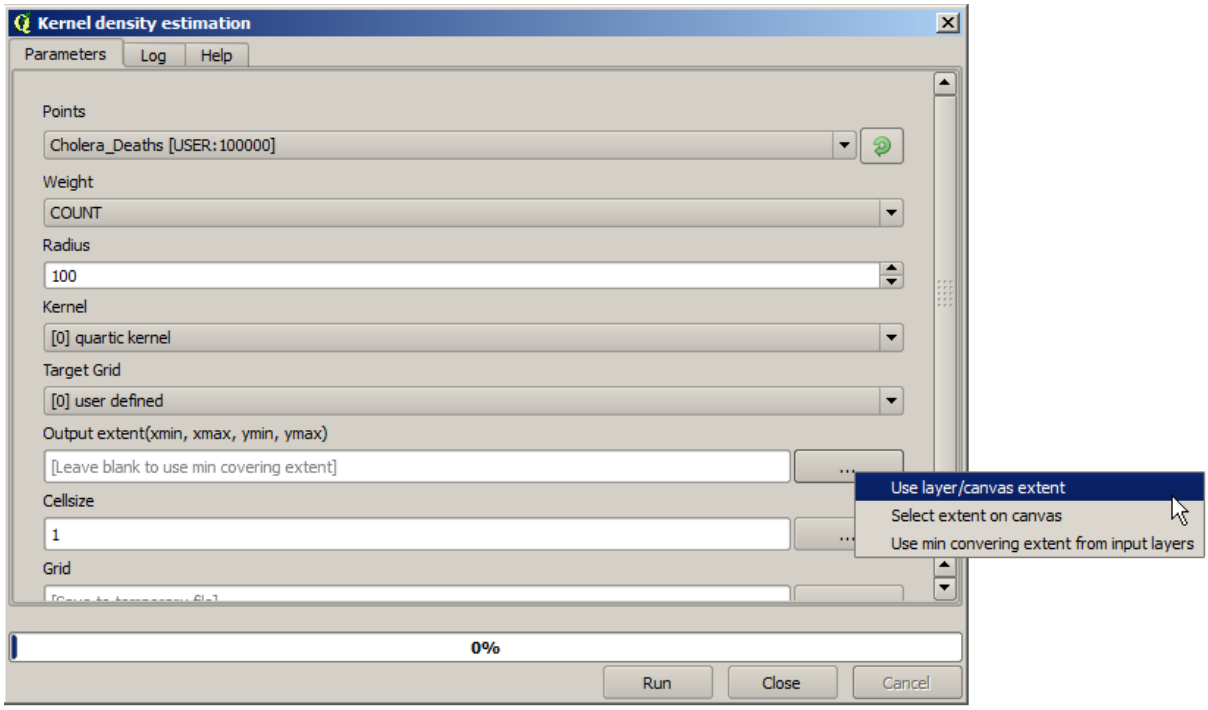


Although the number of lines is larger in the case of the central pump, do not forget that this does not represent the number of deaths, but the number of locations where cholera cases were found. It is a representative parameter, but it is not considering that some locations might have more cases than other.

A density layer will also give us a very clear view of what is happening. We can create it with the *Kernel density* algorithm. Using the *Cholera_deaths* layer, its *COUNT* field as weight field, with a radius of 100, the extent and cellsize of the streets raster layer, we get something like this.



Remember that, to get the output extent, you do not have to type it. Click on the button on the right-hand side and select *Use layer/canvas extent*.



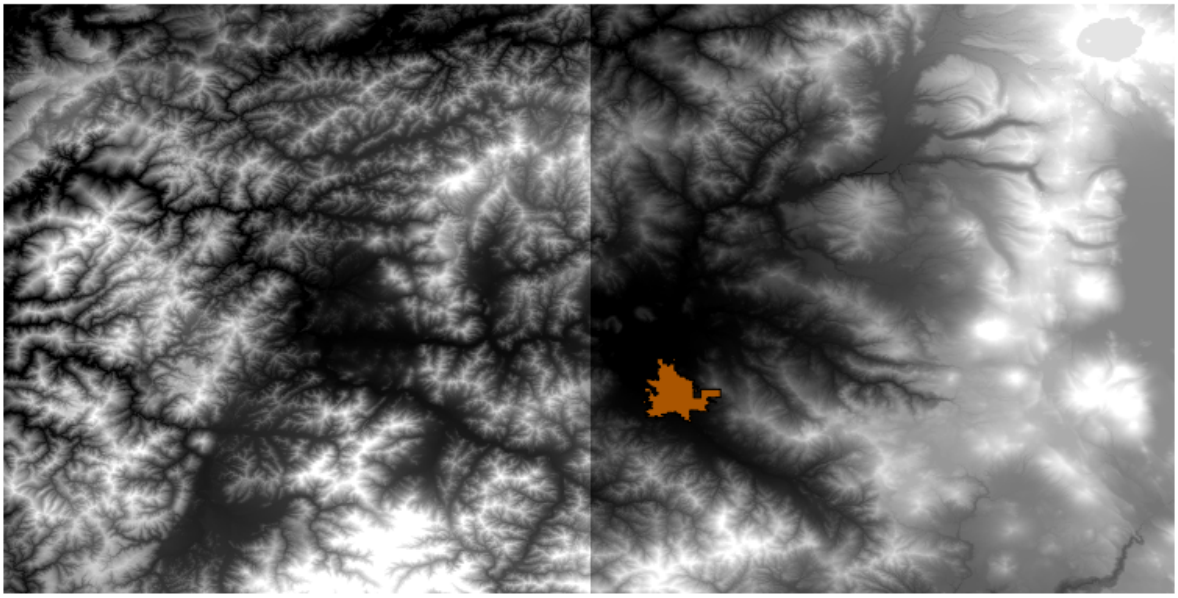
Select the streets raster layer and its extent will be automatically added to the text field. You must do the same with the cellsize, selecting the cellsize of that layer as well.

Combining with the pumps layer, we see that there is one pump clearly in the hotspot where the maximum density of death cases is found.

17.15 Clipping and merging raster layers

Примечание: In this lesson we will see another example of spatial data preparation, to continue using geoalgorithms in real-world scenarios.

For this lesson, we are going to calculate a slope layer for an area surrounding a city area, which is given in a vector layer with a single polygon. The base DEM is divided in two raster layers that, together, cover an area much larger than that around the city that we want to work with. If you open the project corresponding to this lesson, you will see something like this.



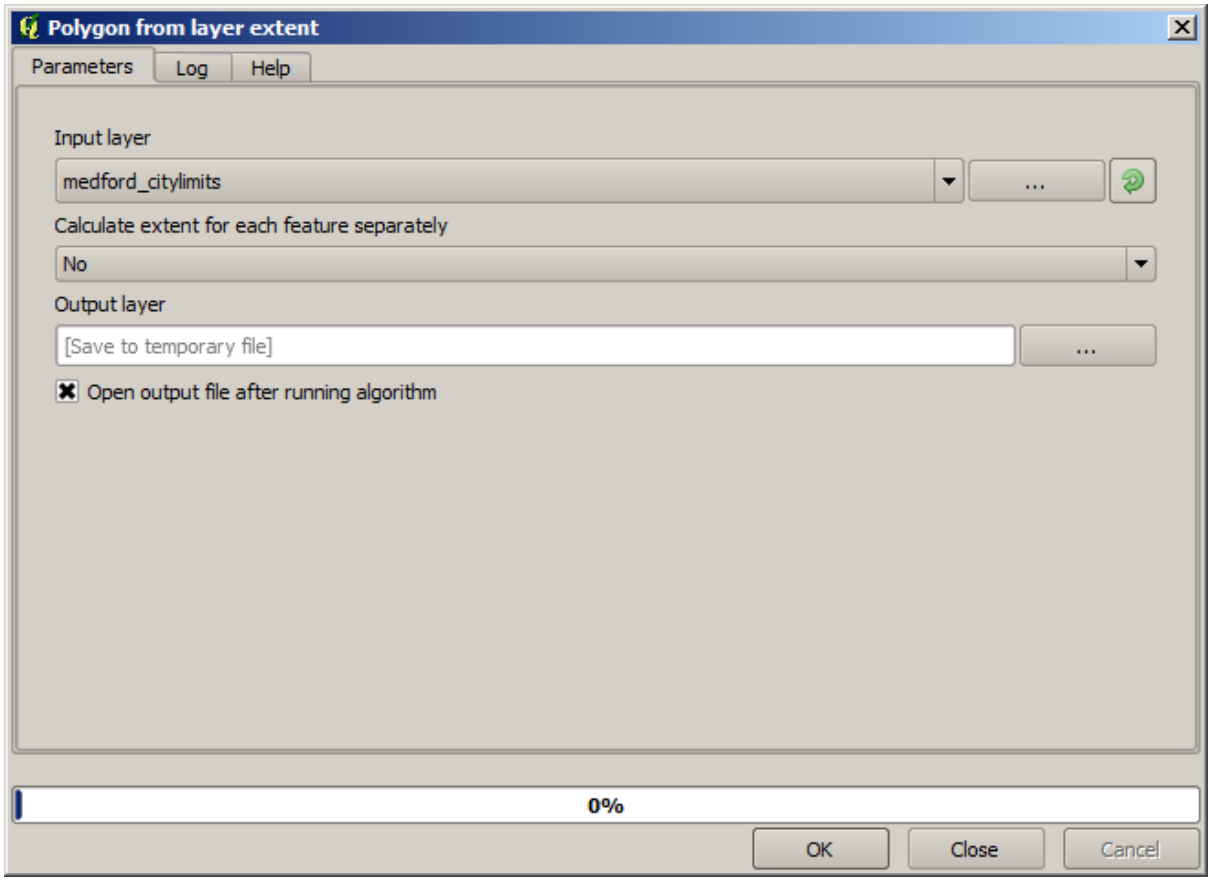
These layers have two problems:

- They cover an area that is too large for what we want (we are interested in a smaller region around the city center)
- They are in two different files (the city limits fall into just one single raster layer, but, as it's been said, we want some extra area around it).

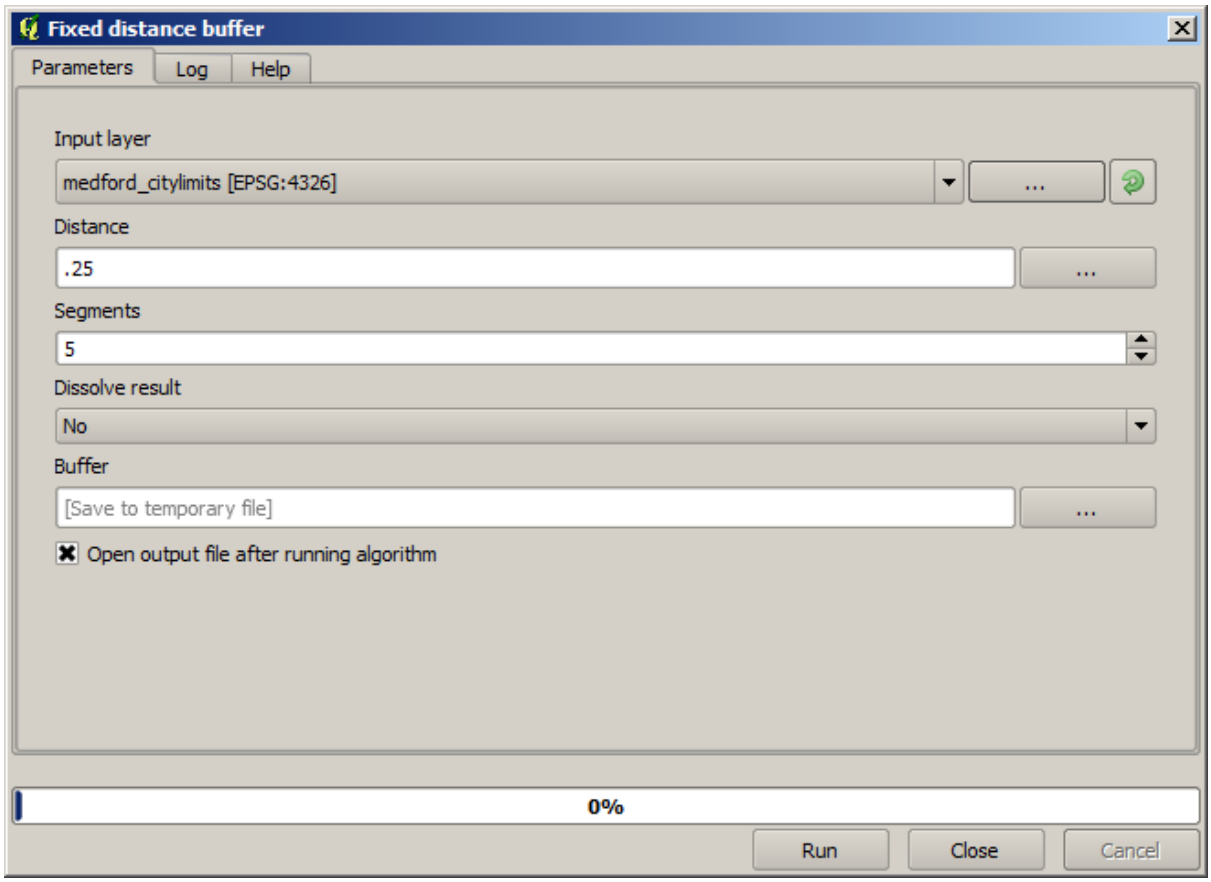
Both of them are easily solvable with the appropriate geoalgorithms.

First, we create a rectangle defining the area that we want. To do it, we create a layer containing the bounding box of the layer with the limits of the city area, and then we buffer it, so as to have a raster layer that covers a bit more that the strictly necessary.

To calculate the bounding box , we can use the *Polygon from layer extent* algorithm

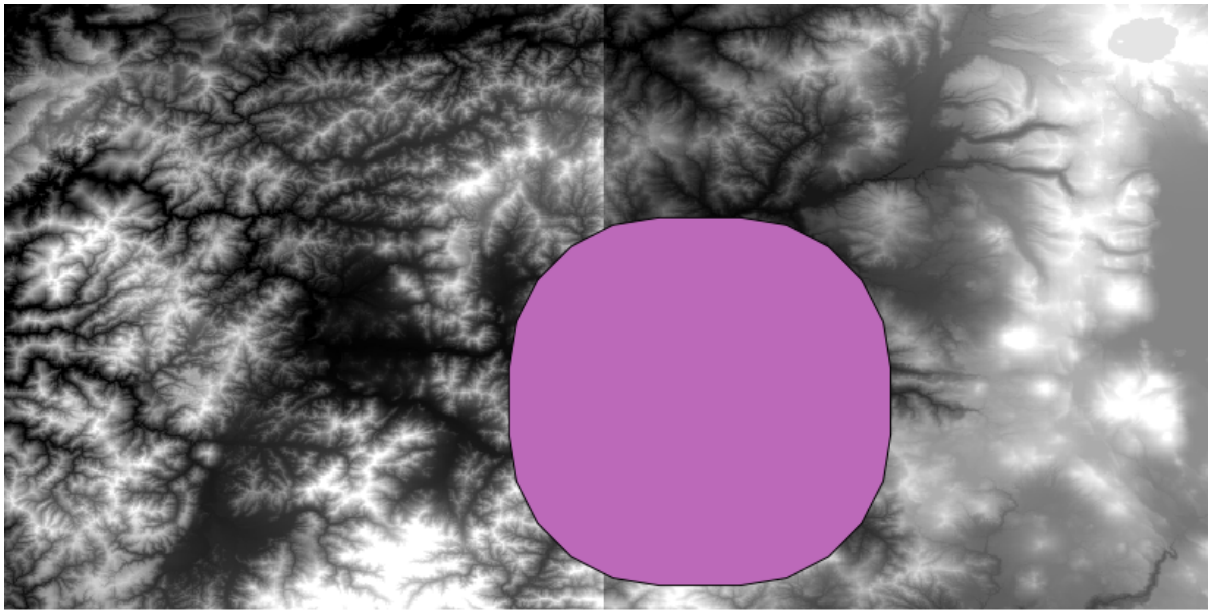


To buffer it, we use the *Fixed distance buffer* algorithm, with the following parameter values.

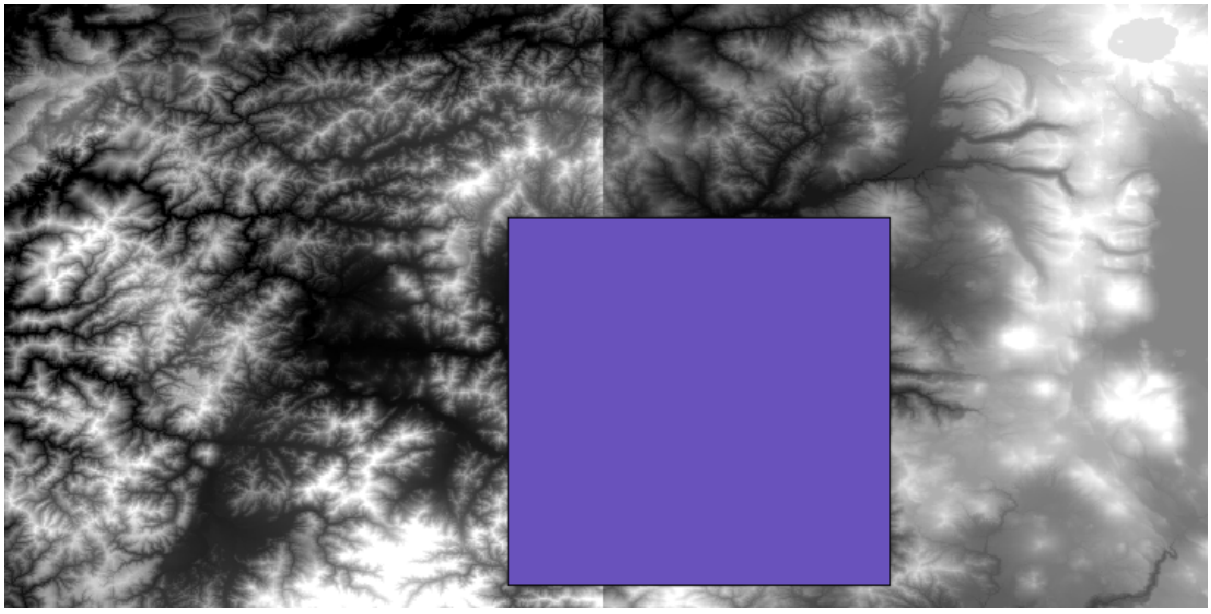


Предупреждение: Syntax changed in recent versions; set both Distance and Arc vertex to .25

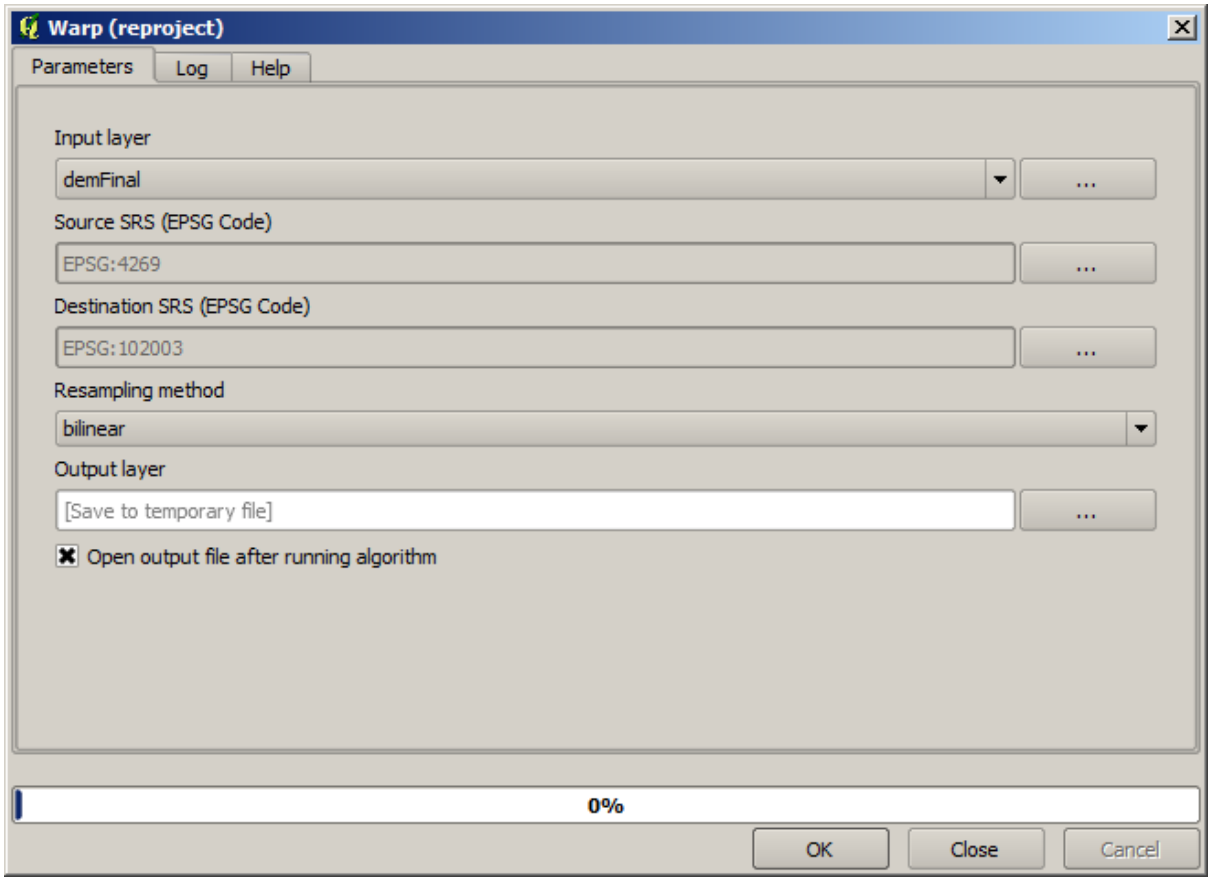
Here is the resulting bounding box obtained using the parameters shown above



It is a rounded box, but we can easily get the equivalent box with square angles, by running the *Polygon from layer extent* algorithm on it. We could have buffered the city limits first, and then calculate the extent rectangle, saving one step.

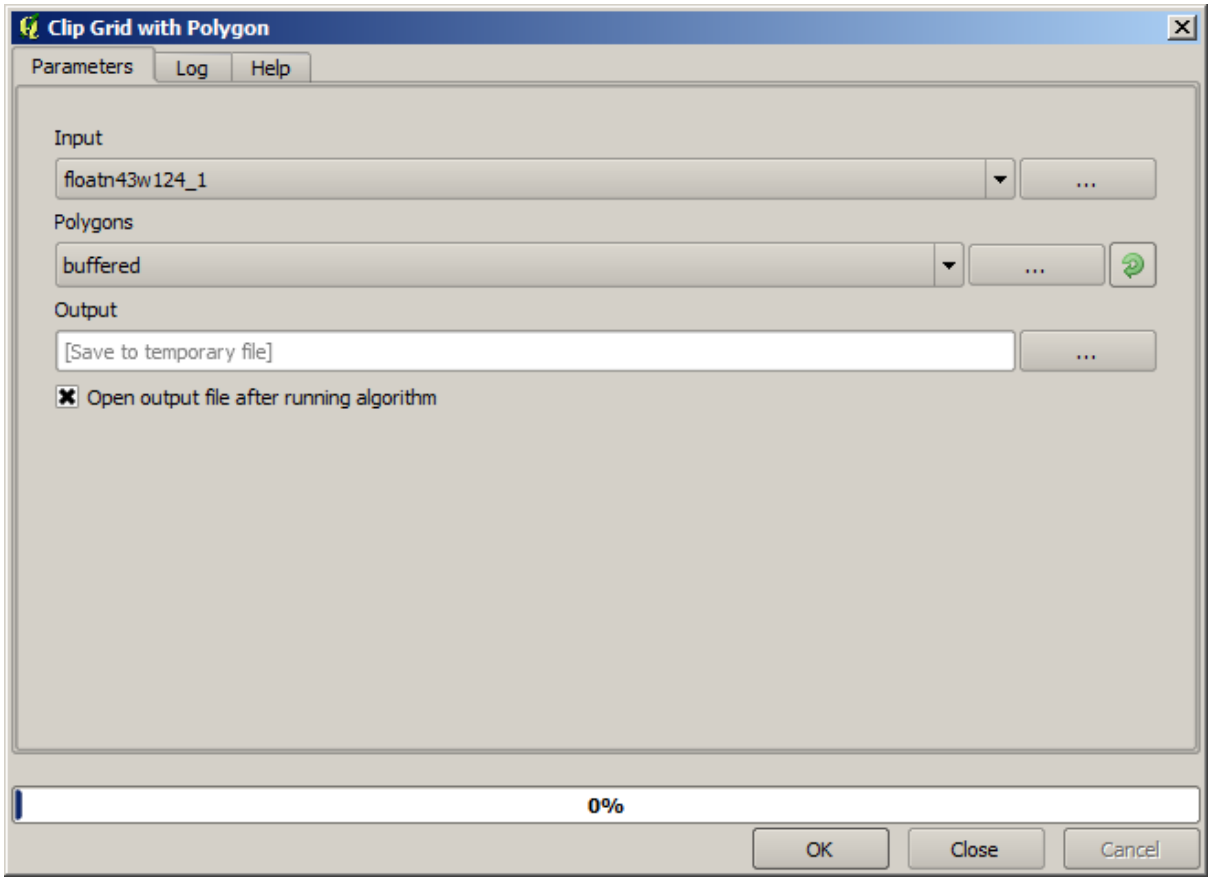


You will notice that the rasters has a different projection from the vector. We should therefore reproject them before proceeding further, using the *Warp (reproject)* tool.

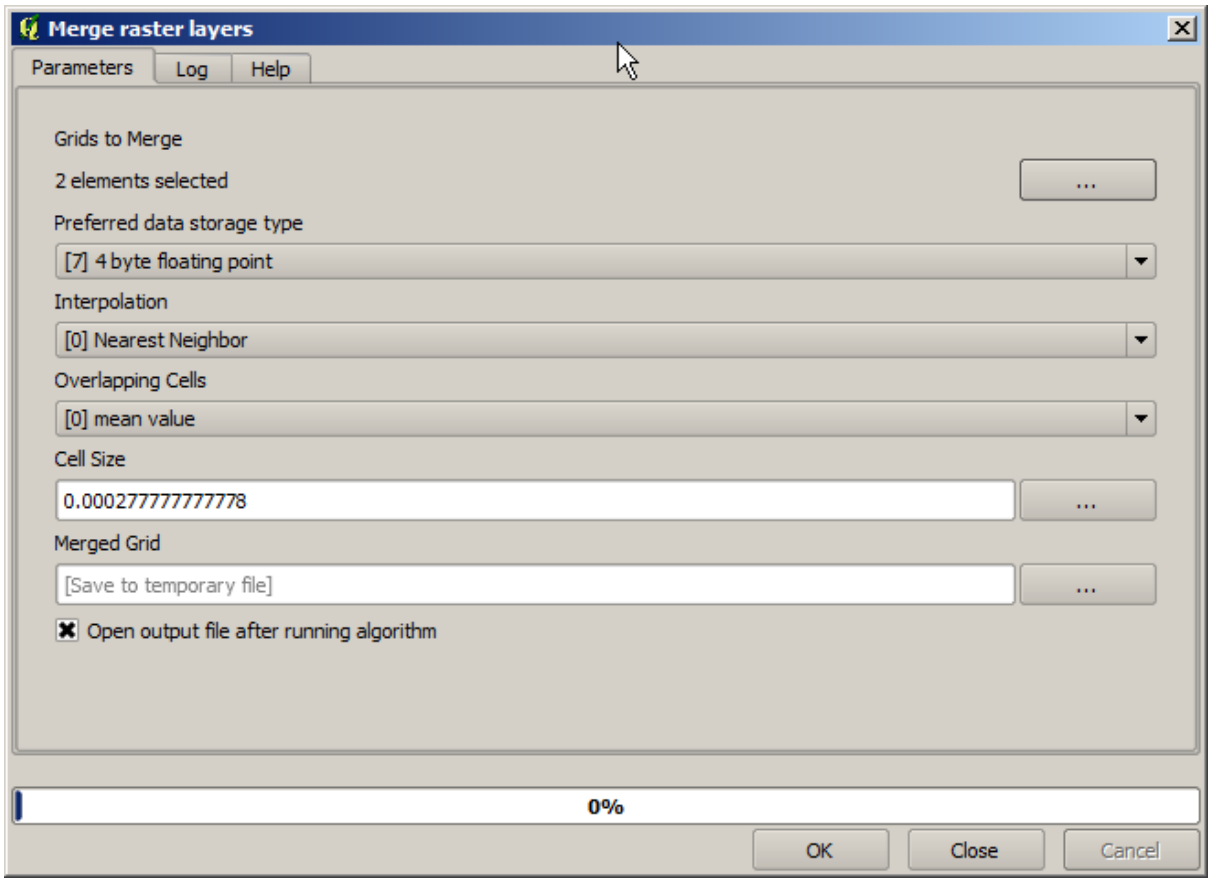


Примечание: Recent versions have a more complex interface. Make sure at least one compression method is selected.

With this layer that contains the bounding box of the raster layer that we want to obtain, we can crop both of the raster layers, using the *Clip raster with polygon* algorithm.

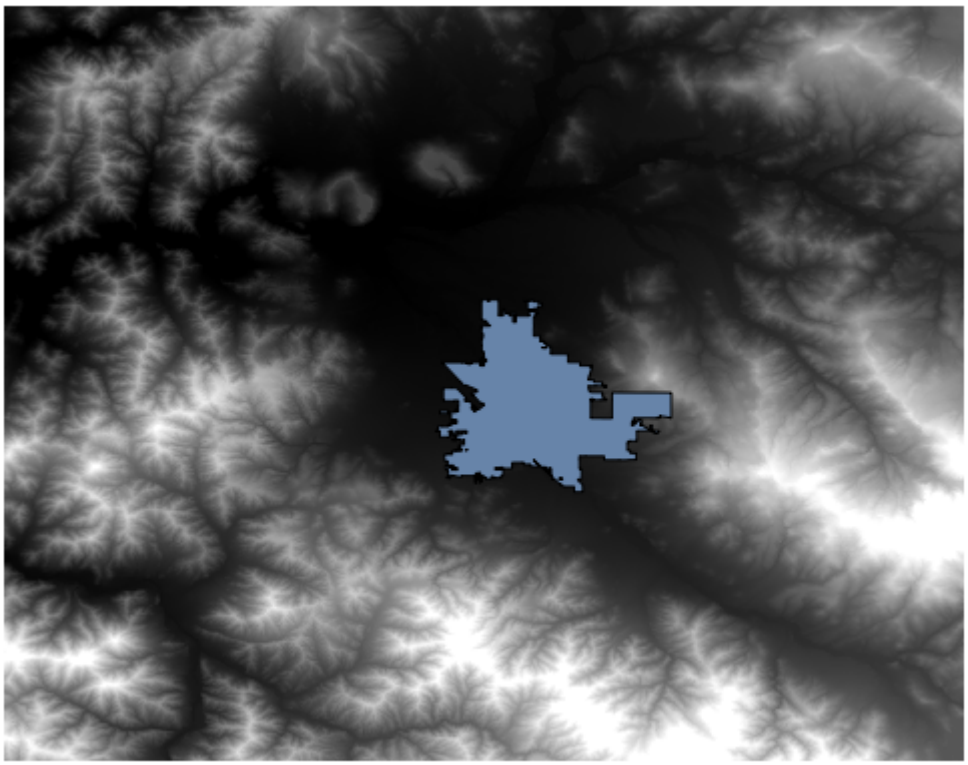


Once the layers have been cropped, they can be merged using the SAGA *Mosaic raster layers* algorithm.



Примечание: You can save time merging first and then cropping, and you will avoid calling the clipping algorithm twice. However, if there are several layers to merge and they have a rather big size, you will end up with a large layer than it can later be difficult to process. In that case, you might have to call the clipping algorithm several times, which might be time consuming, but don't worry, we will soon see that there are some additional tools to automate that operation. In this example, we just have two layers, so you shouldn't worry about that now.

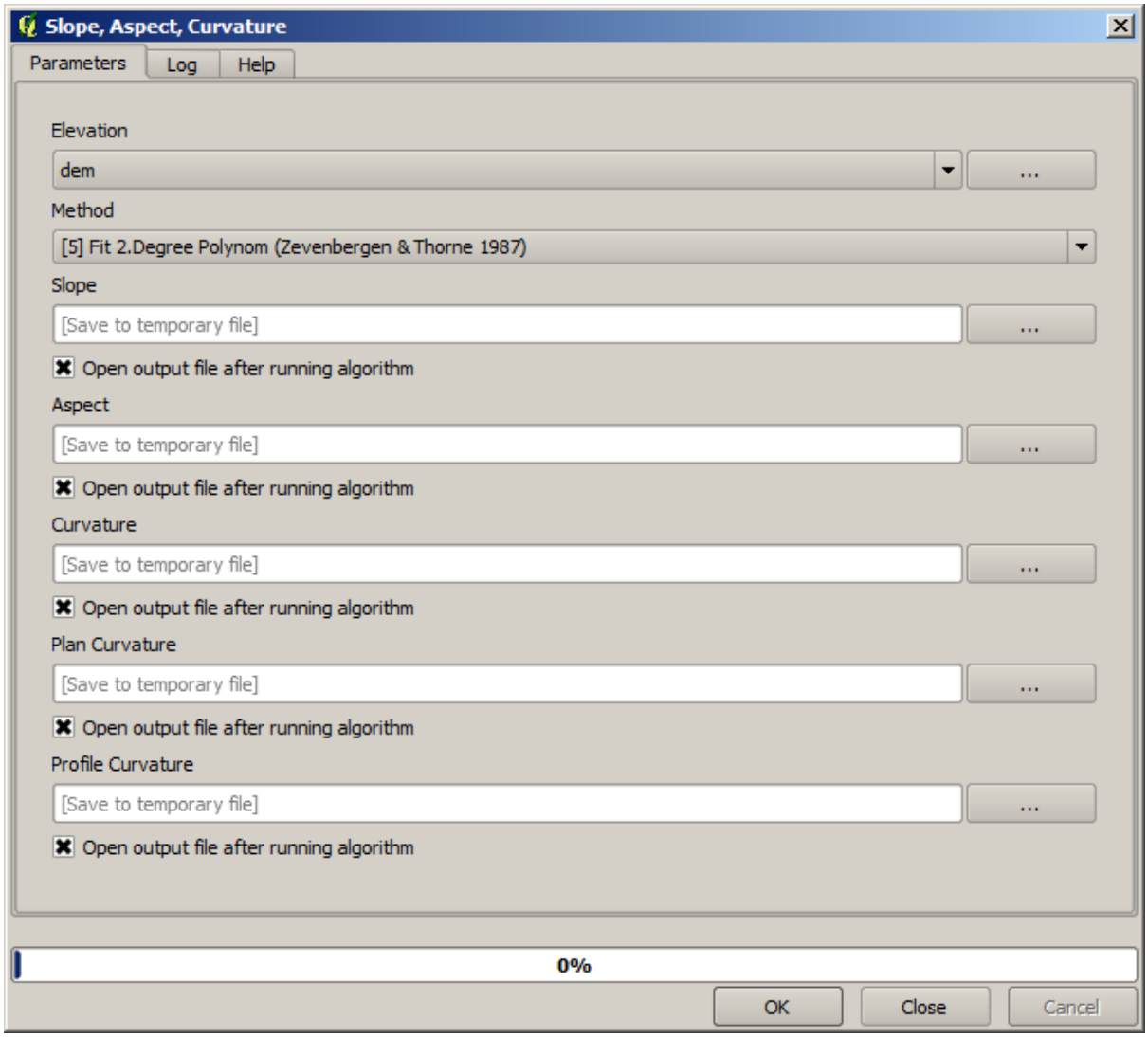
With that, we get the final DEM we want.



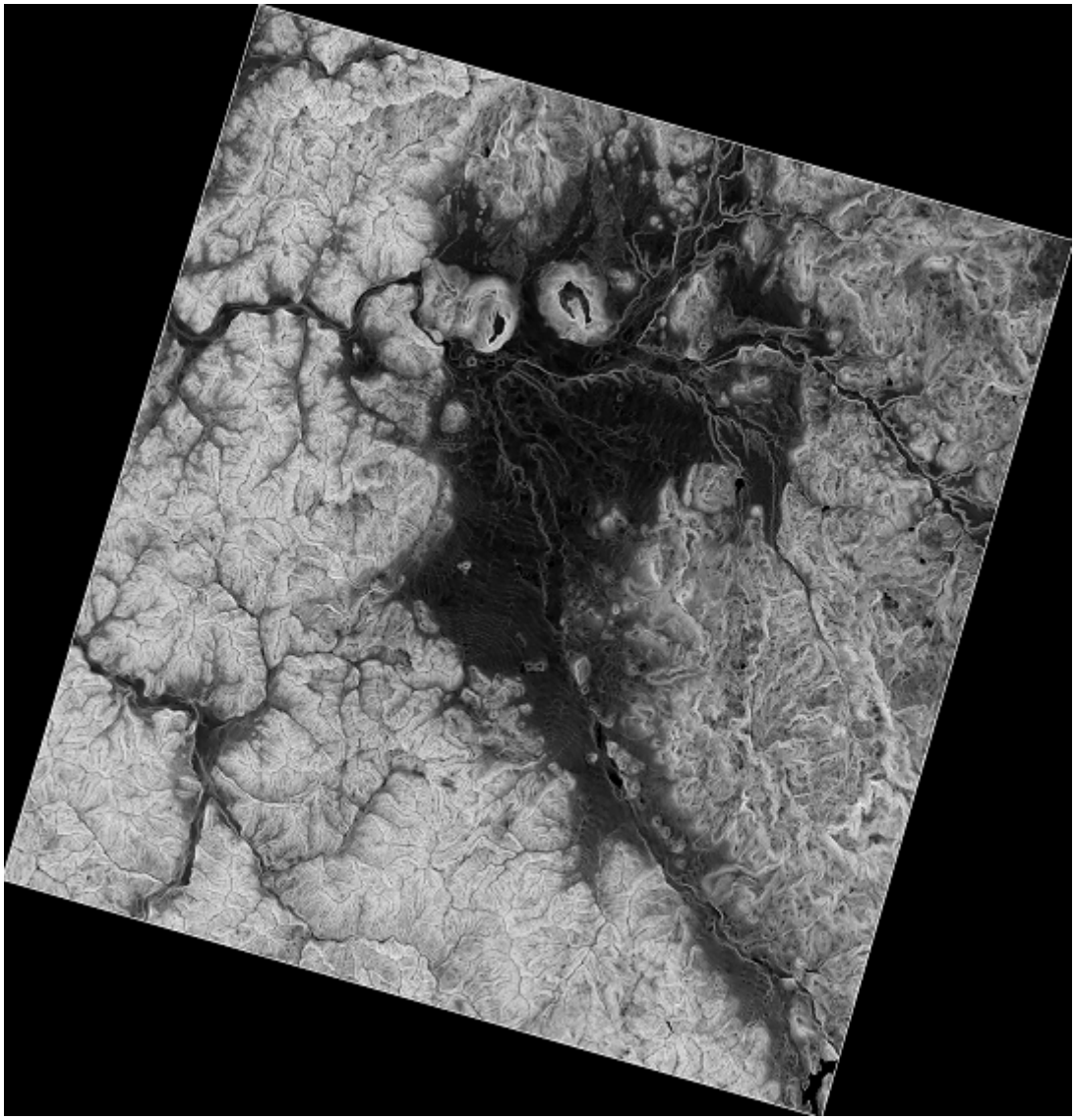
Now it is time to compute the slope layer.

A slope layer can be computed with the *Slope, Aspect, Curvature* algorithm, but the DEM obtained in the last step is not suitable as input, since elevation values are in meters but cellsize is not expressed in meters (the layer uses a CRS with geographic coordinates). A reprojection is needed. To reproject a raster layer, the *Warp (reproject)* algorithm can be used again. We reproject into a CRS with meters as units (e.g. 3857), so we can then correctly calculate the slope, with either SAGA or GDAL.

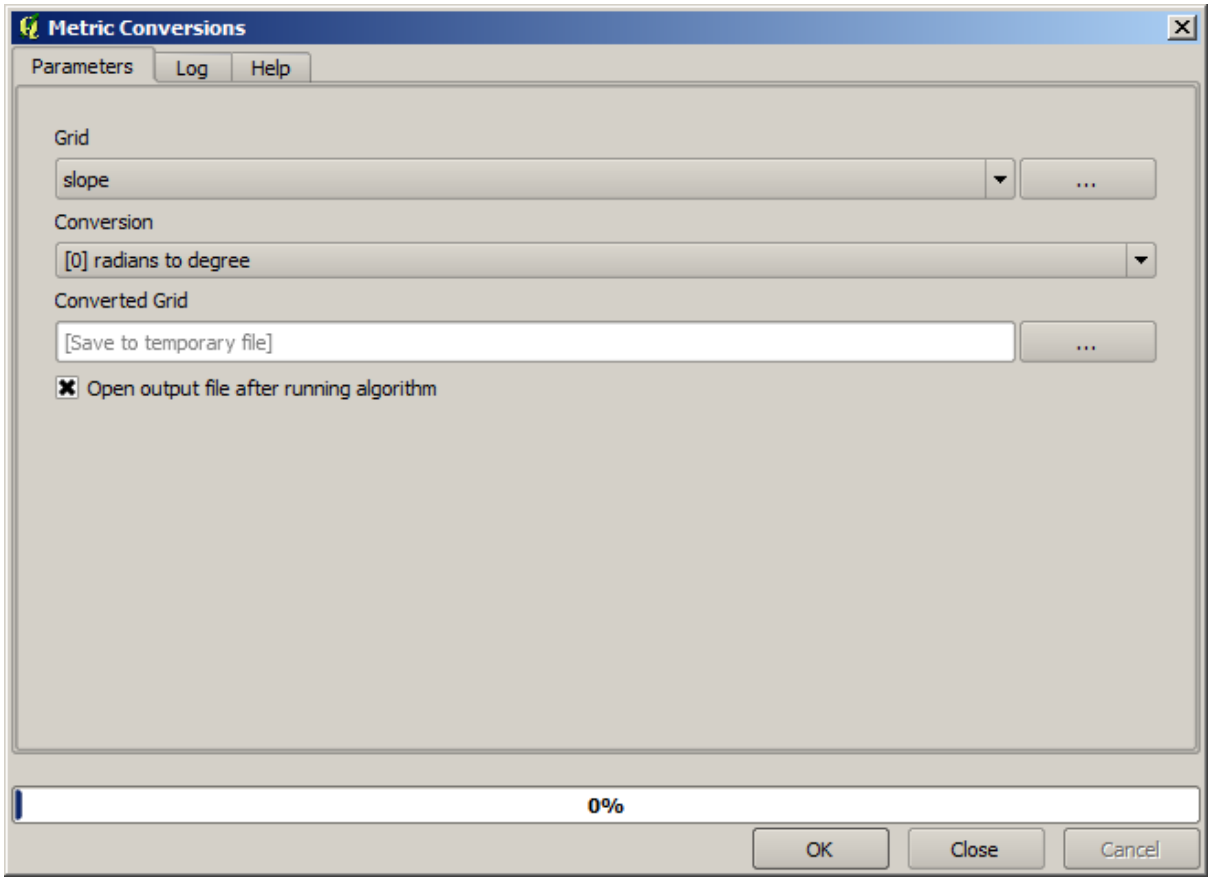
With the new DEM, slope can now be computed.



And here is the resulting slope layer.



The slope produced by the *Slope, Aspect, Curvature* algorithm can be expressed in degrees or radians; degrees are a more practical and common unit. In case you calculated it in radians, the *Metric conversions* algorithm will help us to do the conversion (but in case you didn't know that algorithm existed, you could use the raster calculator that we have already used).



Reprojecting the converted slope layer back with the *Reproject raster layer*, we get the final layer we wanted.

Предупреждение: todo: Add image

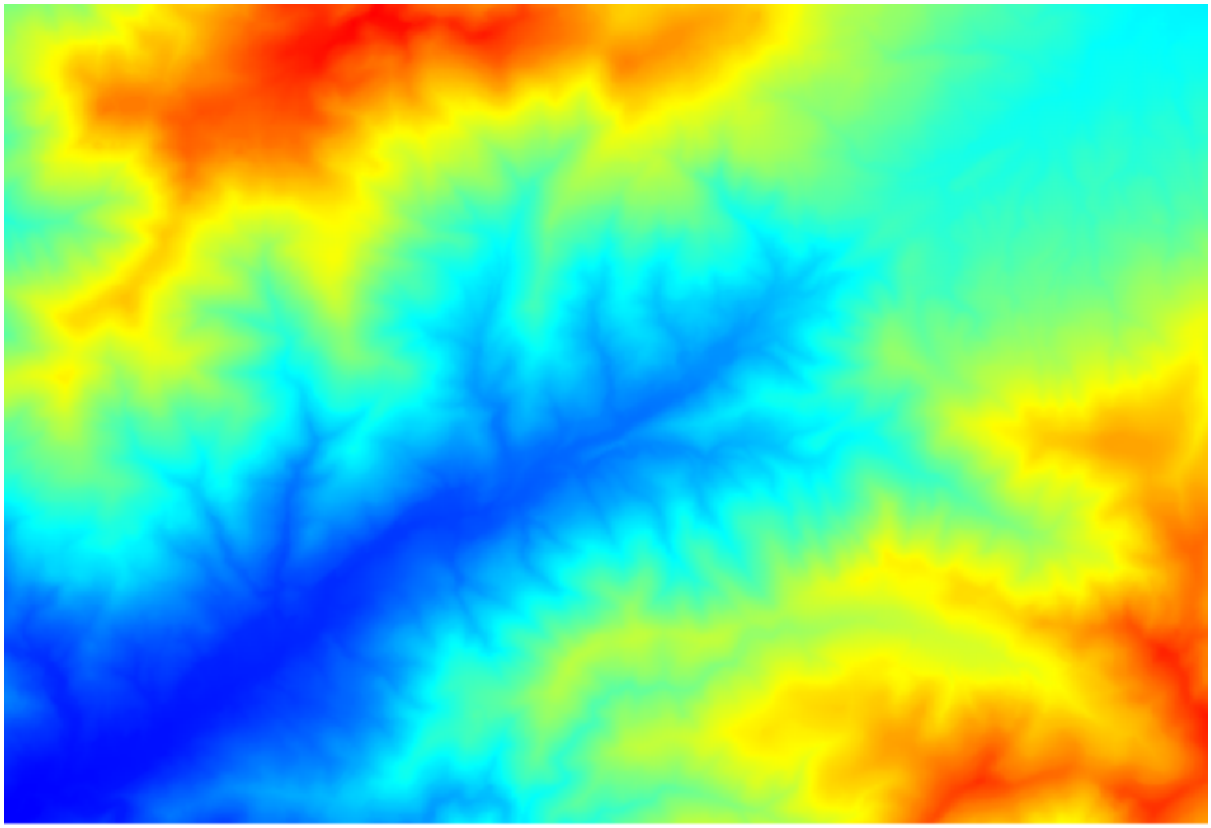
The reprojection processes might have caused the final layer to contain data outside the bounding box that we calculated in one of the first steps. This can be solved by clipping it again, as we did to obtain the base DEM.

17.16 Hydrological analysis

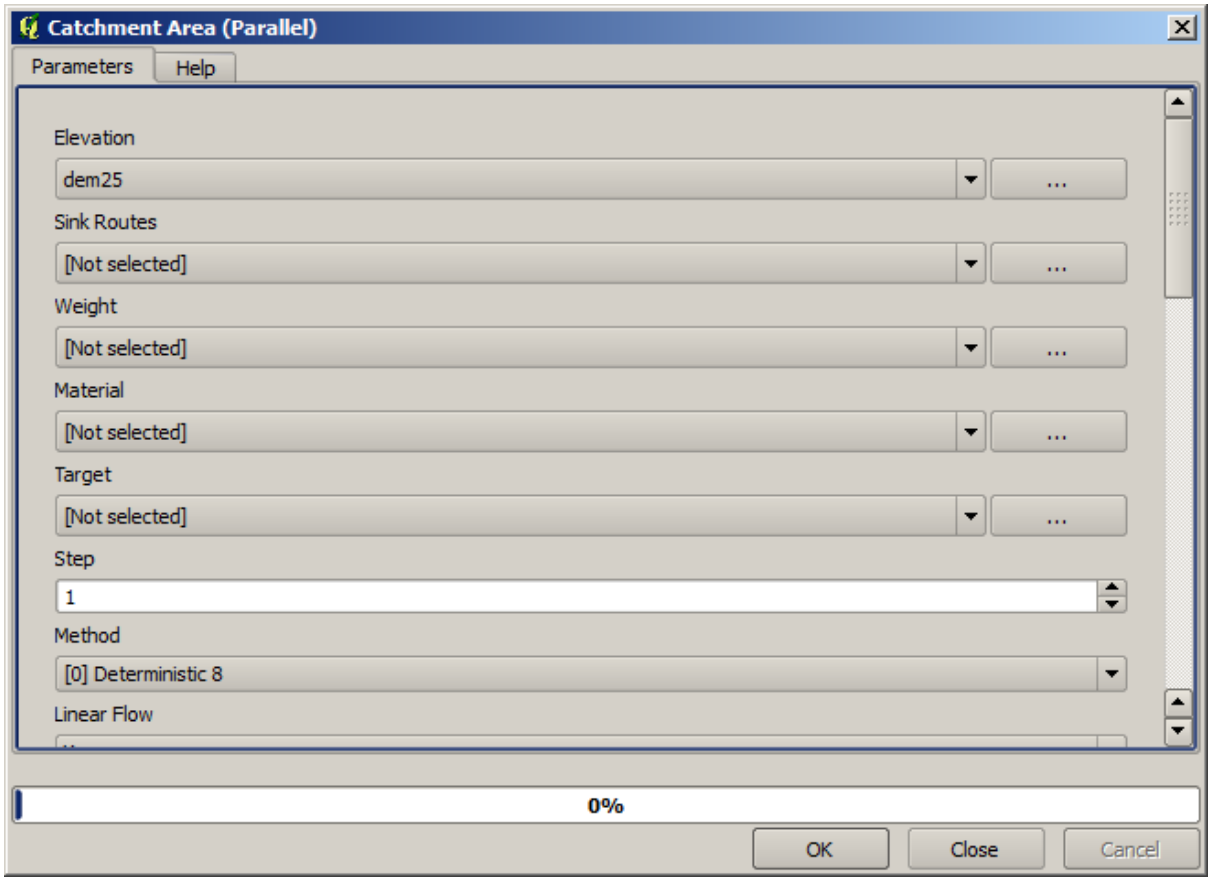
Примечание: In this lesson we will perform some hydrological analysis. This analysis will be used in some of the following lessons, as it constitutes a very good example of an analysis workflow, and we will use it to demonstrate some advanced features.

Objectives: Starting with a DEM, we are going to extract a channel network, delineate watersheds and calculate some statistics.

1. The first thing is to load the project with the lesson data, which just contains a DEM.

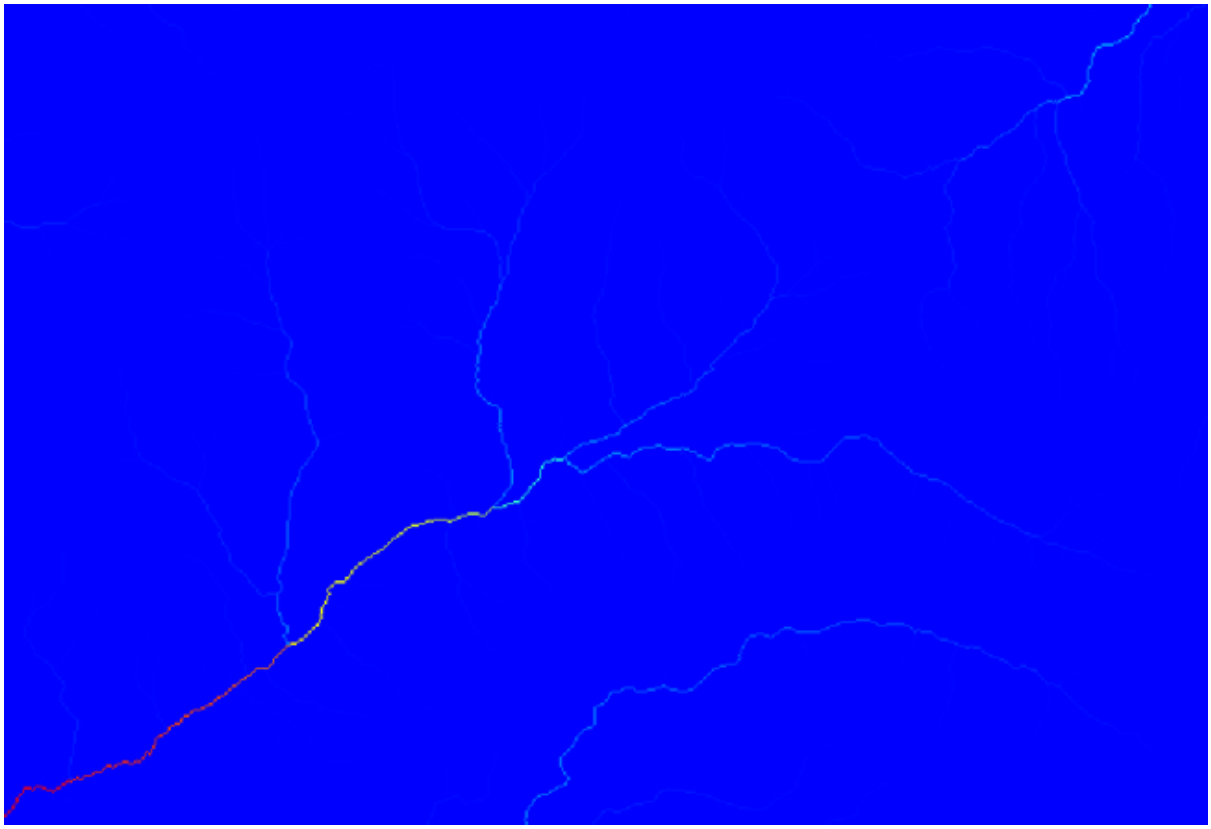


2. The first module to execute is *Catchment area* (in some SAGA versions it is called *Flow accumulation (Top Down)*). You can use any of the others named *Catchment area*. They have different algorithms underneath, but the results are basically the same.
3. Select the DEM in the *Elevation* field, and leave the default values for the rest of the parameters.

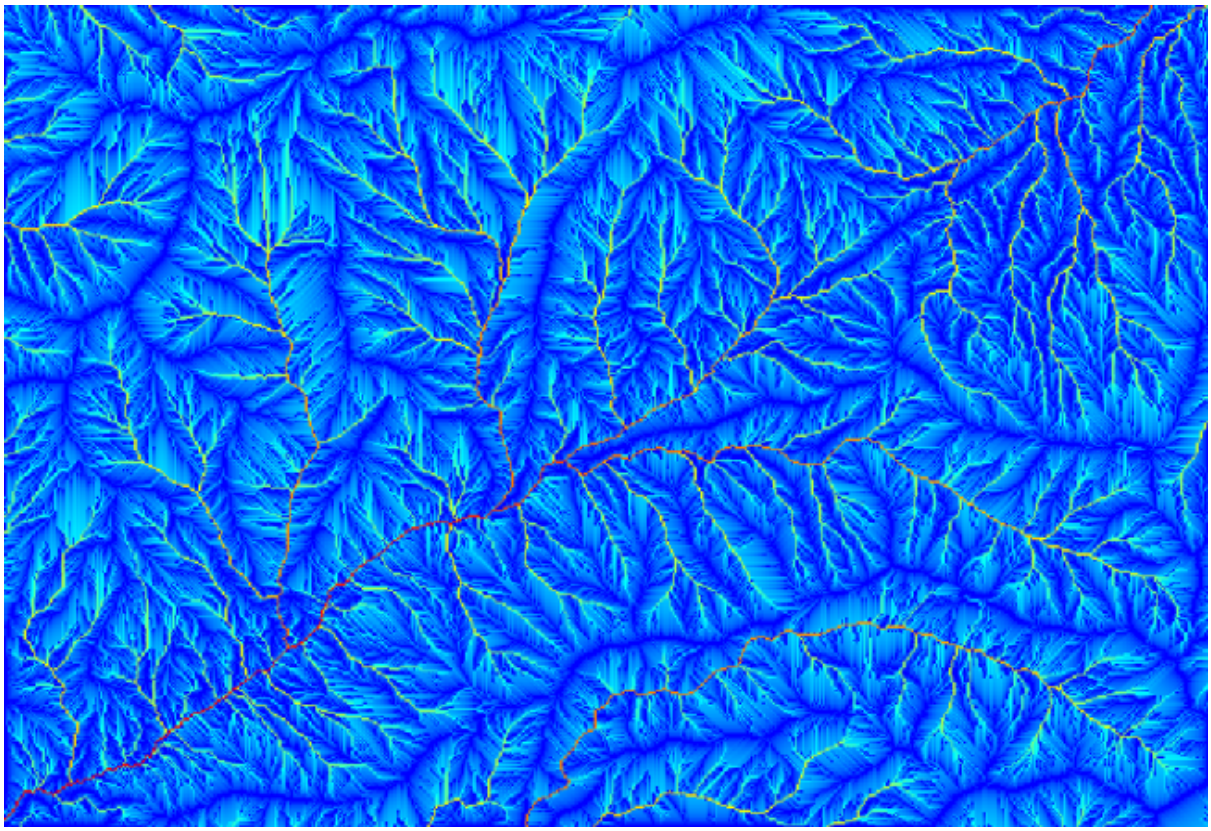


Some algorithms calculate many layers, but the *Catchment Area* layer is the only one we will be using. You can get rid of the other ones if you want.

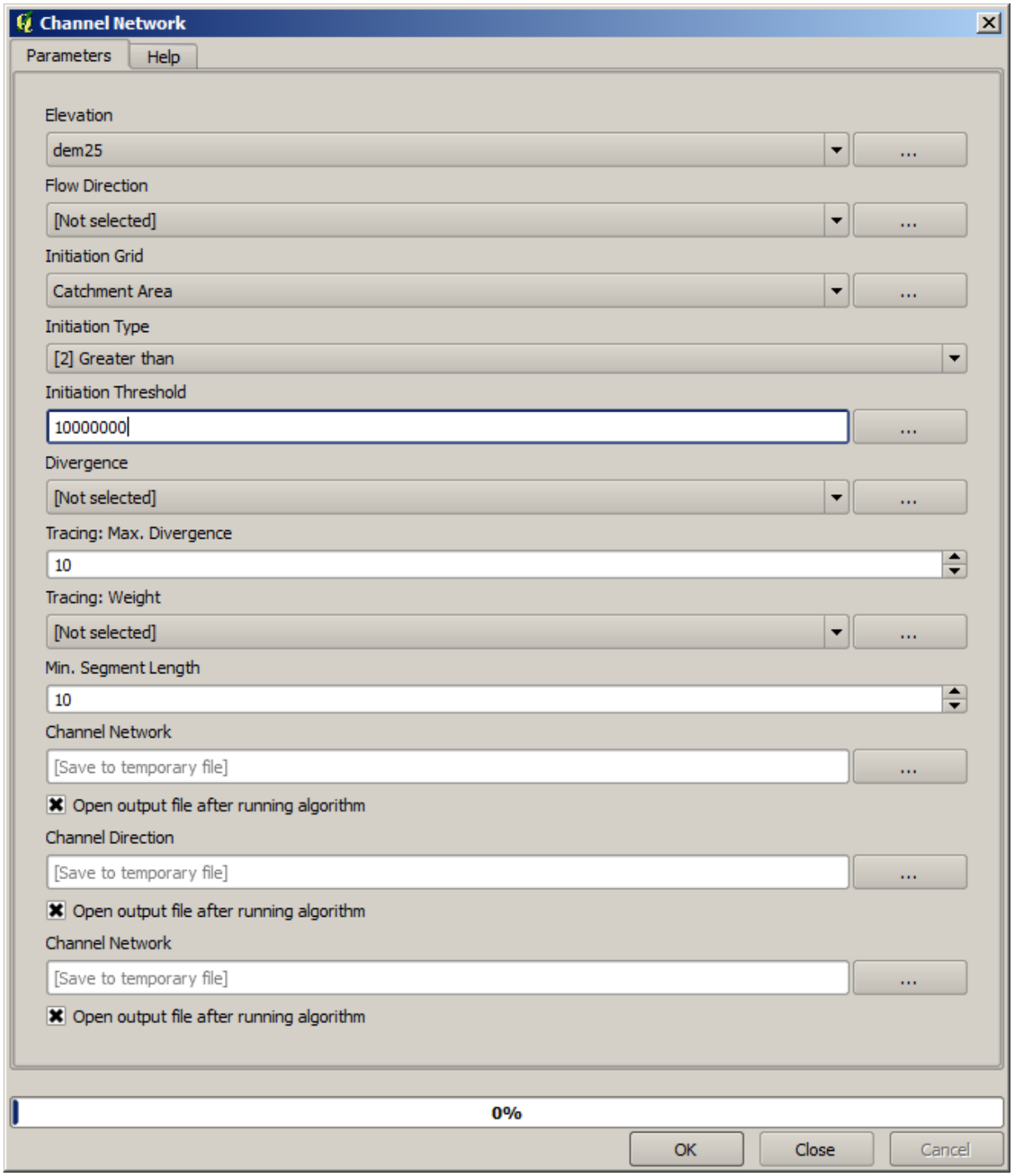
The rendering of the layer is not very informative.



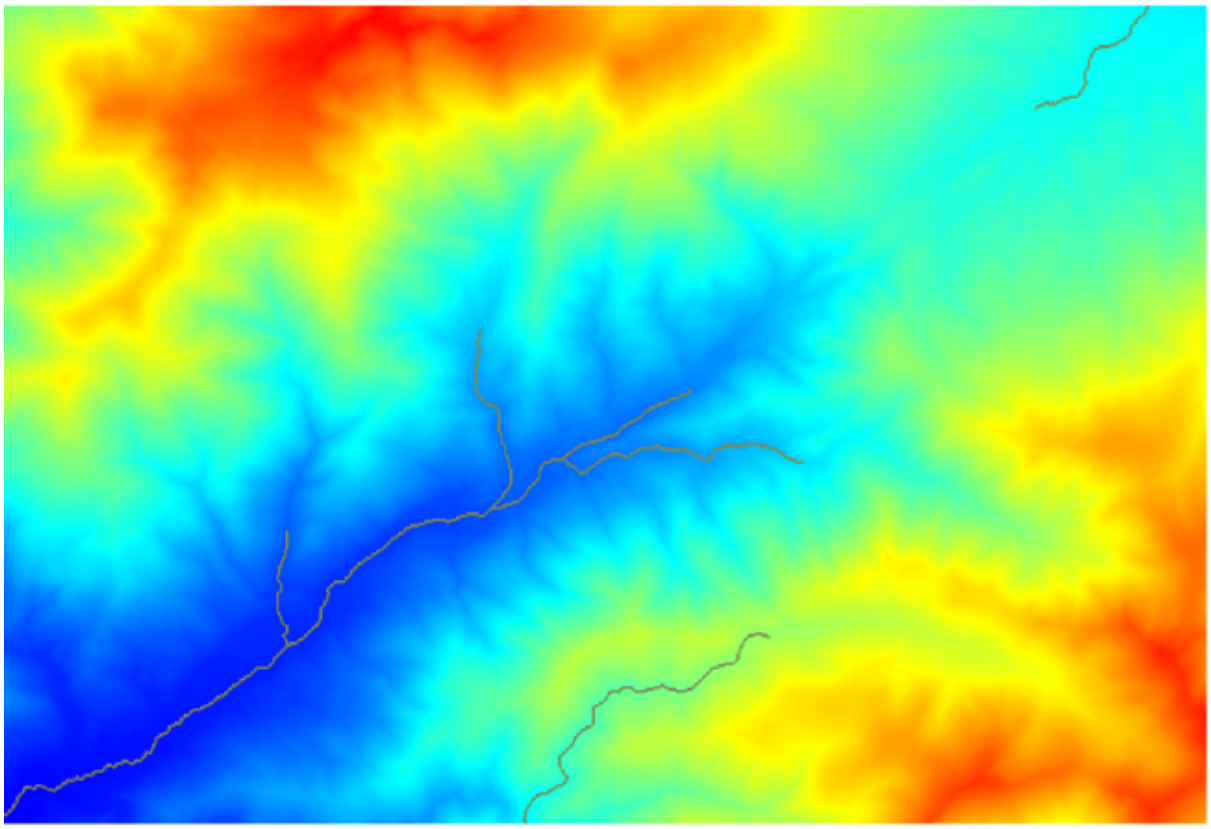
To know why, you can have a look at the histogram and you will see that values are not evenly distributed (there are a few cells with very high value, those corresponding to the channel network). Use the *Raster calculator* algorithm to calculate the logarithm of the catchment value area and you will get a layer with much more information



4. The catchment area (also known as flow accumulation) can be used to set a threshold for channel initiation. This can be done using the *Channel network* algorithm.
   - *Initiation grid*: use the catchment area layer and not the logarithm one.
   - *Initiation threshold*: `10.000.000`
   - *Initiation type*: *Greater than*

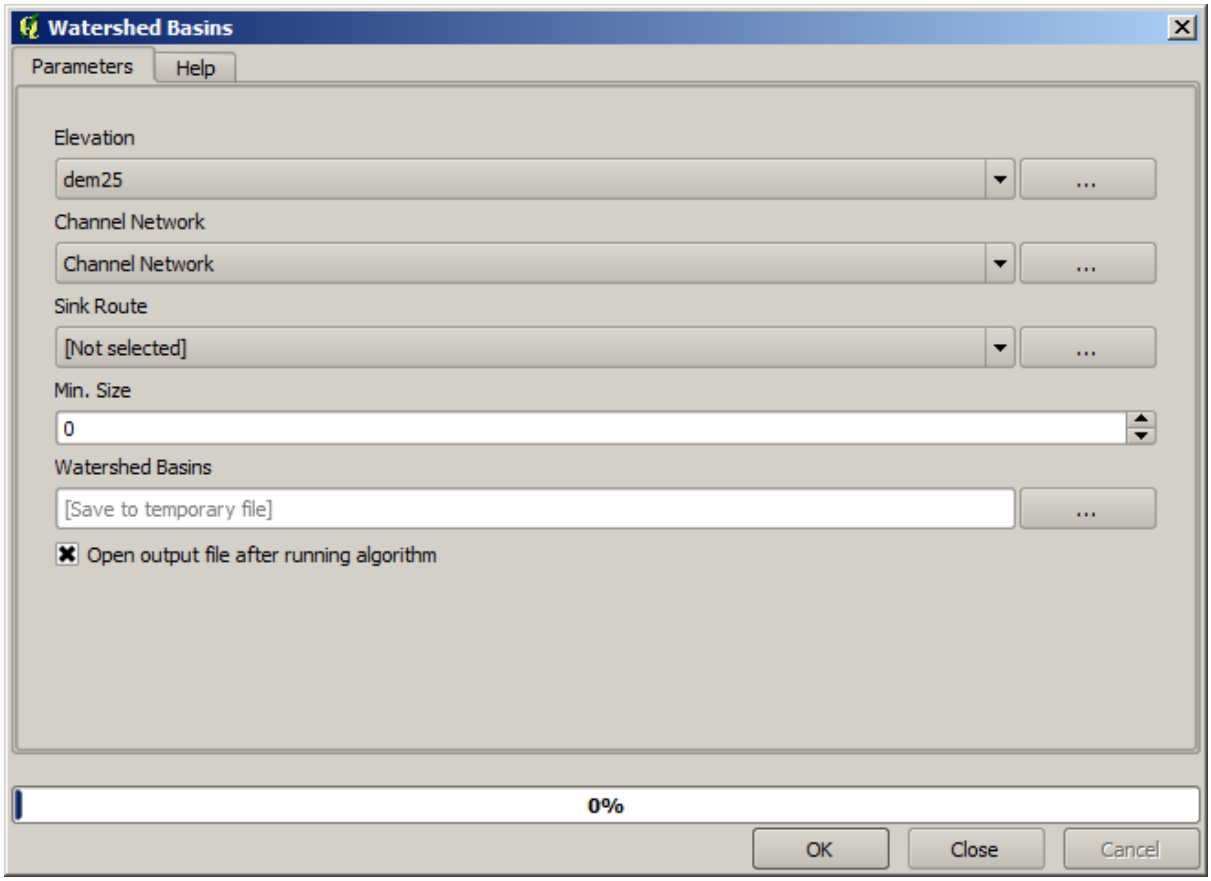


If you increase the *Initiation threshold* value, you will get a more sparse channel network. If you decrease it, you will get a denser one. With the proposed value, this is what you get.

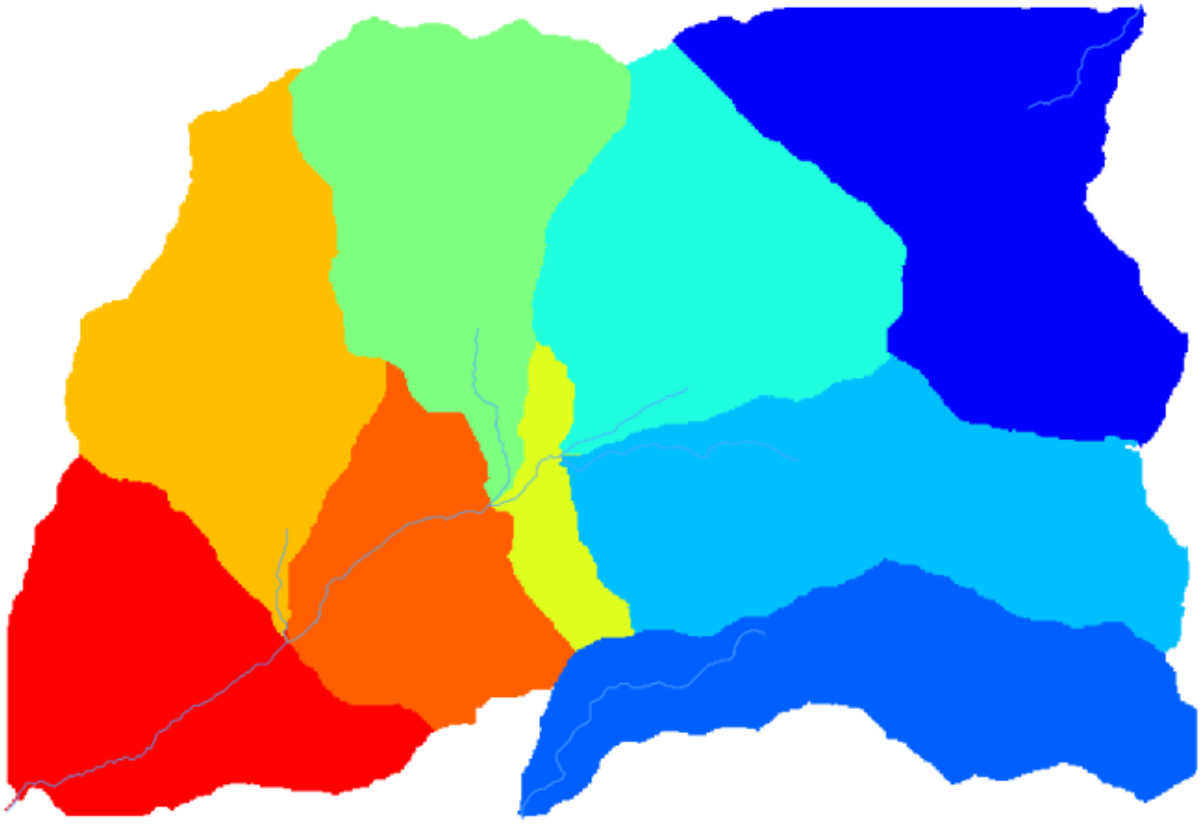


The image above shows just the resulting vector layer and the DEM, but there should be also a raster layer with the same channel network. That raster will be, in fact, the layer we will be using.

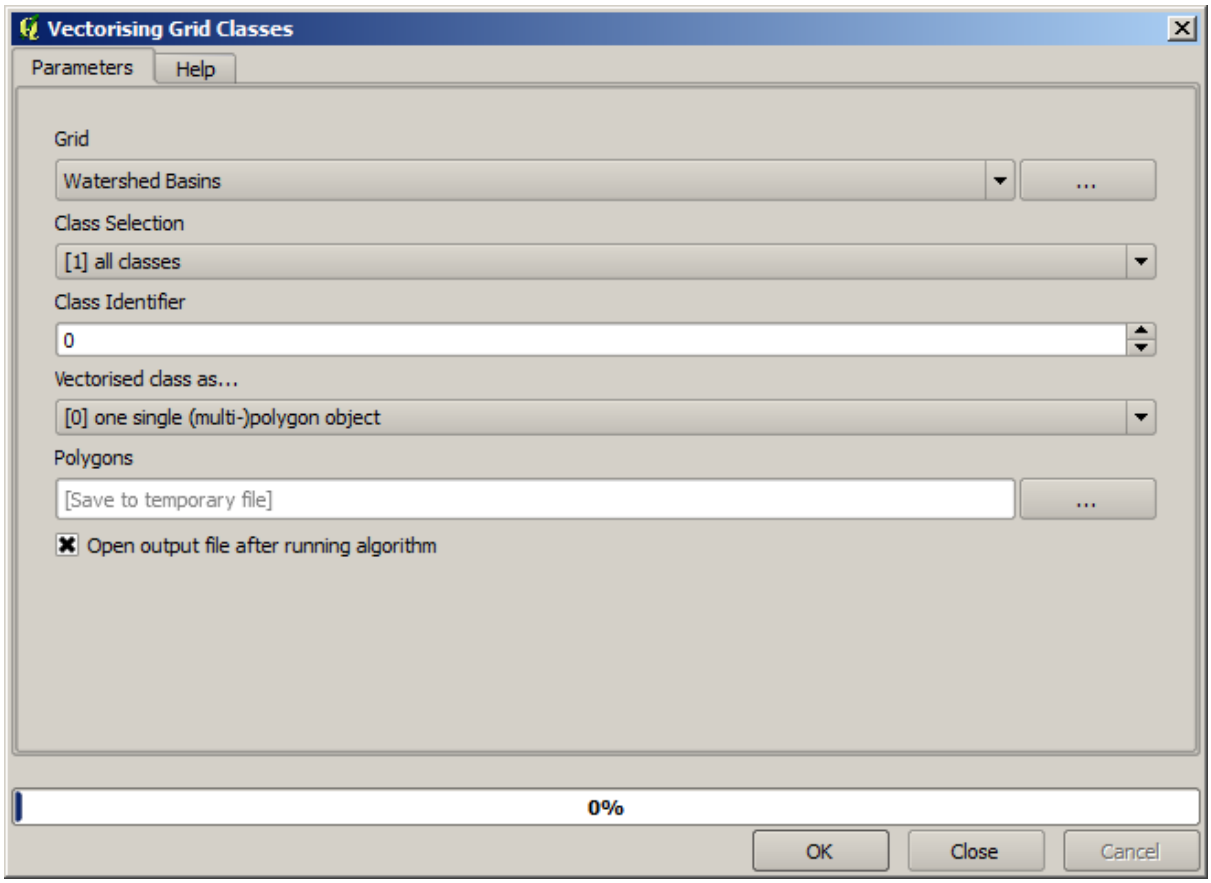
5. Now, we will use the *Watersheds basins* algorithm to delineate the subbasins corresponding to that channel network, using as outlet points all the junctions in it. Here is how you have to set the corresponding parameters dialog.

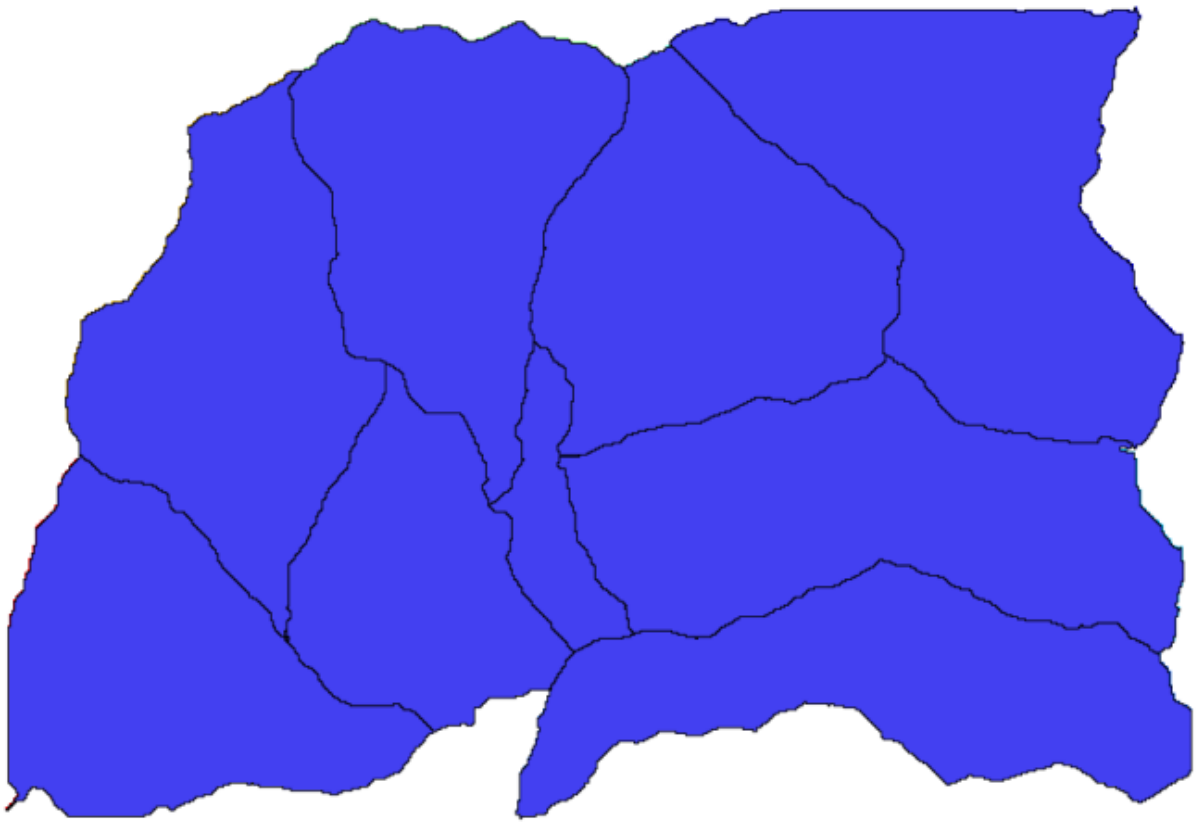


And this is what you will get.



6. This is a raster result. You can vectorise it using the *Vectorising grid classes* algorithm.

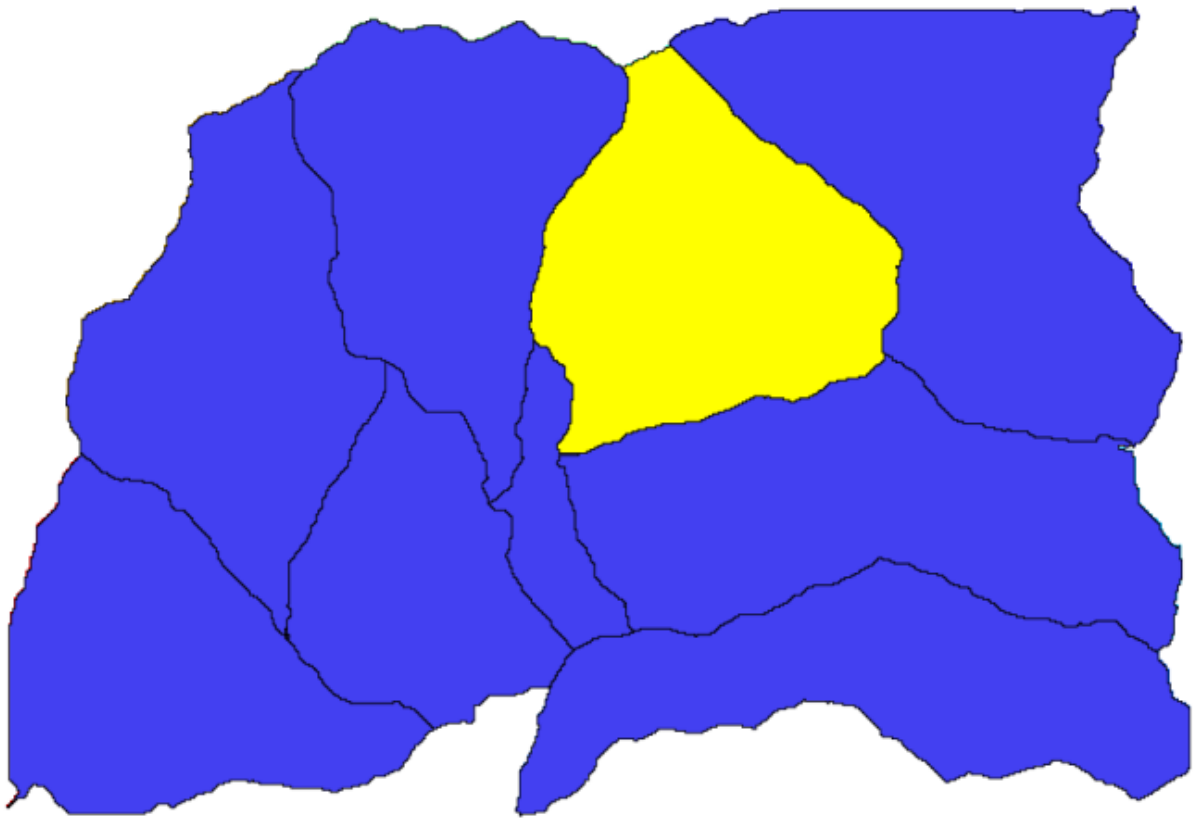




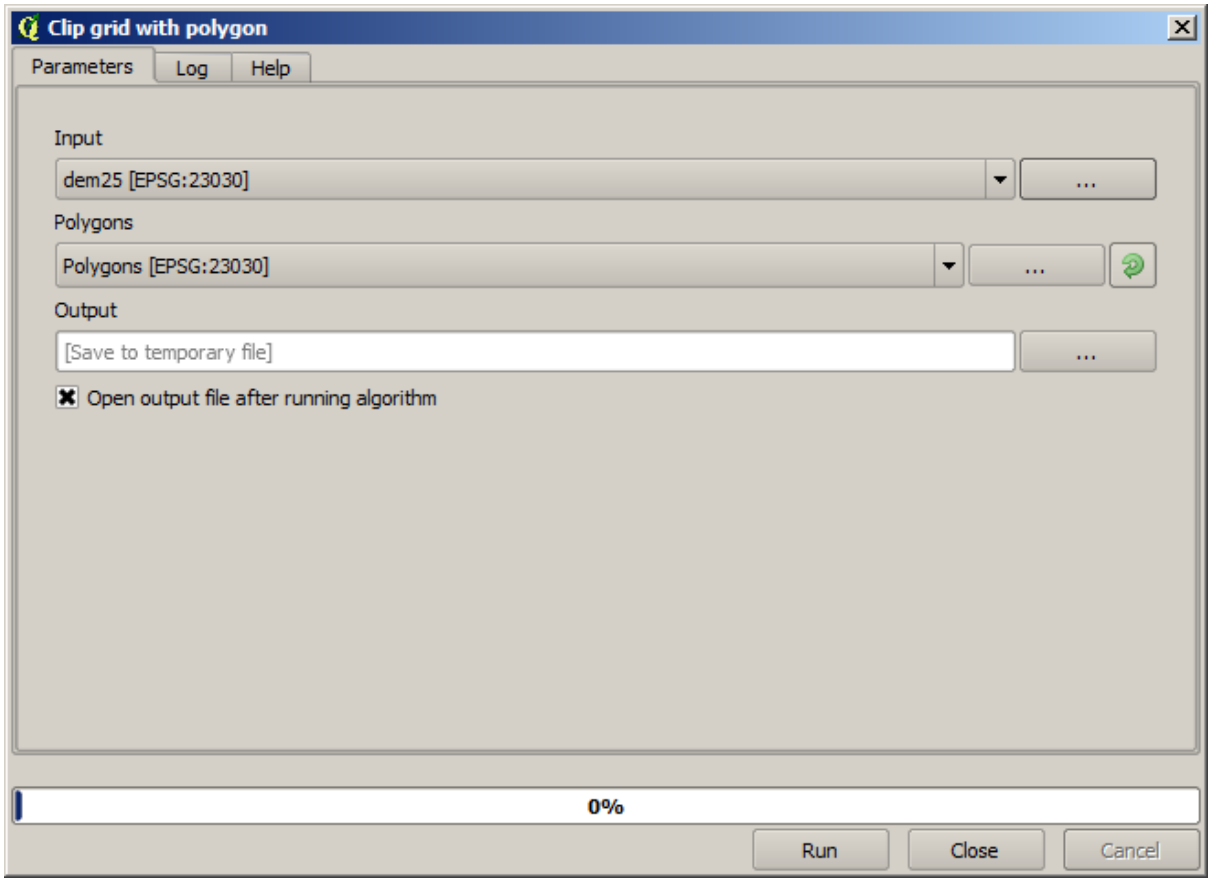
Now, let's try to compute statistics about the elevation values in one of the subbasins. The idea is to have a layer that just represents the elevation within that subbasin and then pass it to the module that calculates those statistics.

1. First, let's clip the original DEM with the polygon representing a subbasin. We will use the *Clip raster with polygon* algorithm. If we select a single subbasin polygon and then call the clipping algorithm, we can clip the DEM to the area covered by that polygon, since the algorithm is aware of the selection.

    1. Select a polygon

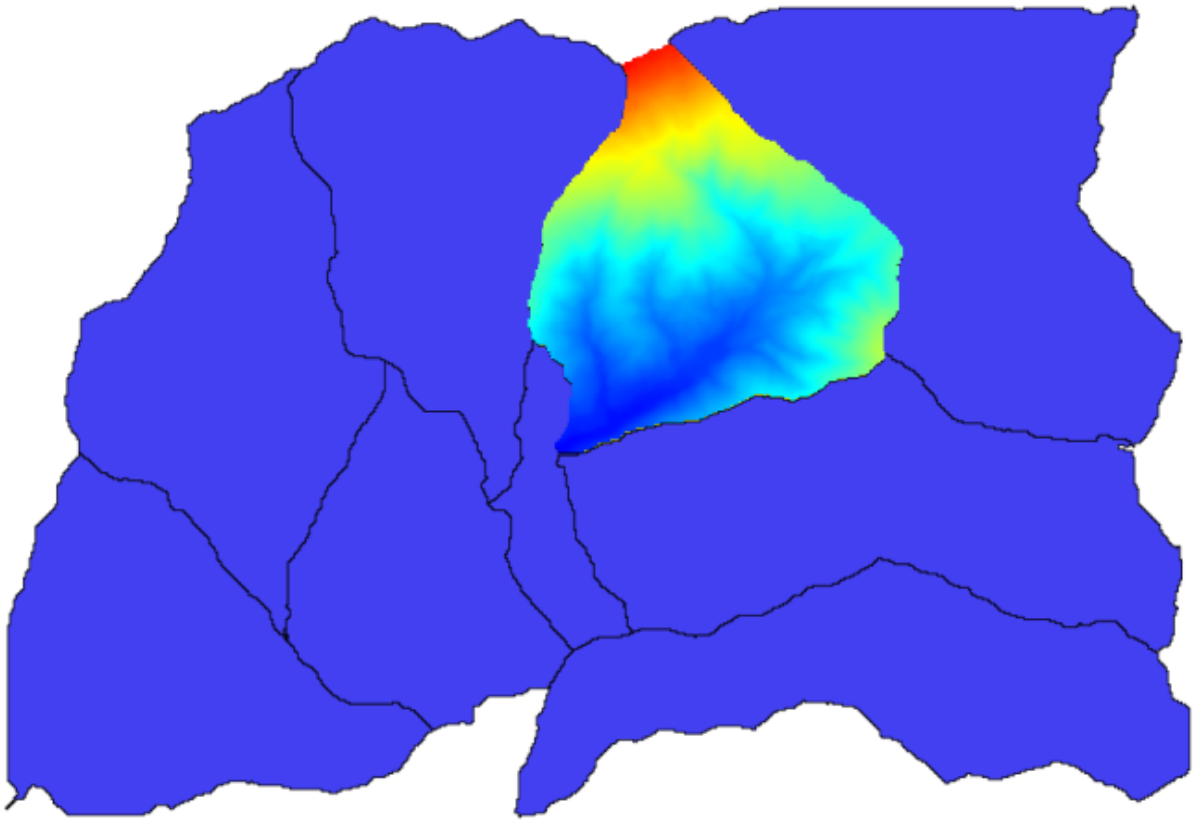


2. Call the clipping algorithm with the following parameters:

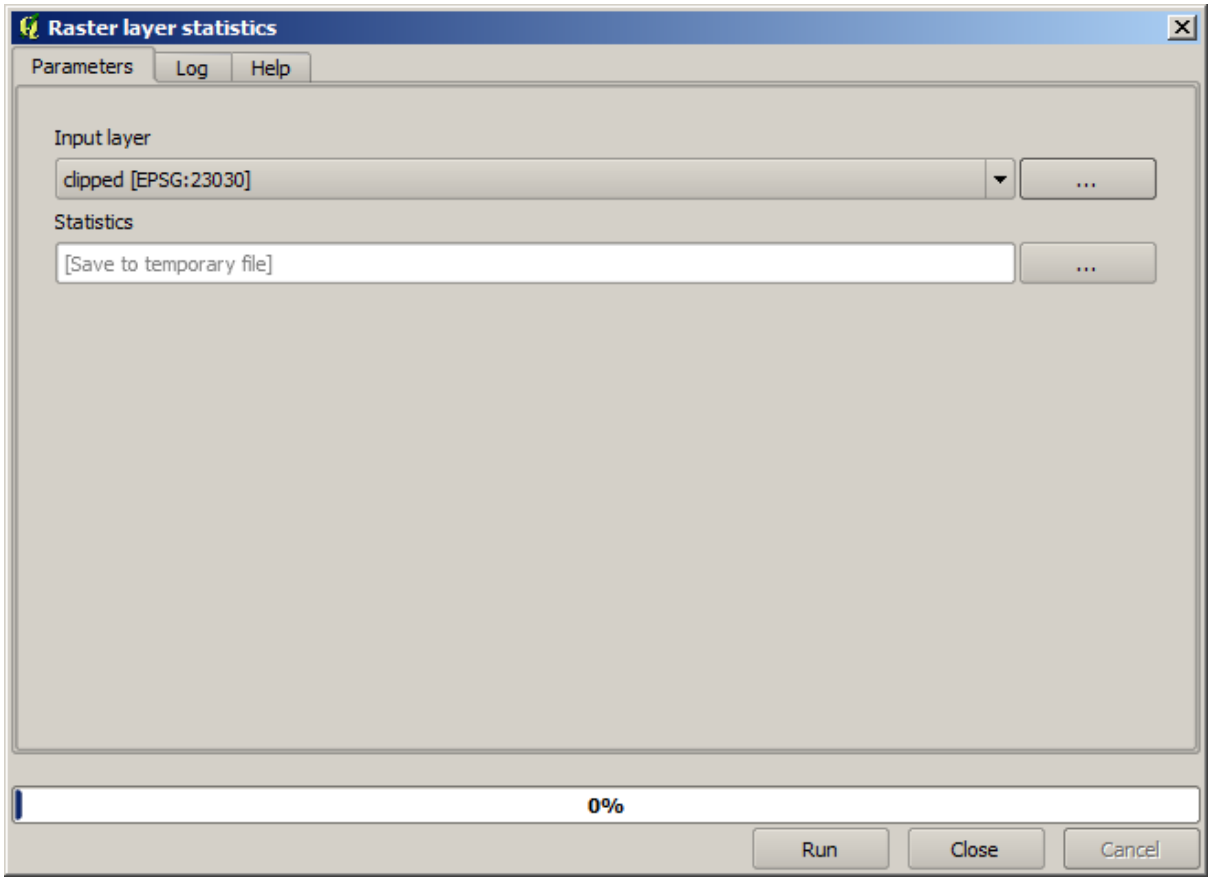


The element selected in the input field is, of course, the DEM we want to clip.

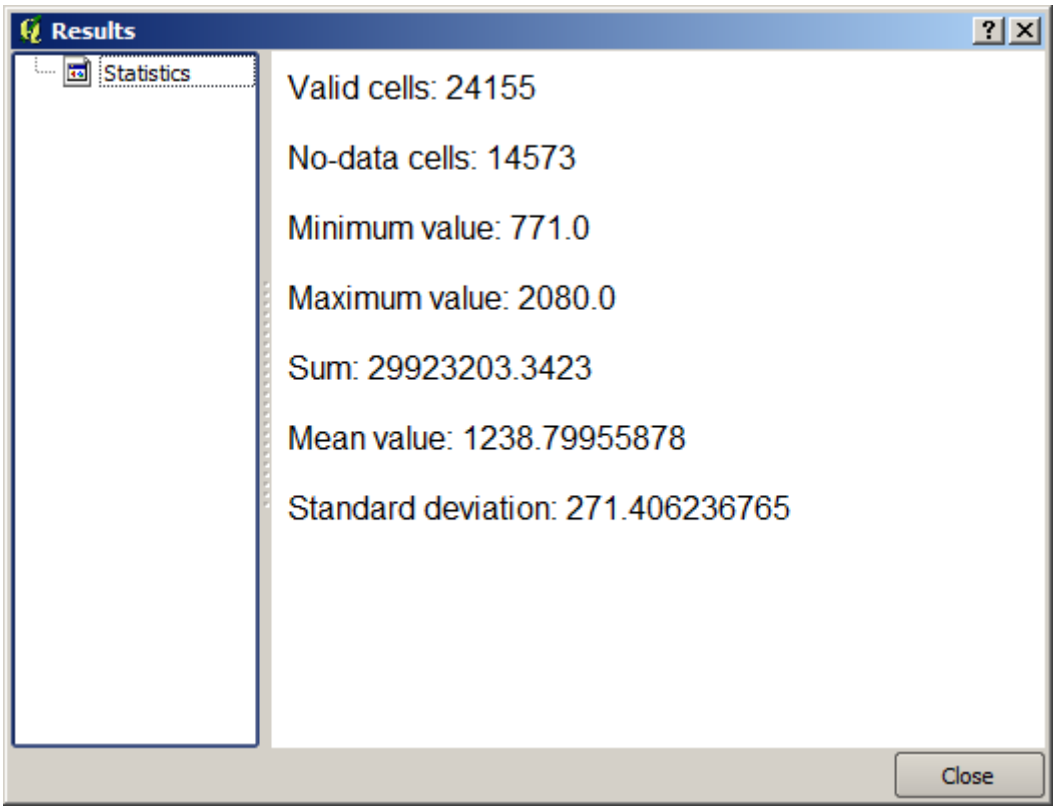
You will get something like this.



2. This layer is ready to be used in the *Raster layer statistics* algorithm.



The resulting statistics are the following ones.



We will use both the basin calculations procedure and the statistics calculation in other lessons, to find out how

other elements can help us automate both of them and work more effectively.

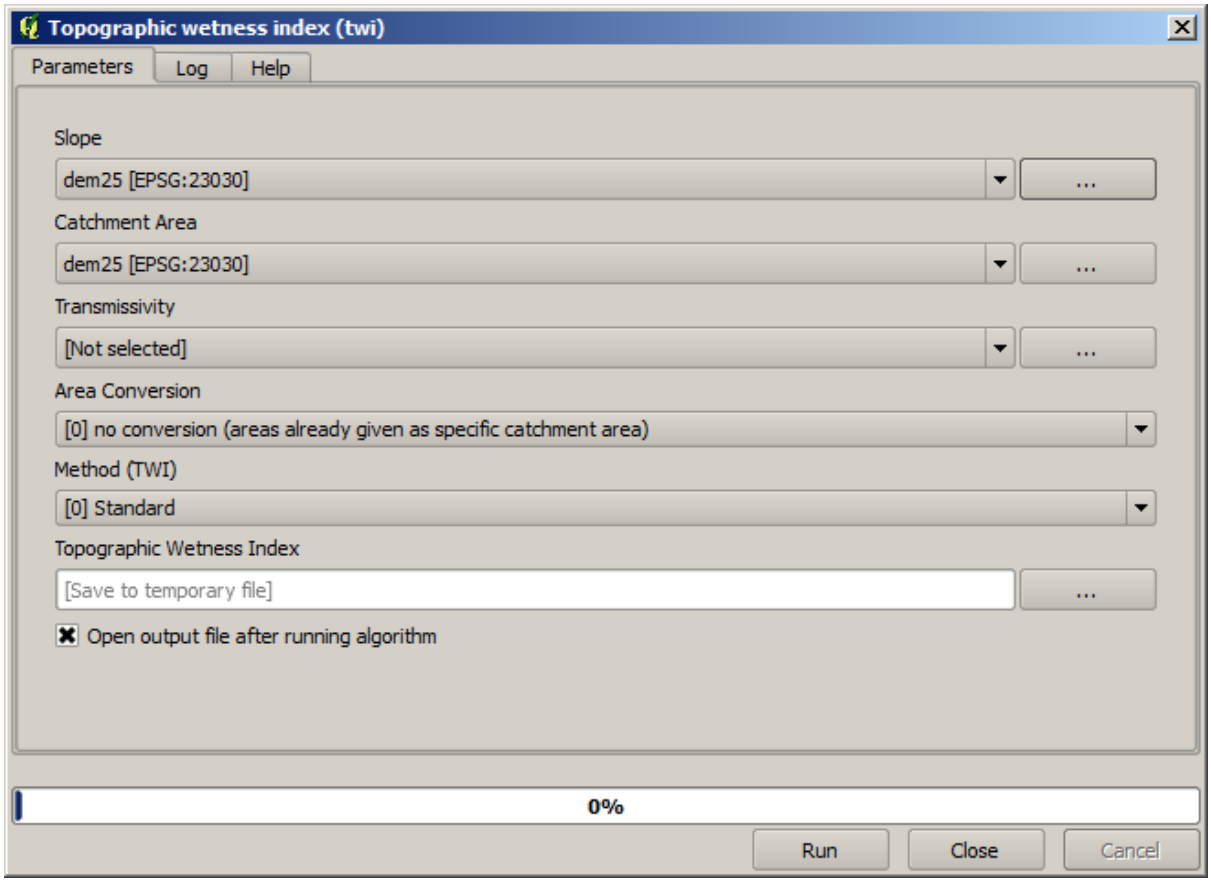
17.17 Starting with the graphical modeler

Примечание: In this lesson we will use the graphical modeler, a powerful component that we can use to define a workflow and run a chain of algorithms.

A normal session with the processing tools includes more than running a single algorithm. Usually several of them are run to obtain a result, and the outputs of some of those algorithms are used as input for some of the other ones.

Using the graphical modeler, that workflow can be put into a model, which will run all the necessary algorithms in a single run, thus simplifying the whole process and automating it.

To start this lesson, we are going to calculate a parameter named Topographic Wetness Index. The algorithm that computes it is called *Topographic wetness index (twi)*.

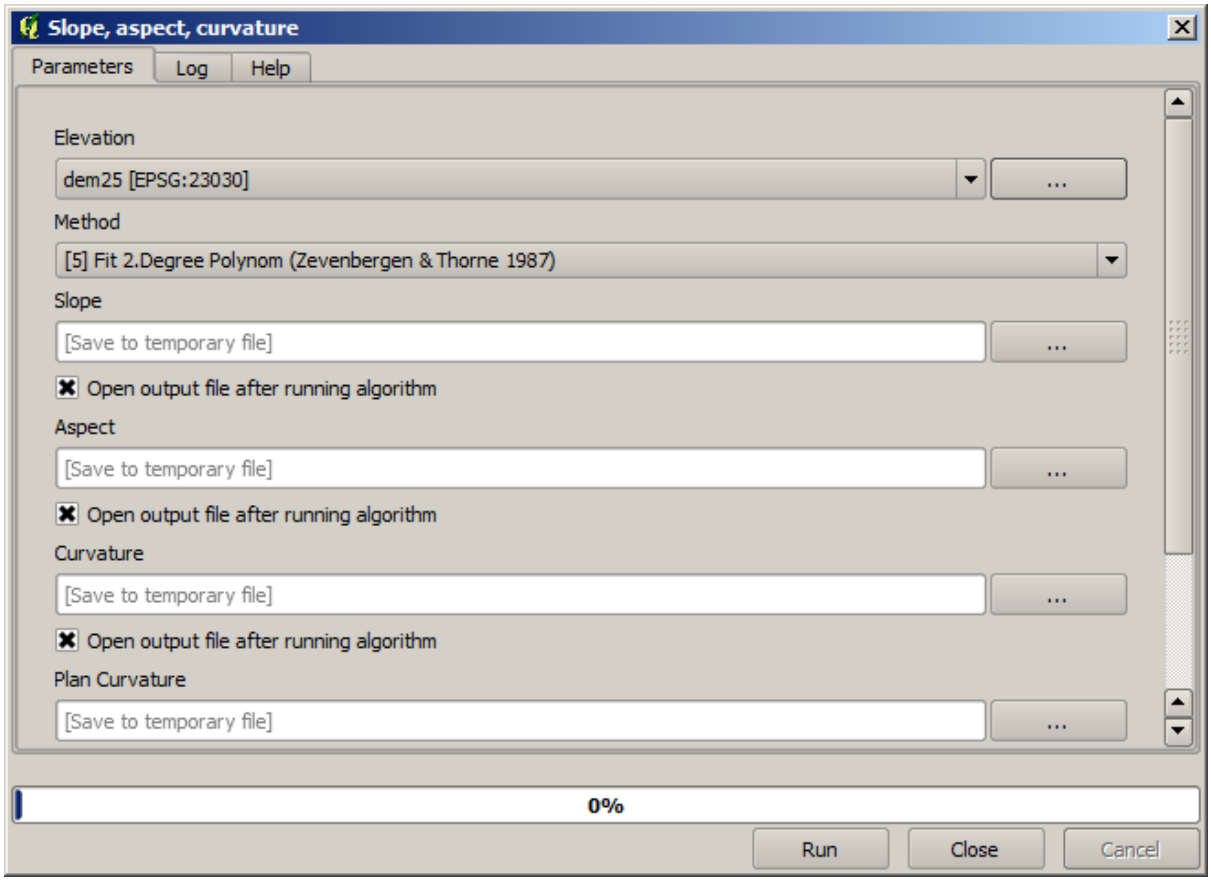


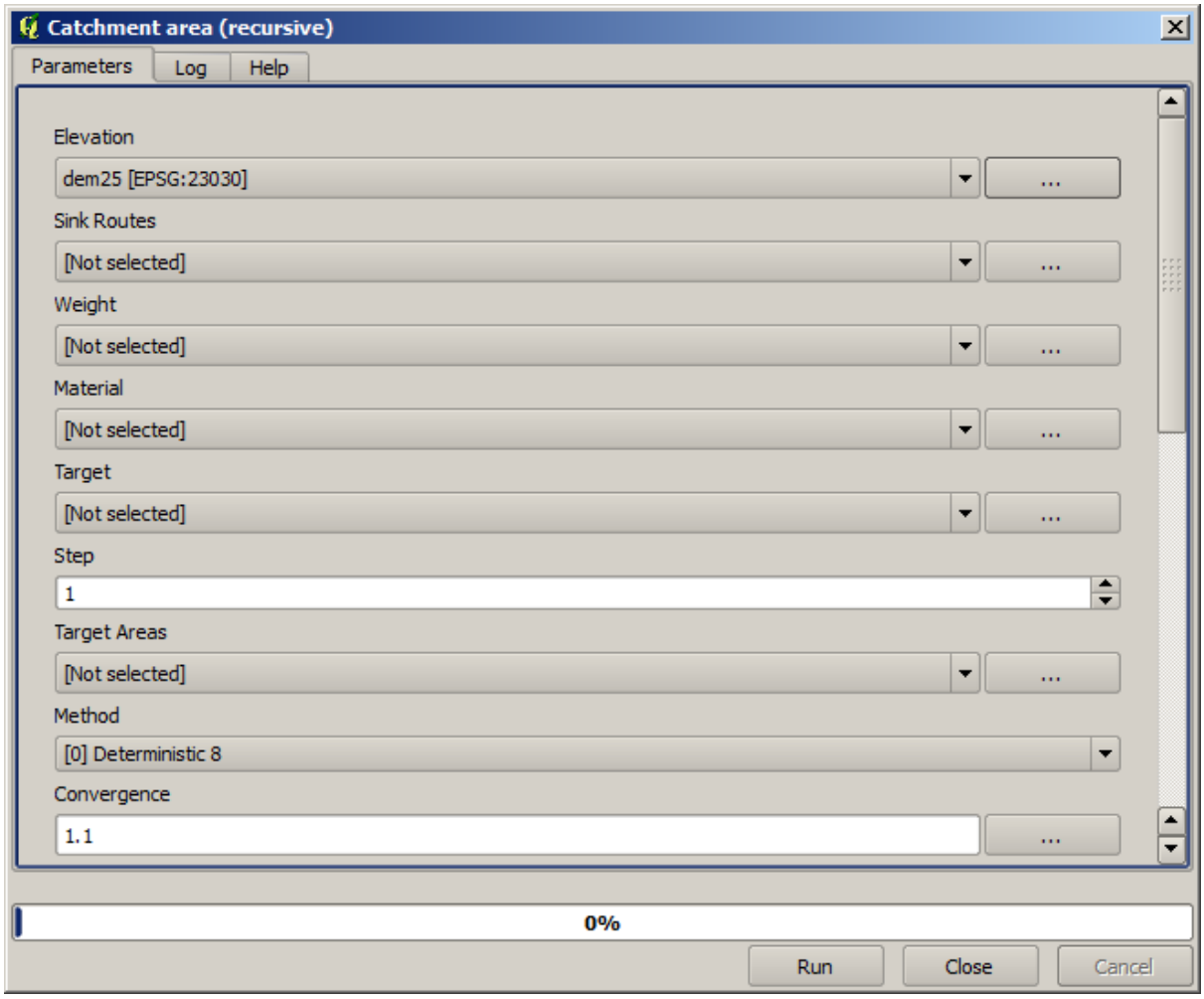
As you can see, there are two mandatory inputs: *Slope* and *Catchment area*. There is also an optional input, but we will not be using it, so we can ignore it.

The data for this lesson contains just a DEM, so we do not have any of the required inputs. However, we know how to calculate both of them from that DEM, since we have already seen the algorithms to compute slope and catchment area. So we can first compute those layers and then use them for the TWI algorithm.

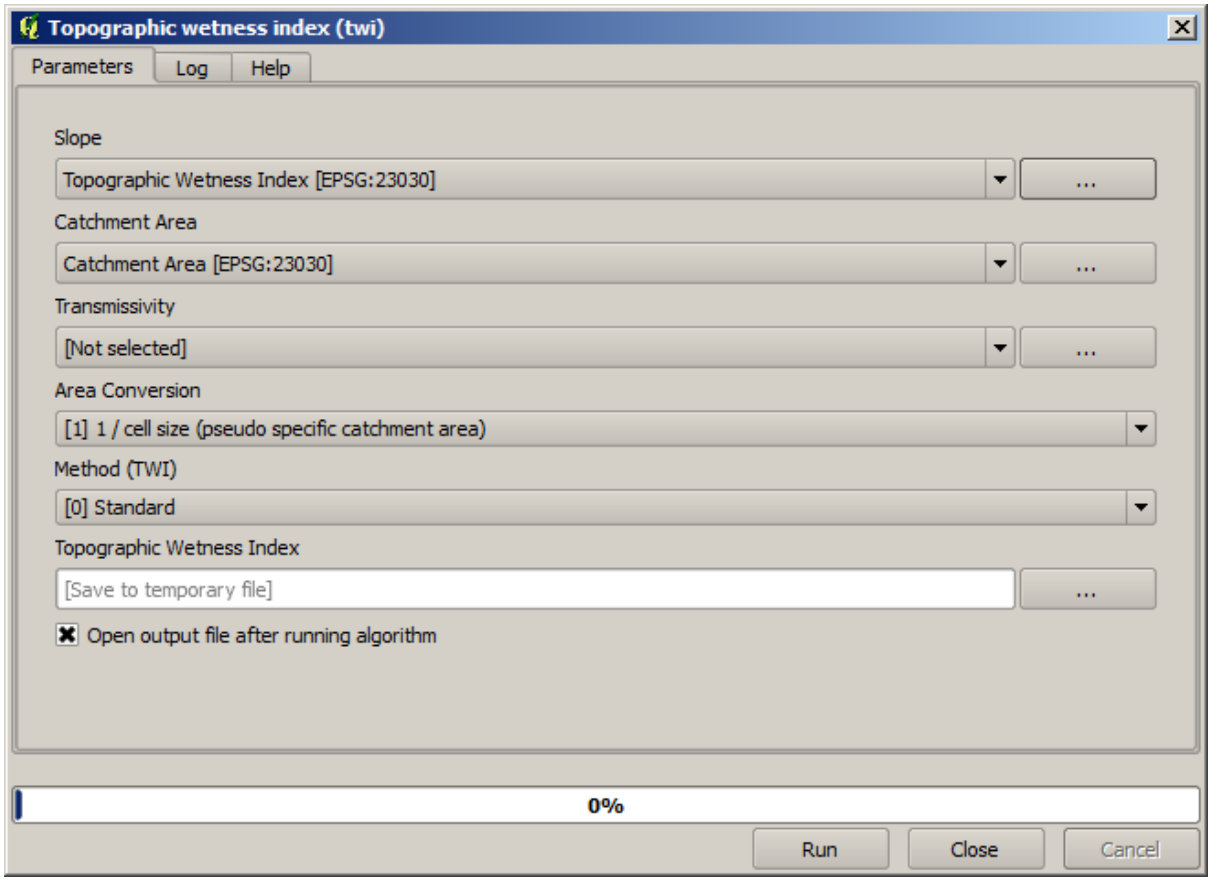
Here are the parameter dialogs that you should use to calculate the 2 intermediate layers.

Примечание: Slope must be calculated in radians, not in degrees.

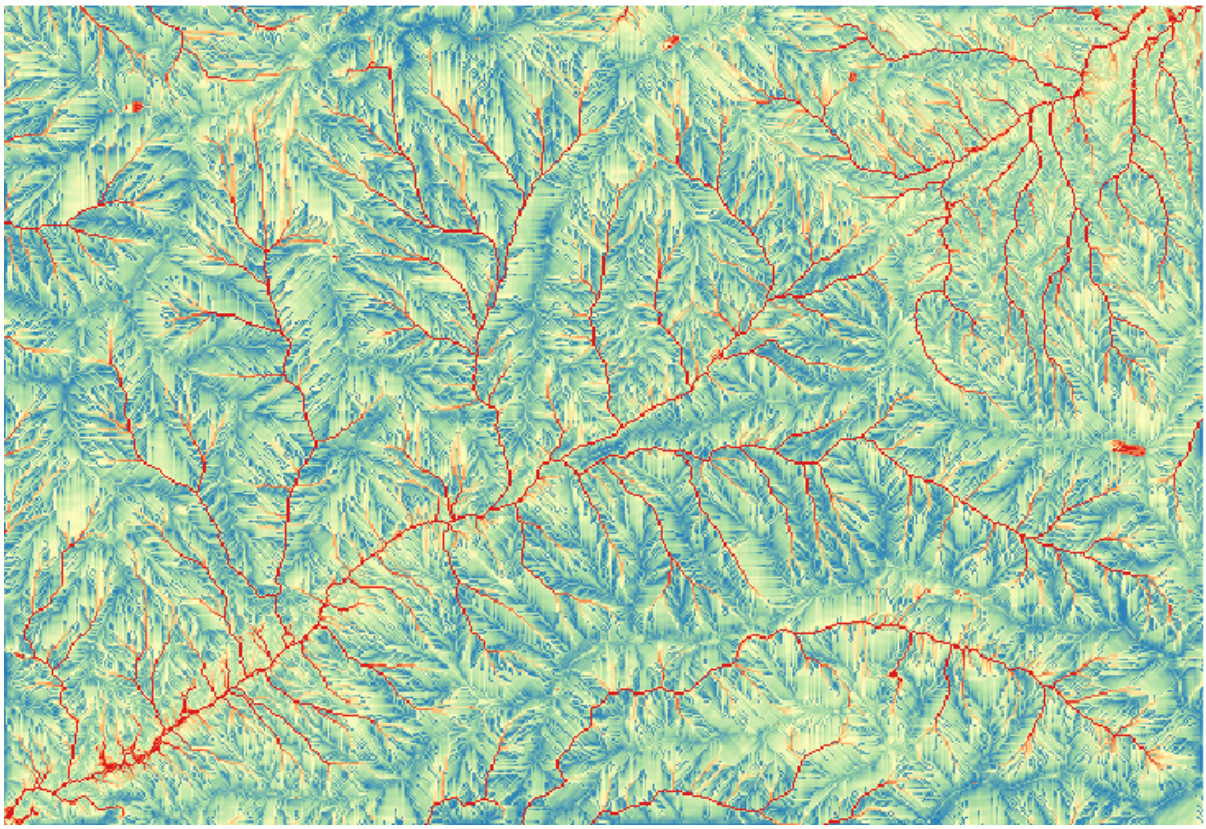

Slope, aspect, curvature
Parameters
Log
Help
Elevation
dem25 [EPSG:23030]
...
Method
[5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
Slope
[Save to temporary file]
...
Open output file after running algorithm
Aspect
[Save to temporary file]
...
Open output file after running algorithm
Curvature
[Save to temporary file]
...
Open output file after running algorithm
Plan Curvature
[Save to temporary file]
...
0%
Run
Close
Cancel



And this is how you have to set the parameters dialog of the TWI algorithm.

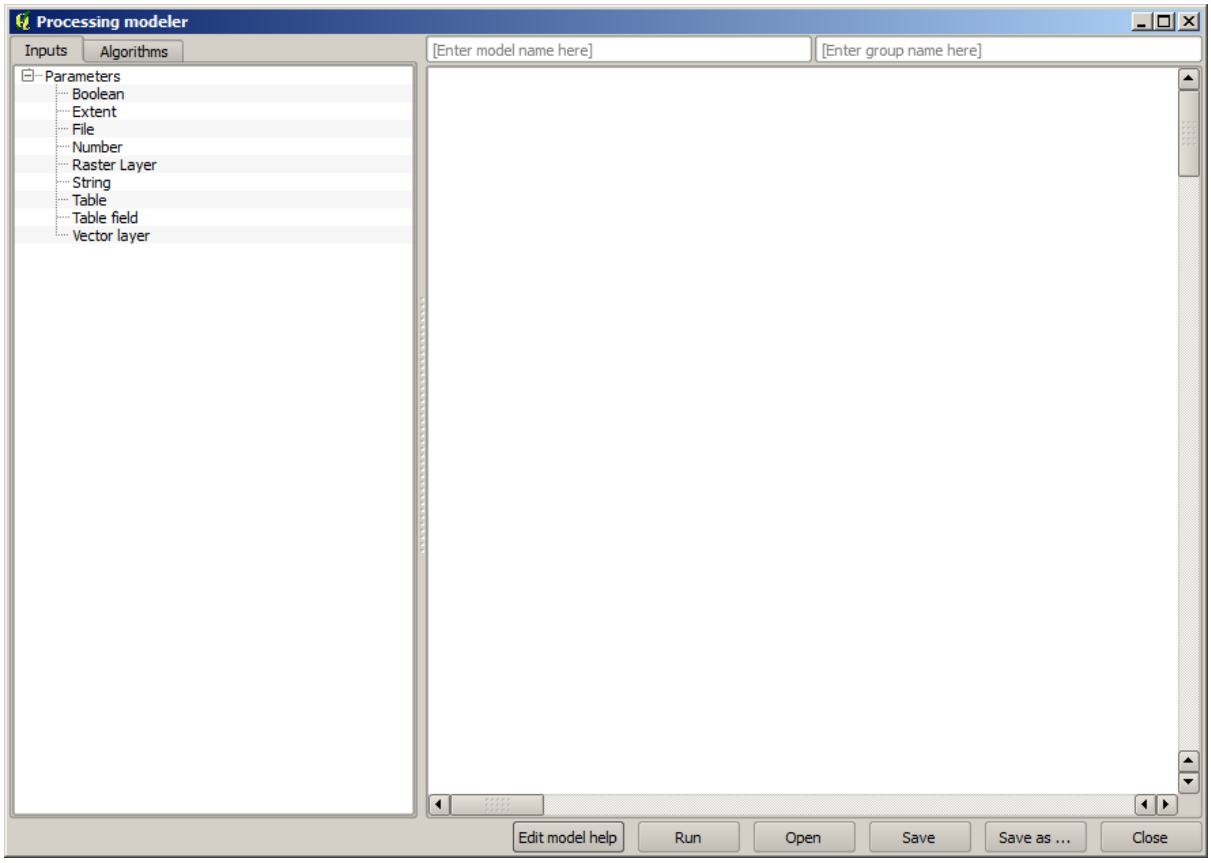


This is the result that you will obtain (the default singleband pseudocolor inverted palette has been used for rendering). You can use the `twi.qml` style provided.



What we will try to do now is to create an algorithm that calculates the TWI from a DEM in just one single step. That will save us work in case we later have to compute a TWI layer from another DEM, since we will need just one single step to do it instead of the three above. All the processes that we need are found in the toolbox, so what we have to do is to define the workflow to wrap them. This is where the graphical modeler comes in.

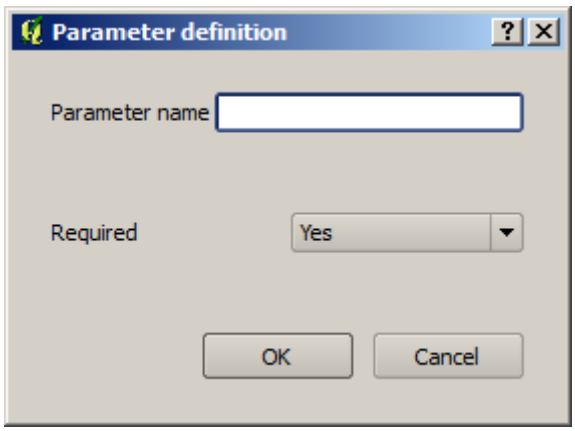
1. Open the modeler by selecting its menu entry in the processing menu.



   Two things are needed to create a model: setting the inputs that it will need, and defining the algorithm that it contains. Both of them are done by adding elements from the two tabs in the left-hand side of the modeler window: *Inputs* and *Algorithms*.

2. Let's start with the inputs. In this case we do not have much to add. We just need a raster layer with the DEM, and that will be our only input data.

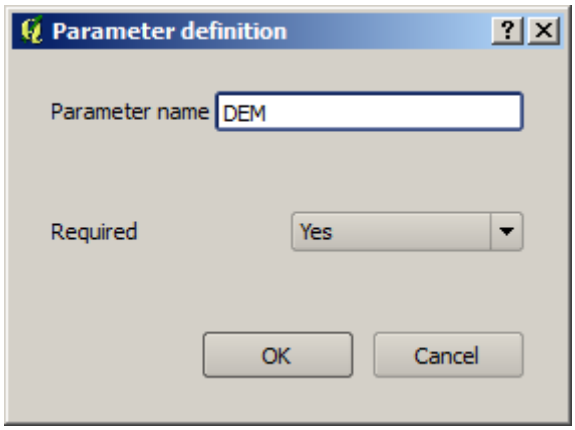
3. Double click on the *Raster layer* input and you will see the following dialog.



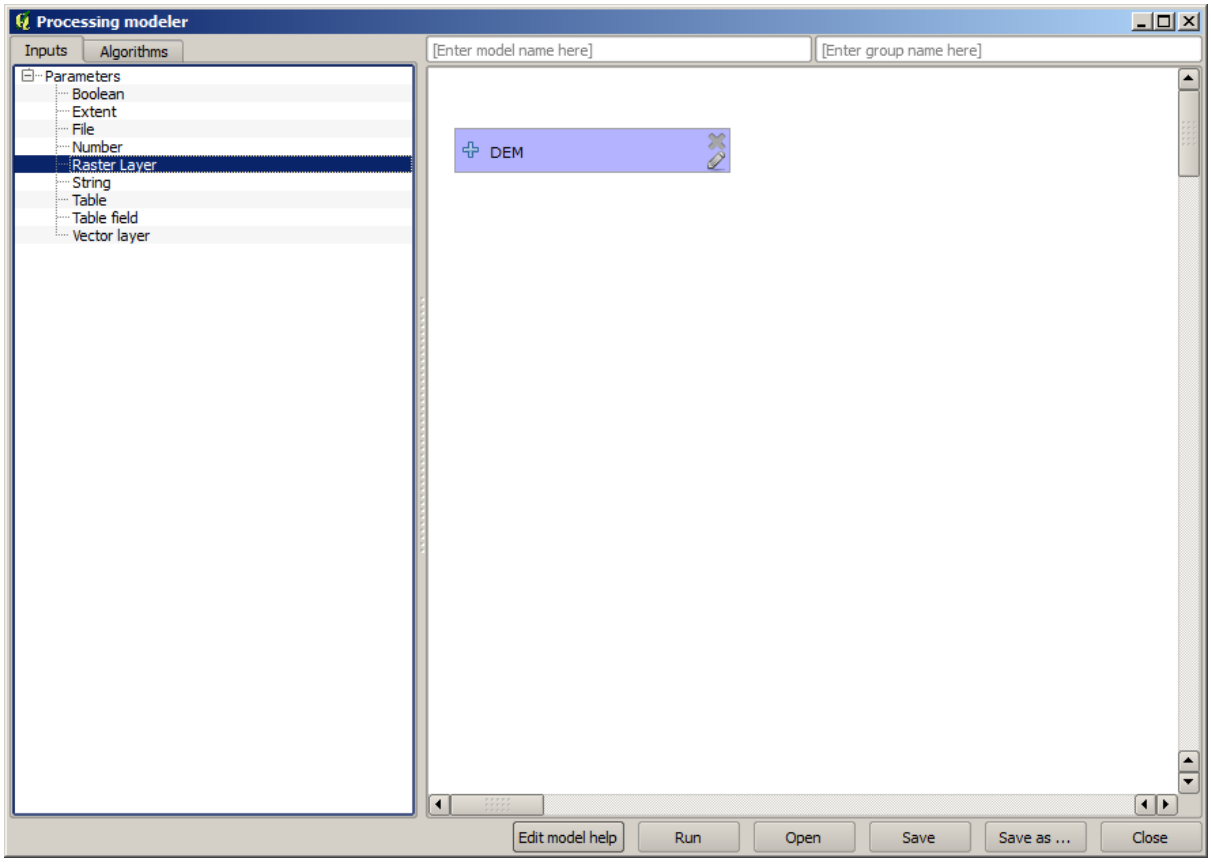
4. Here we will have to define the input we want:

1. Since we expect this raster layer to be a DEM, we will call it `DEM`. That's the name that the user of the model will see when running it.
2. Since we need that layer to work, we will define it as a mandatory layer.

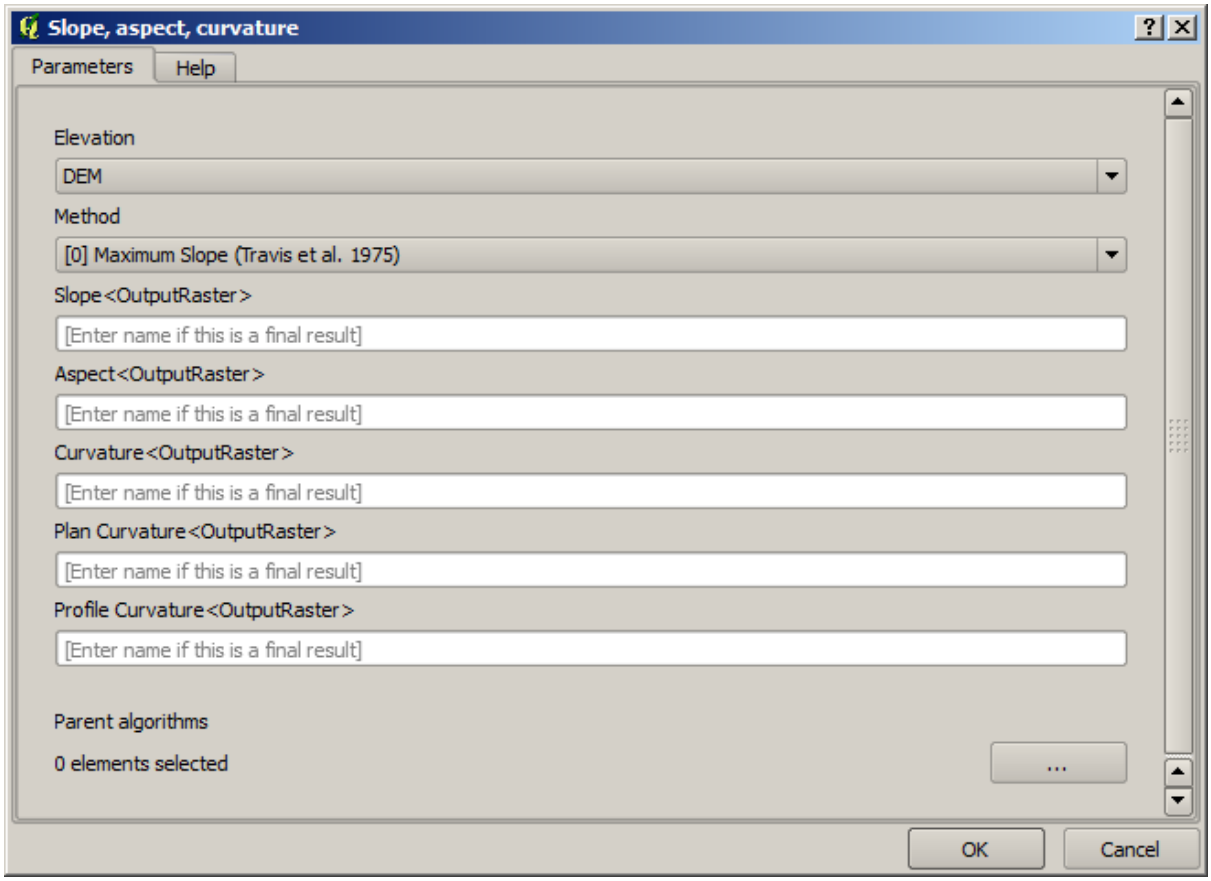
Here is how the dialog should be configured.



5. Click on *OK* and the input will appear in the modeler canvas.



6. Now let's move to the *Algorithms* tab.
7. The first algorithm we have to run is the *Slope, aspect, curvature* algorithm. Locate it in the algorithm list, double-click on it and you will see the dialog shown below.

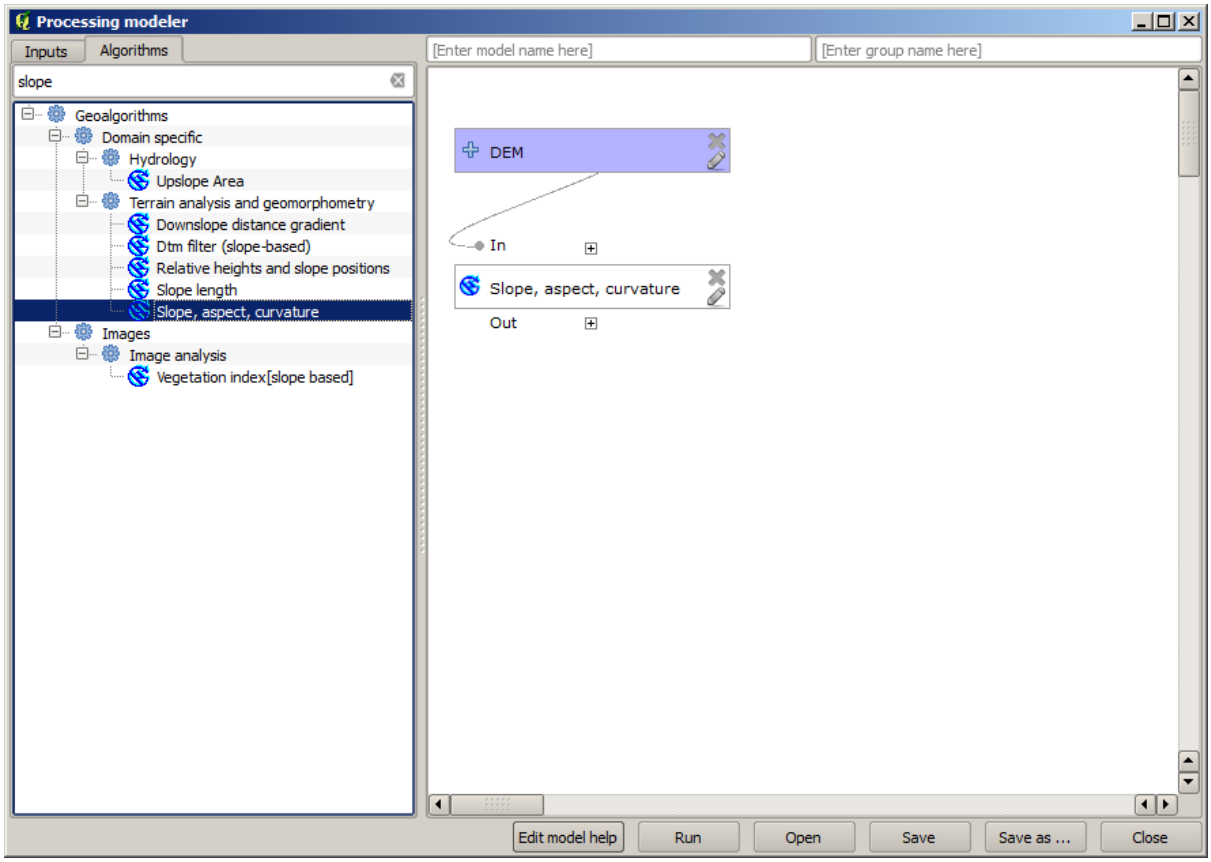


This dialog is very similar to the one that you can find when running the algorithm from the toolbox, but the element that you can use as parameter values are not taken from the current QGIS project, but from the model itself. That means that, in this case, we will not have all the raster layers of our project available for the *Elevation* field, but just the ones defined in our model. Since we have added just one single raster input named `DEM`, that will be the only raster layer that we will see in the list corresponding to the *Elevation* parameter.

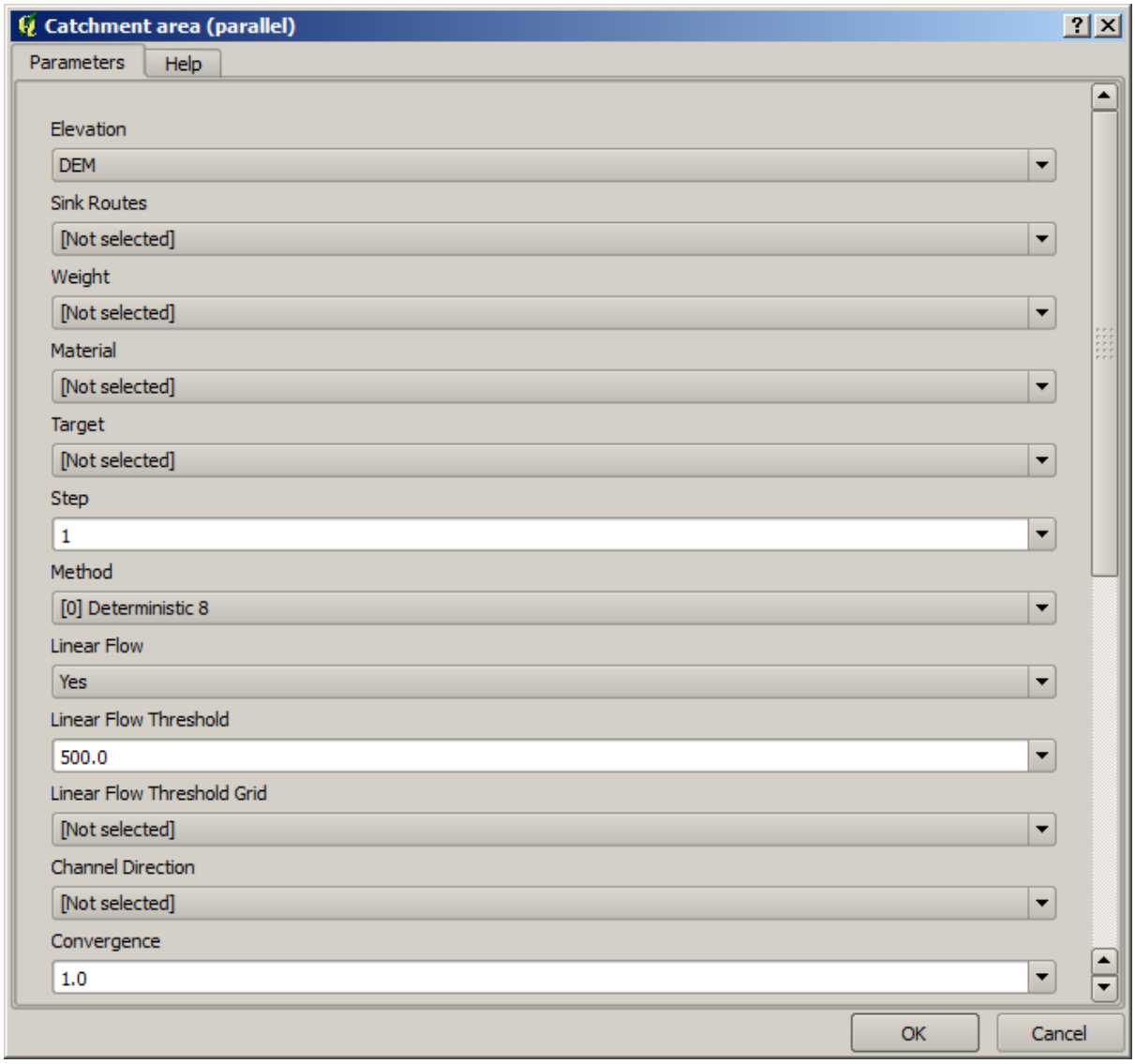
Output generated by an algorithm are handled a bit differently when the algorithm is used as a part of a model. Instead of selecting the filepath where you want to save each output, you just have to specify if that output is an intermediate layer (and you do not want it to be preserved after the model has been executed), or it is a final one. In this case, all layers produced by this algorithm are intermediate. We will only use one of them (the slope layer), but we do not want to keep it, since we just need it to calculate the TWI layer, which is the final result that we want to obtain.

When layers are not a final result, you should just leave the corresponding field. Otherwise, you have to enter a name that will be used to identify the layer in the parameters dialog that will be shown when you run the model later.

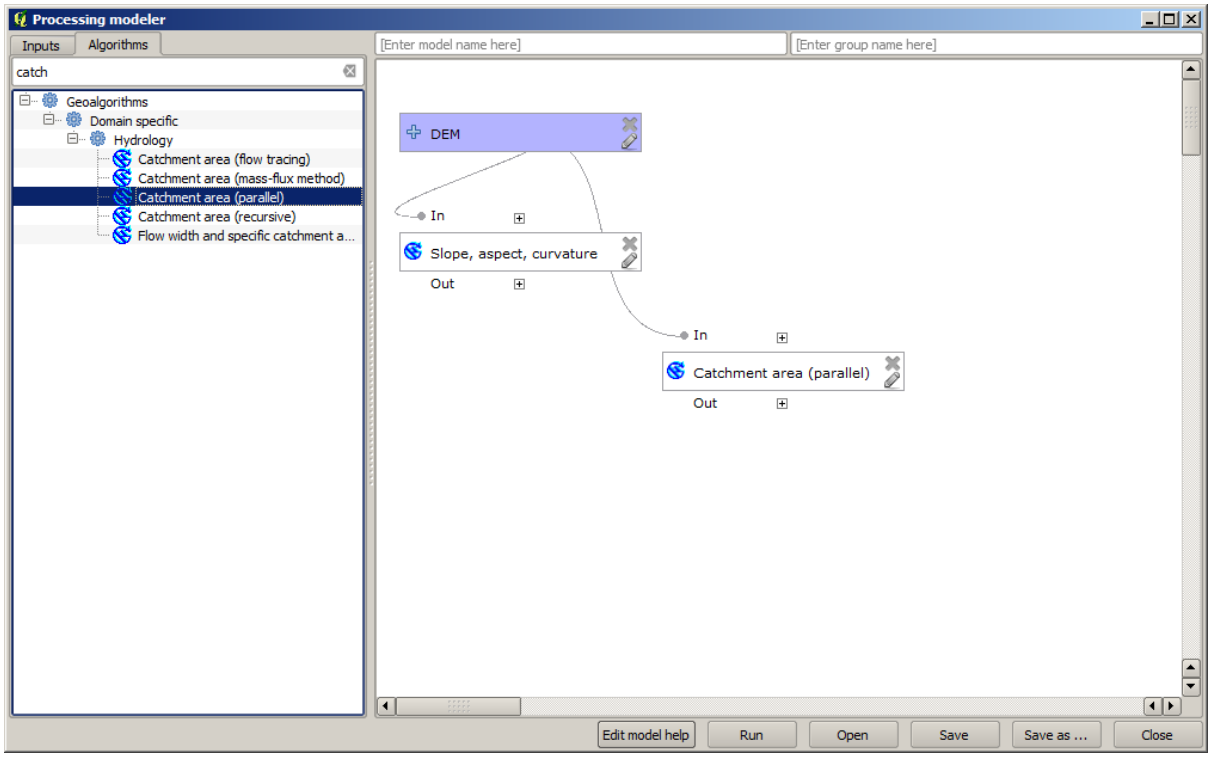
8. There is not much to select in this first dialog, since we do not have but just one layer in our model (The DEM input that we created). Actually, the default configuration of the dialog is the correct one in this case, so you just have to press *OK*. This is what you will now have in the modeler canvas.



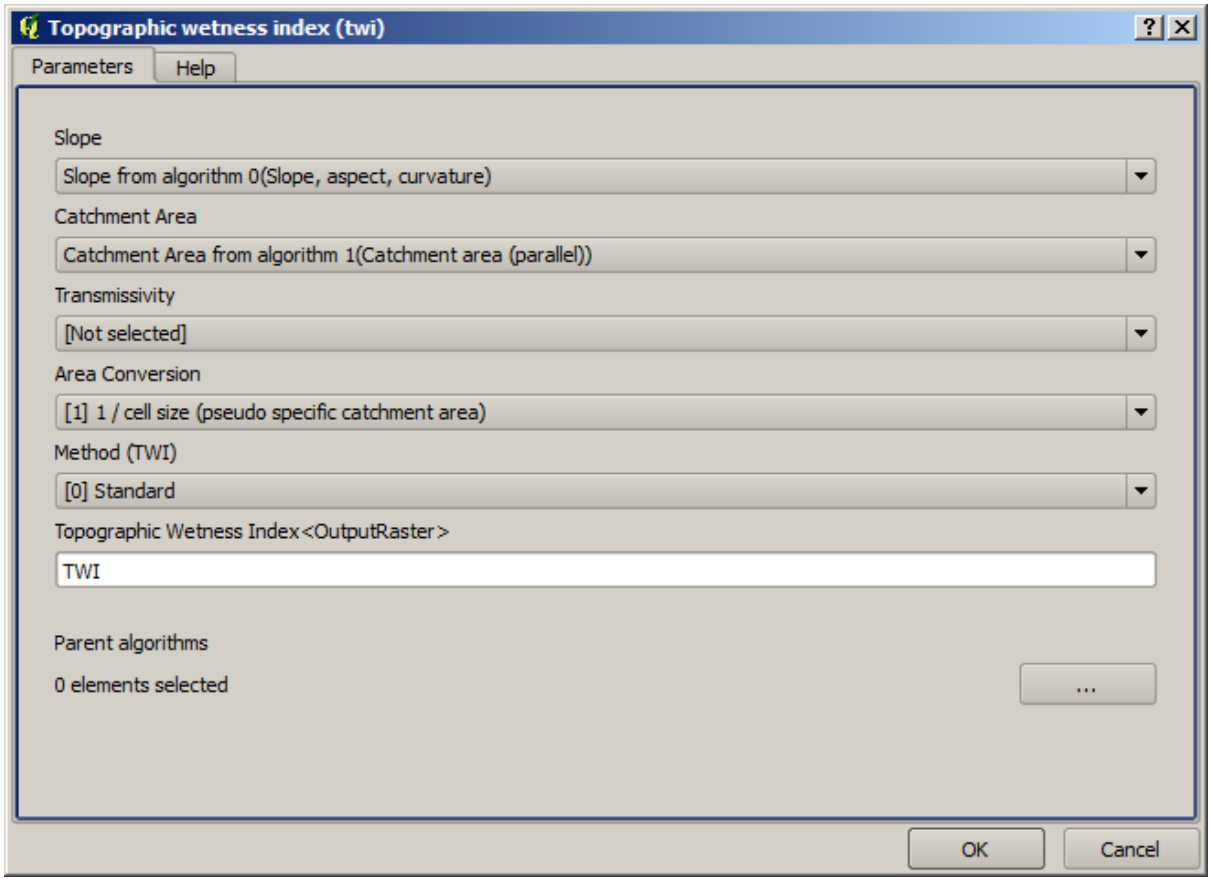
9. The second algorithm we have to add to our model is the catchment area algorithm. We will use the algorithm named *Catchment area (Paralell)*. We will use the DEM layer again as input, and none of the ouputs it produces are final, so here is how you have to fill the corresponding dialog.



Now your model should look like this.



10. The last step is to add the *Topographic wetness index* algorithm, with the following configuration.

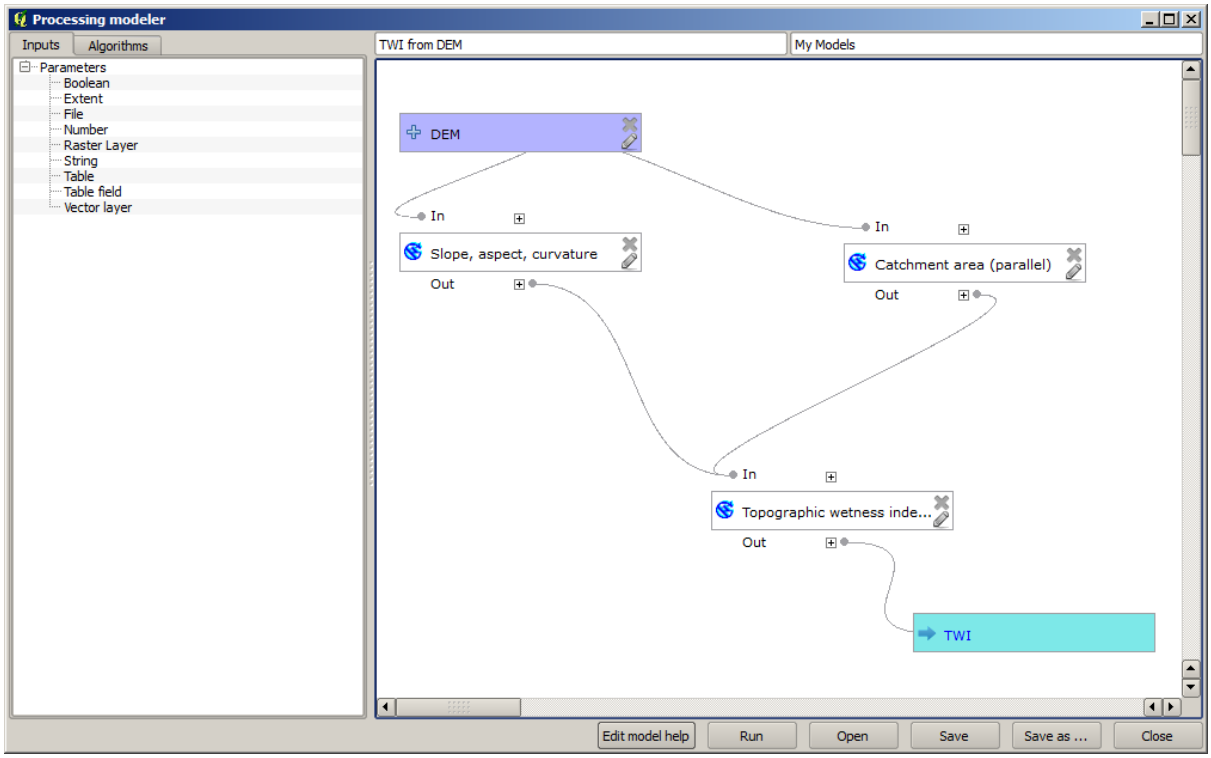


In this case, we will not be using the DEM as input, but instead, we will use the slope and catchment area layers that are calculated by the algorithms that we previously added. As you add new algorithms, the outputs

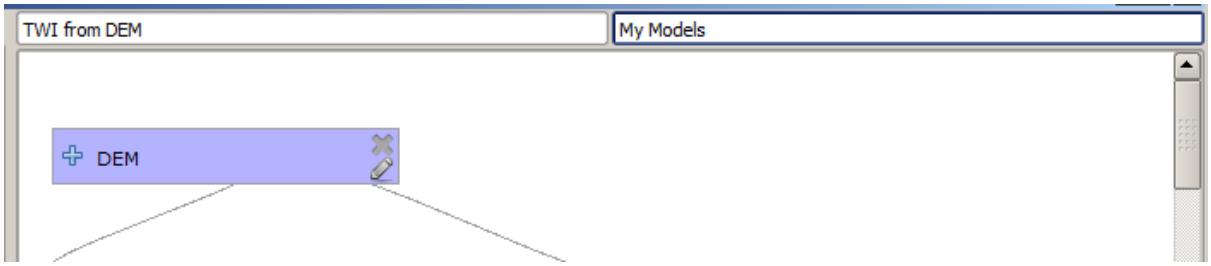
they produce become available for other algorithms, and using them you link the algorithms, creating the workflow.

11. In this case, the output `TWI` layer is a final layer, so we have to indicate so. In the corresponding textbox, enter the name that you want to be shown for this output.

    Now our model is finished and it should look like this.

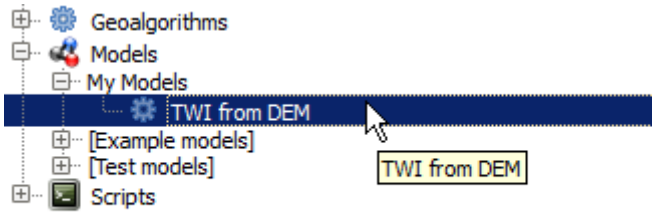


12. Enter a name and a group name in the upper part of the model window.

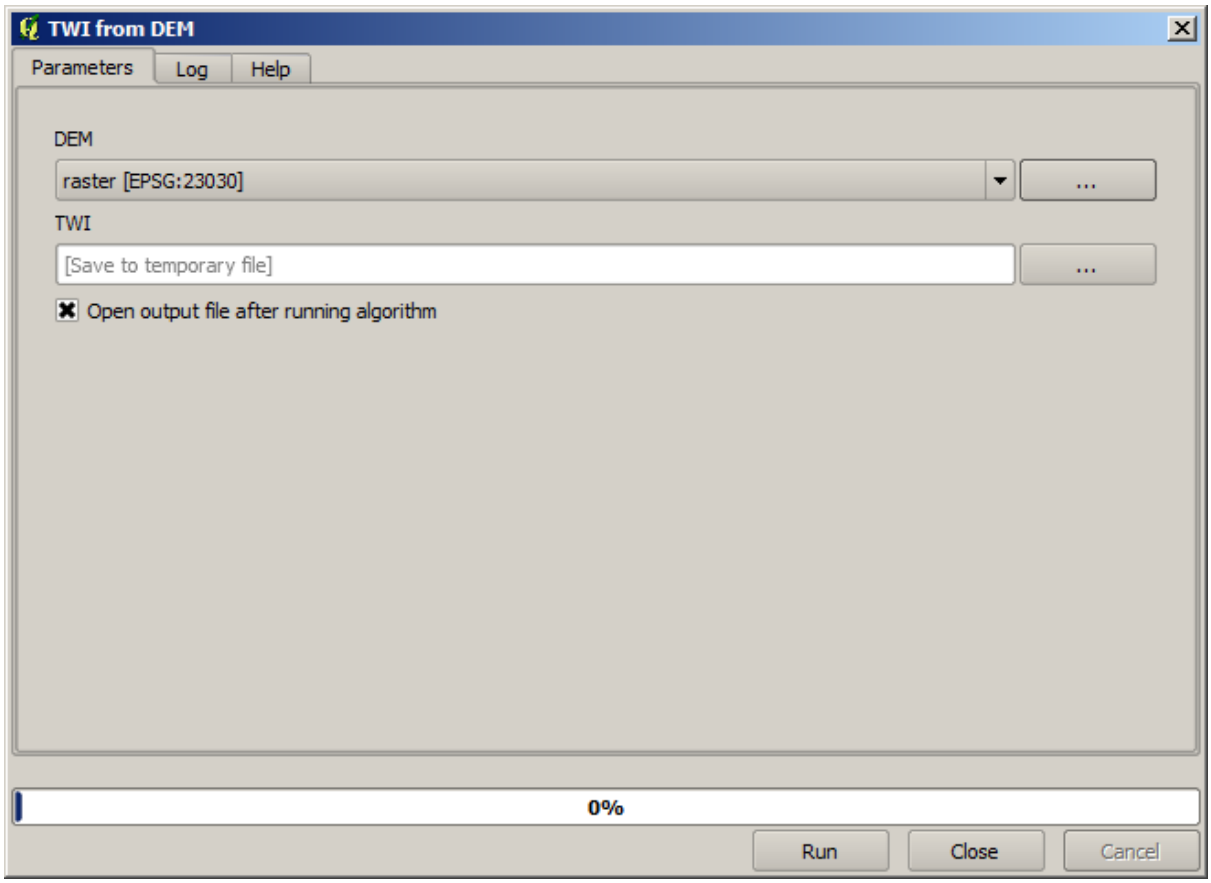


13. Save it clicking on the *Save* button. You can save it anywhere you want and open it later, but if you save it in the models folder (which is the folder that you will see when the save file dialog appears), your model will also be available in the toolbox as well. So stay on that folder and save the model with the filename that you prefer.

14. Now close the modeler dialog and go to the toolbox. In the *Models* entry you will find your model.



15. You can run it just like any normal algorithm, double-clicking on it.



As you can see, the parameters dialog, contain the input that you added to the model, along with the outputs that you set as final when adding the corresponding algorithms.

16. Run it using the DEM as input and you will get the TWI layer in just one single step.

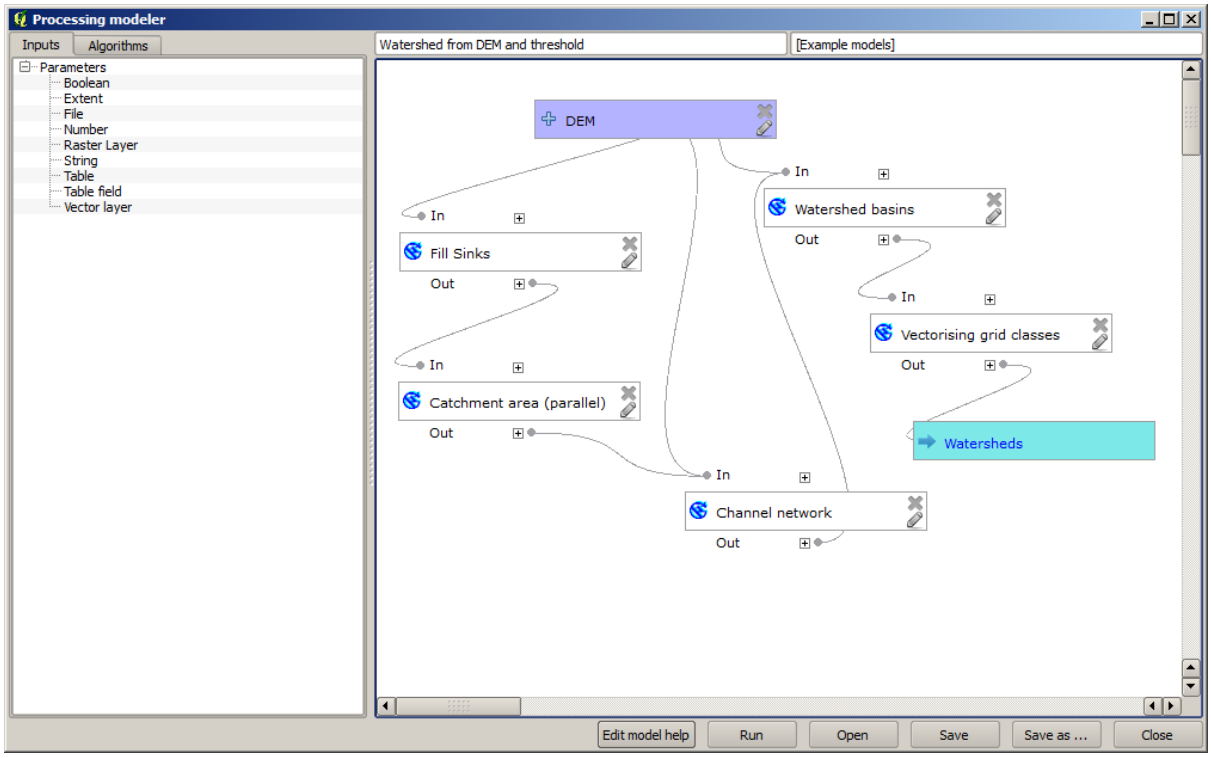
17.18 More complex models

Примечание: In this lesson we will work with a more complex model in the graphical modeler.

The first model that we created in the previous chapter was a very simple one, with just one input and three algorithms. More complex models can be created, with different types of inputs and containing more steps. For this chapter we will work with a model that creates a vector layer with watersheds, based on a DEM and a threshold value. That will be very useful for calculating several vector layers corresponding to different thresholds, without having to repeat each single step each time.

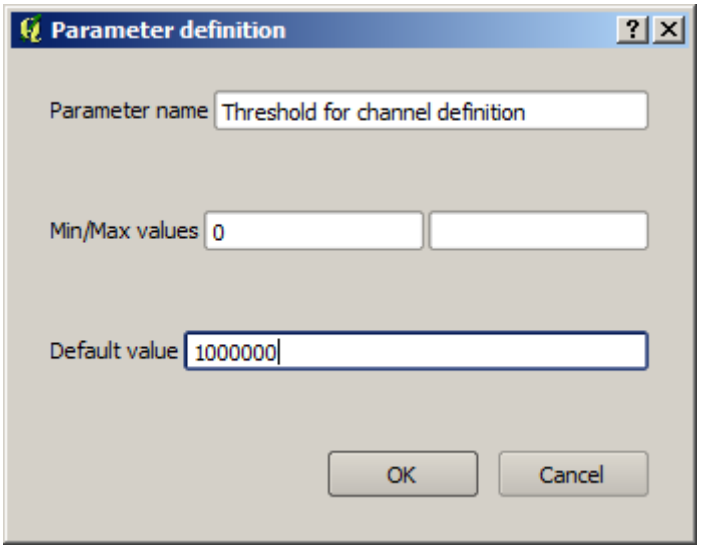
This lesson does not contain instructions about how to create your model. You already know the necessary steps (we saw them in a *previous lesson*) and you have already seen the basic ideas about the modeler, so you should try it yourself. Spend a few minutes trying to create your model, and don't worry about making mistakes. Remember: first add the inputs and then add the algorithms that use them to create the workflow.

Примечание: In case you could not create the full model yourself and you need some extra help, the data folder corresponding to this lesson contains an „almost" finished version of it. Open the modeler and then open the model file that you will find in the data folder. You should see something like this.

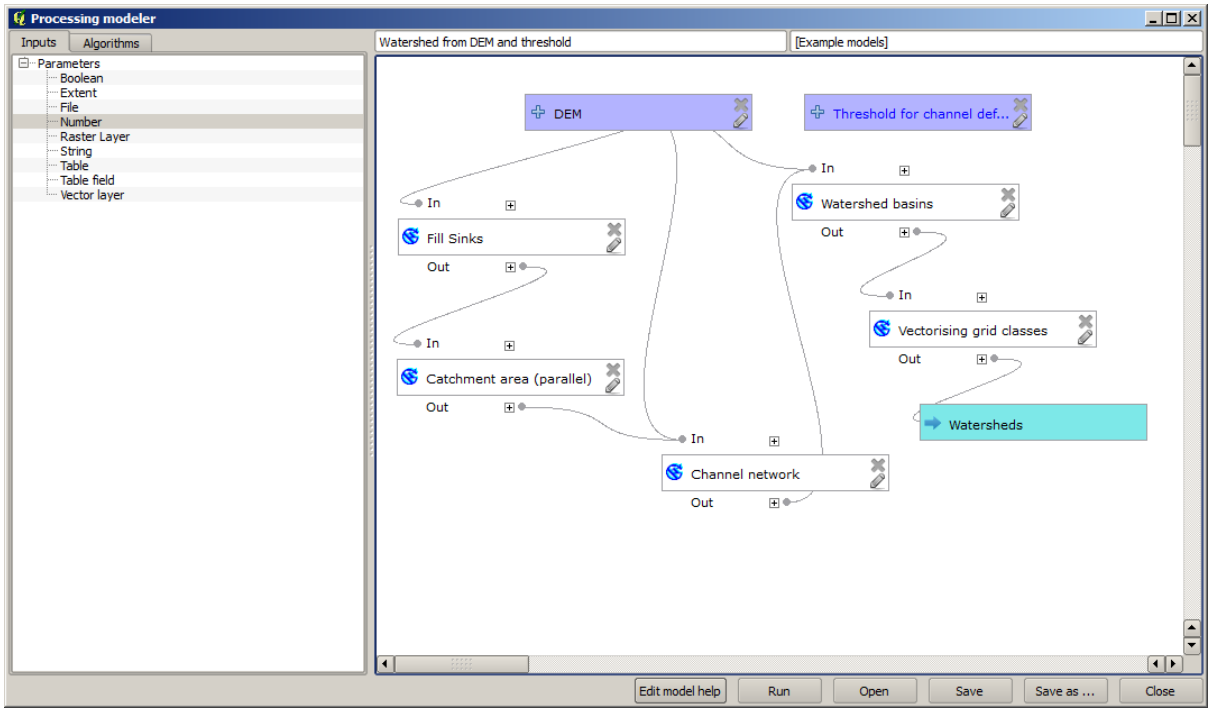


This model contains all the steps needed to complete the calculation, but it just has one input: the DEM. That means that the threshold for channel definition uses a fixed value, which makes the model not as useful as it could be. That is not a problem, since we can edit the model, and that is exactly what we will do.

1. First, let's add a numerical input. That will ask the user for a numerical input that we can use when such a value is needed in any of the algorithms included in our model.
2. Click on the *Number* entry in the *Inputs* tree, and you will see the corresponding dialog.
3. Fill it with the following values.
   - *Parameter name*: `Threshold for channel definition`
   - *Default value*: `1,000,000`

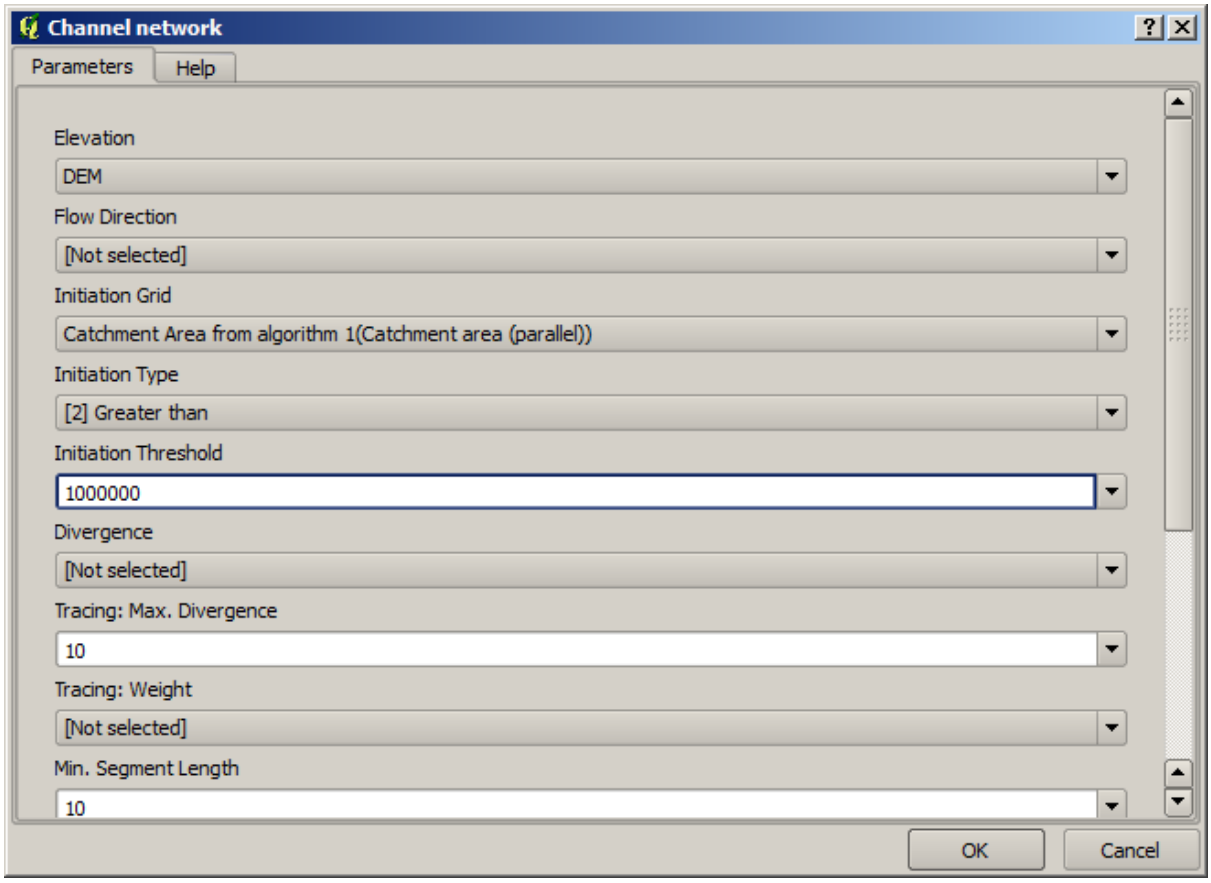


Now your model should look like this.



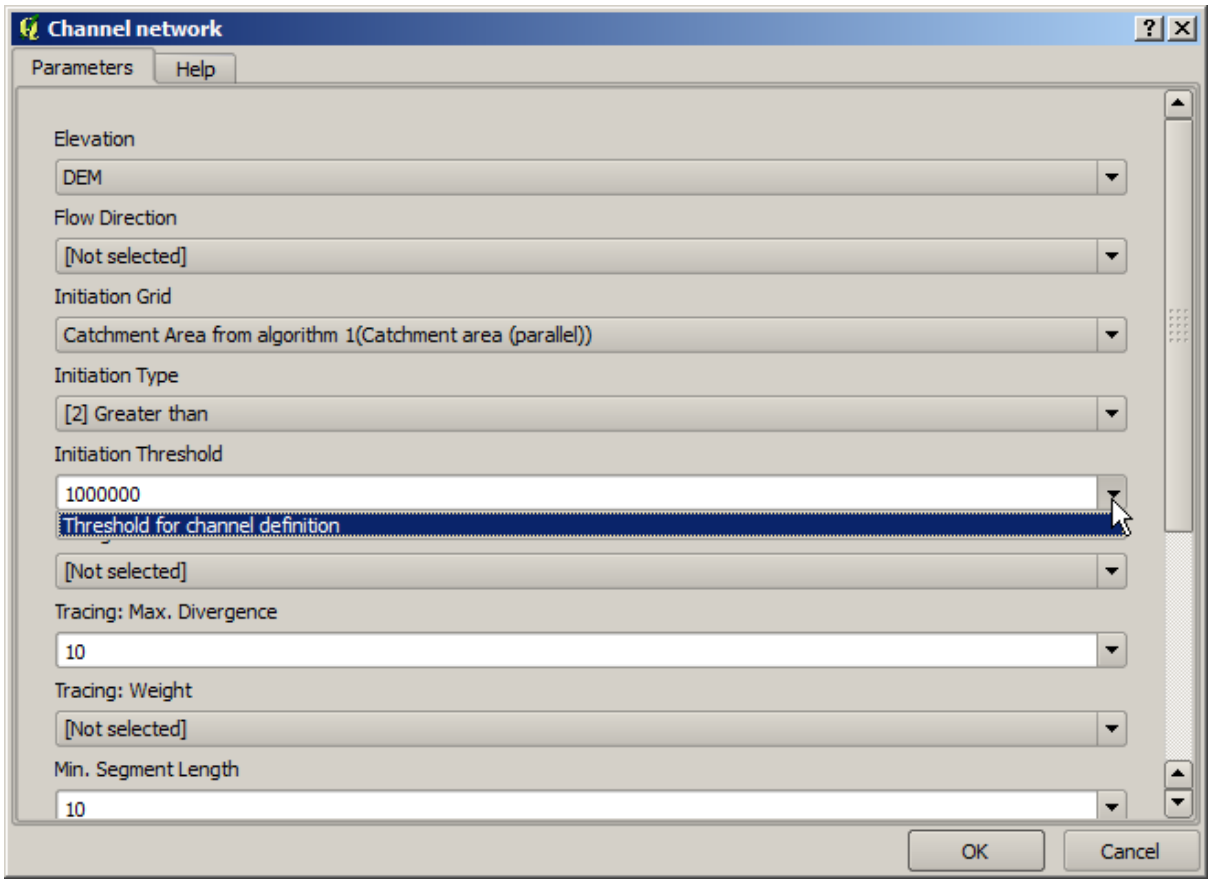
The input that we have just added is not used, so the model hasn't actually changed. We have to link that input to the algorithm that uses it, in this case the *Channel network* one. To edit an algorithm that already exists in the modeler, just click on the pen icon on the corresponding box in the canvas.

4. Click on the *Channel network* algorithm and you will see something like this.



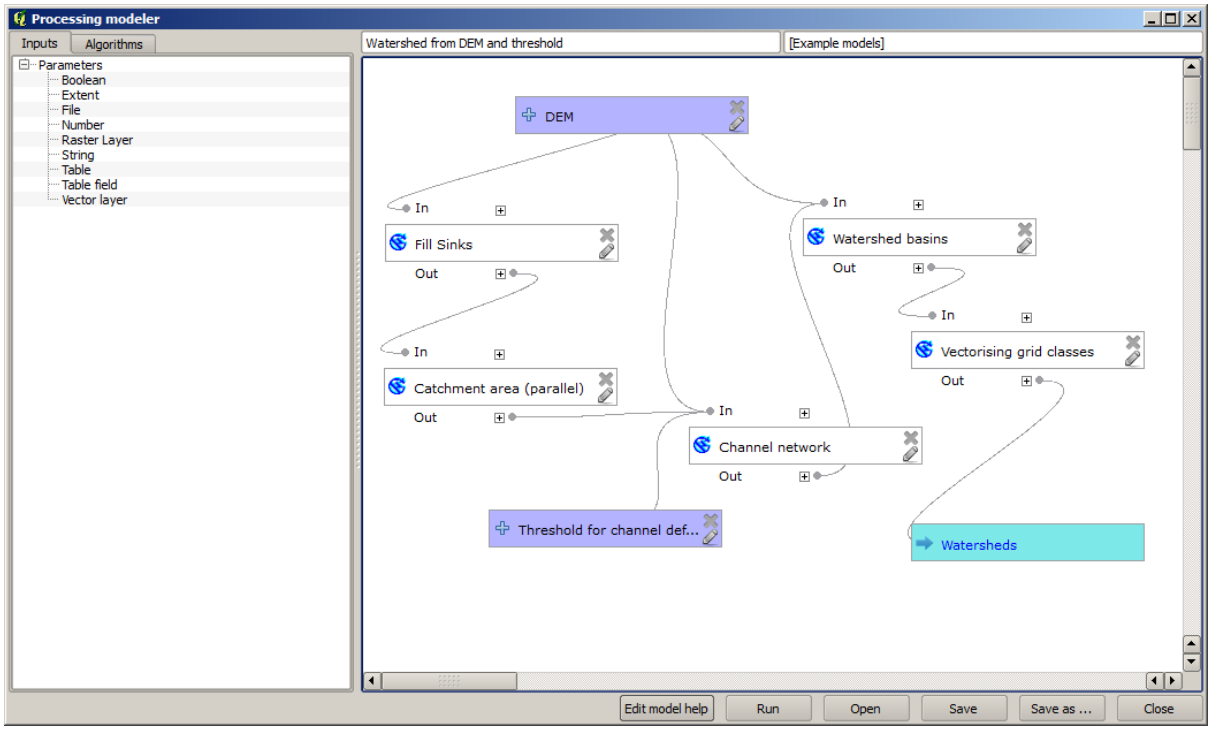
The dialog is filled with the current values used by the algorithm. You can see that the *Initiation threshold* parameter has a fixed value of 1,000,000 (this is also the default value of the algorithm, but any other value could be put in there). However, you might notice that the parameter is not entered in a common text box, but in a drop-down menu.

5. Unfold the threshold parameter menu and you will see something like this.



The input that we added is there and we can select it. Whenever an algorithm in a model requires a numerical value, you can hardcode it and directly type it, or you can use any of the available inputs and values (remember that some algorithms generate single numerical values. We will see more about this soon). In the case of a string parameter, you will also see string inputs and you will be able to select one of them or type the desired fixed value.

6. Select the `Threshold for channel definition` input in the *Initiation threshold* parameter.
7. Click on *OK* to apply the changes to your model. Now the design of the model should look like this.



8. The model is now complete. Run it using the DEM that we have used in previous lessons, and with different threshold values.

   Below you have a sample of the result obtained for different values. You can compare with the result for the default value, which is the one we obtained in the *hydrological analysis lesson*.

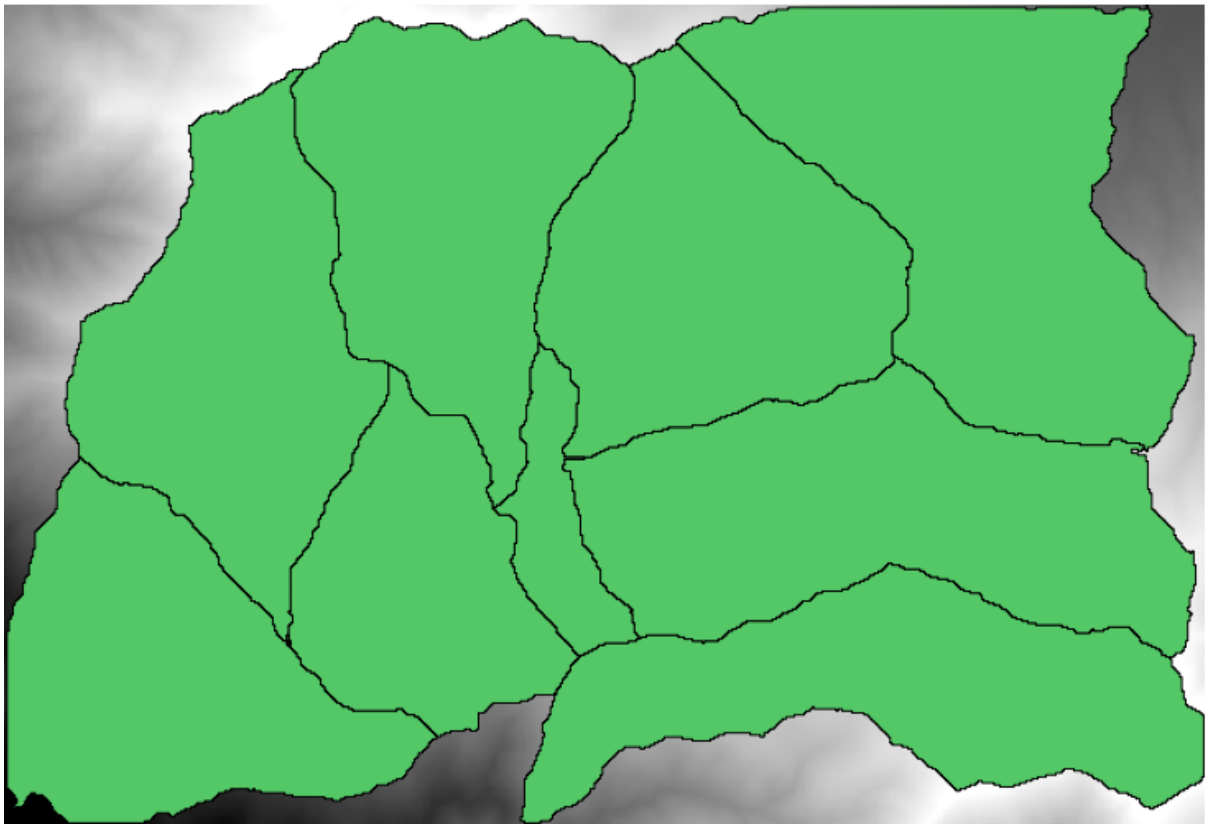


Рис. 17.1: Threshold = 100,000

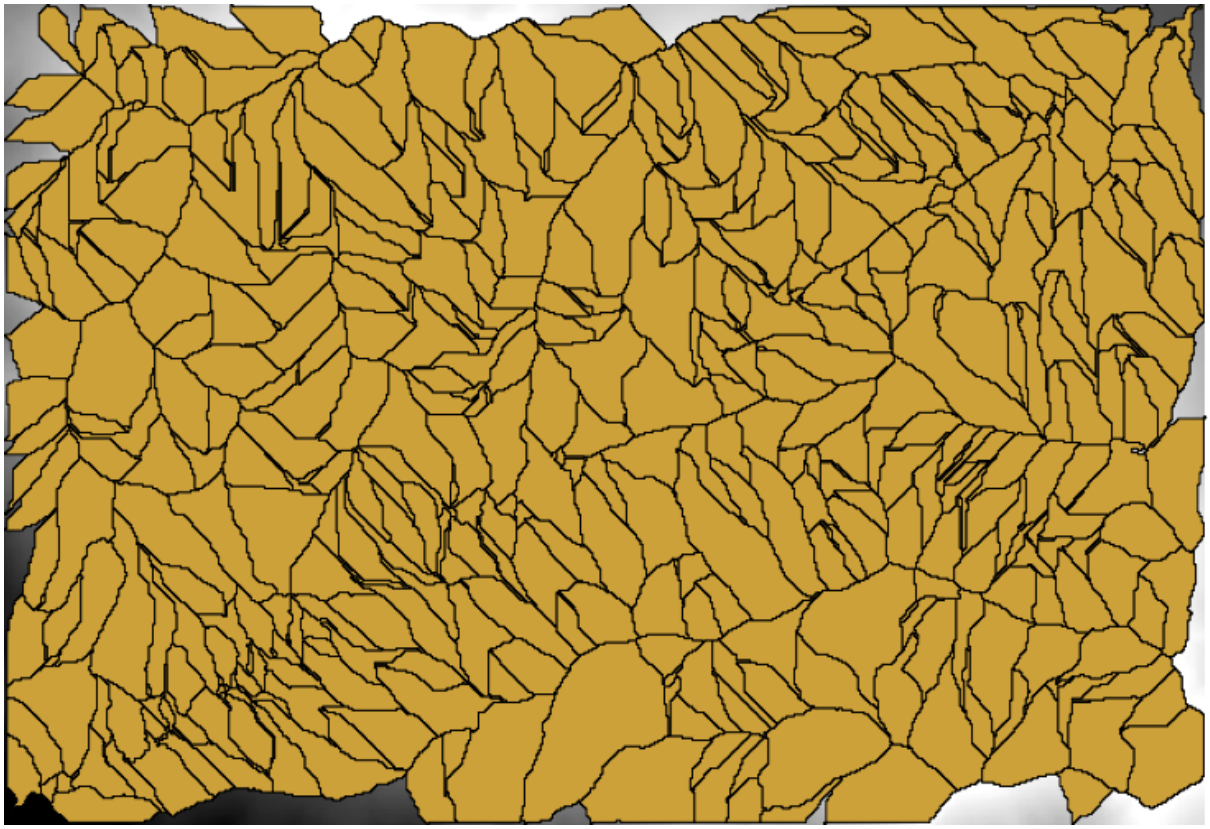


Рис. 17.2: Threshold = 1,0000,000

17.19 Numeric calculations in the modeler

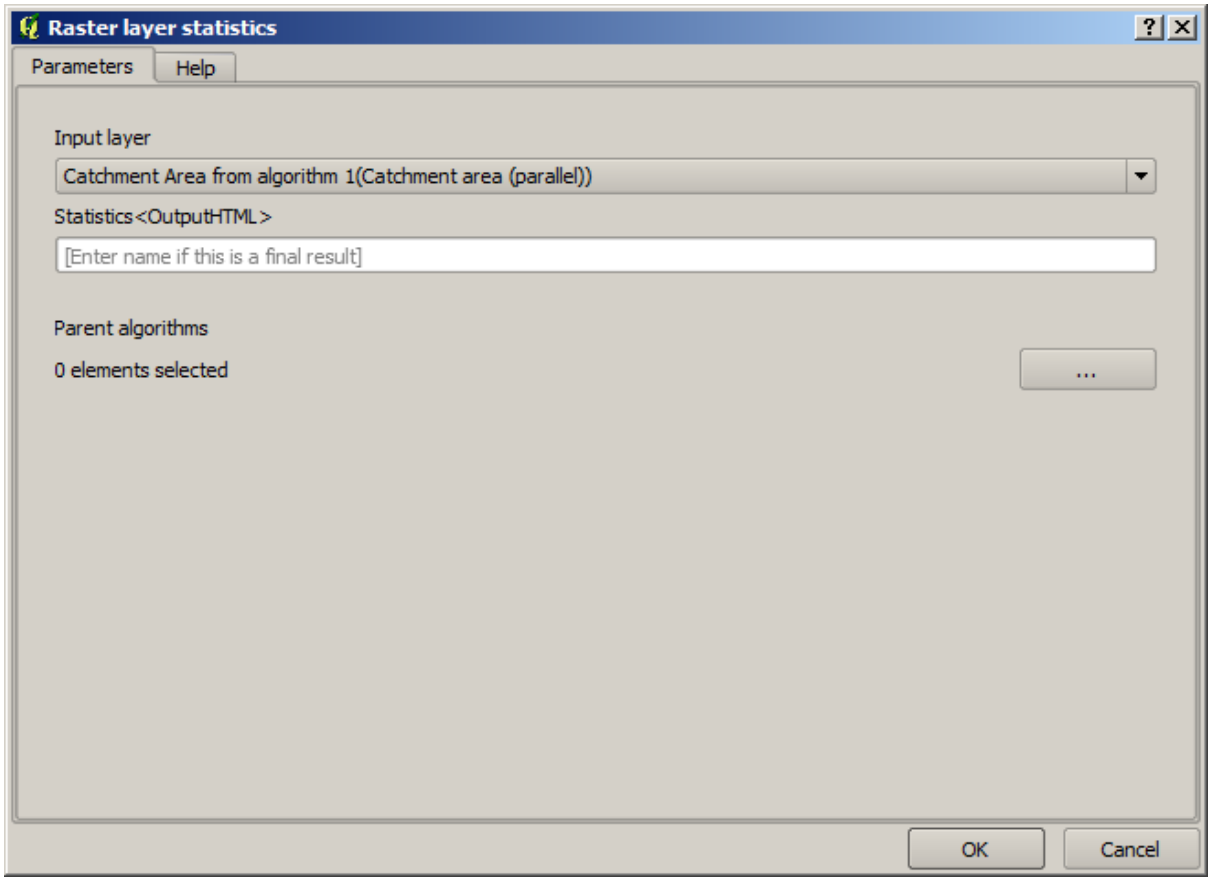
Предупреждение: Beware, this chapter is not well tested, please report any issue; images are missing

Примечание: In this lesson we will see how to use numeric outputs in the modeler

For this lesson, we are going to modify the hydrological model that we created in the last chapter (open it in the modeler before starting), so we can automate the calculation of a valid threshold value and we do not have to ask the user to enter it. Since that value refers to the variable in the threshold raster layer, we will extract it from that layer, based on some simple statistical analysis.

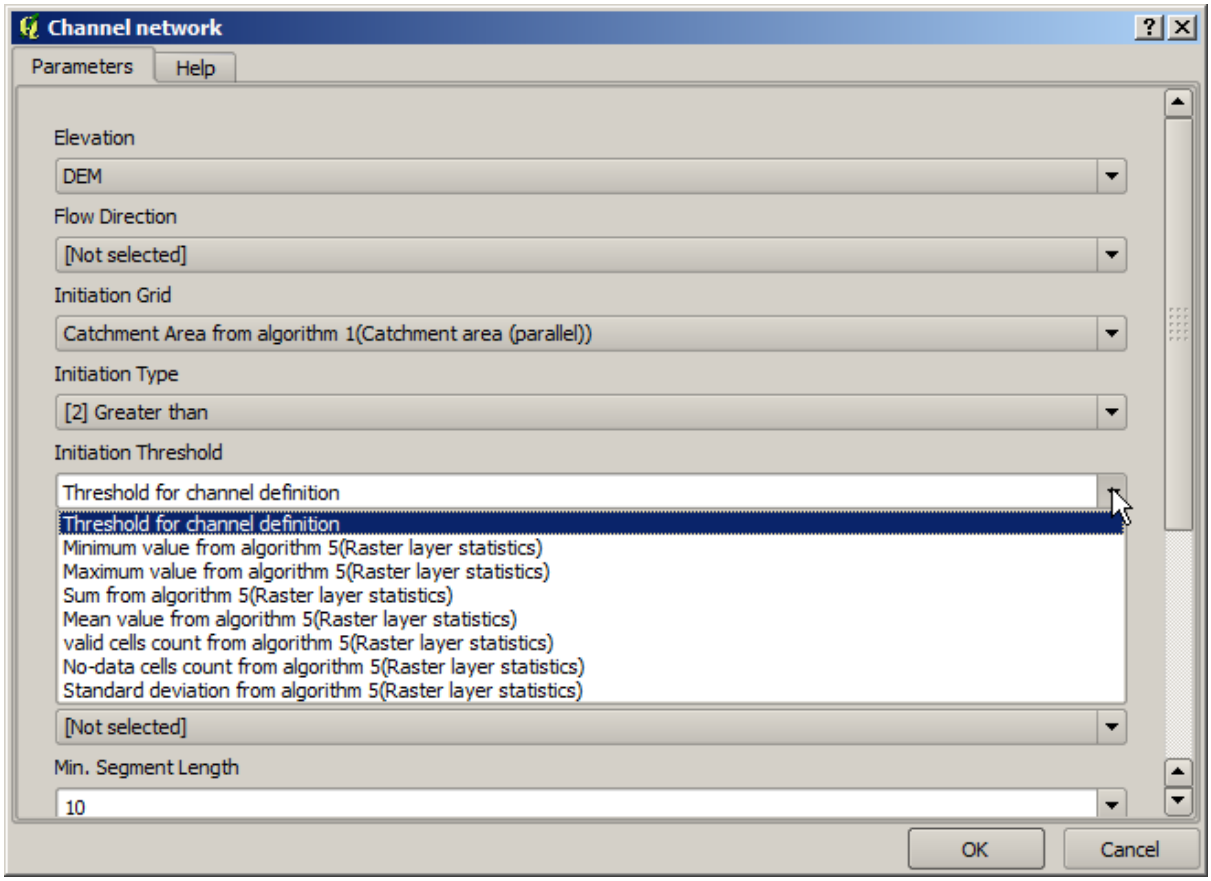
Starting with the aforementioned model, let's do the following modifications:

First, calculate statistics of the flow accumulation layer using the *Raster layer statistics* algorithm.



This will generate a set of statistical values that will now be available for all numeric fields in other algorithms.

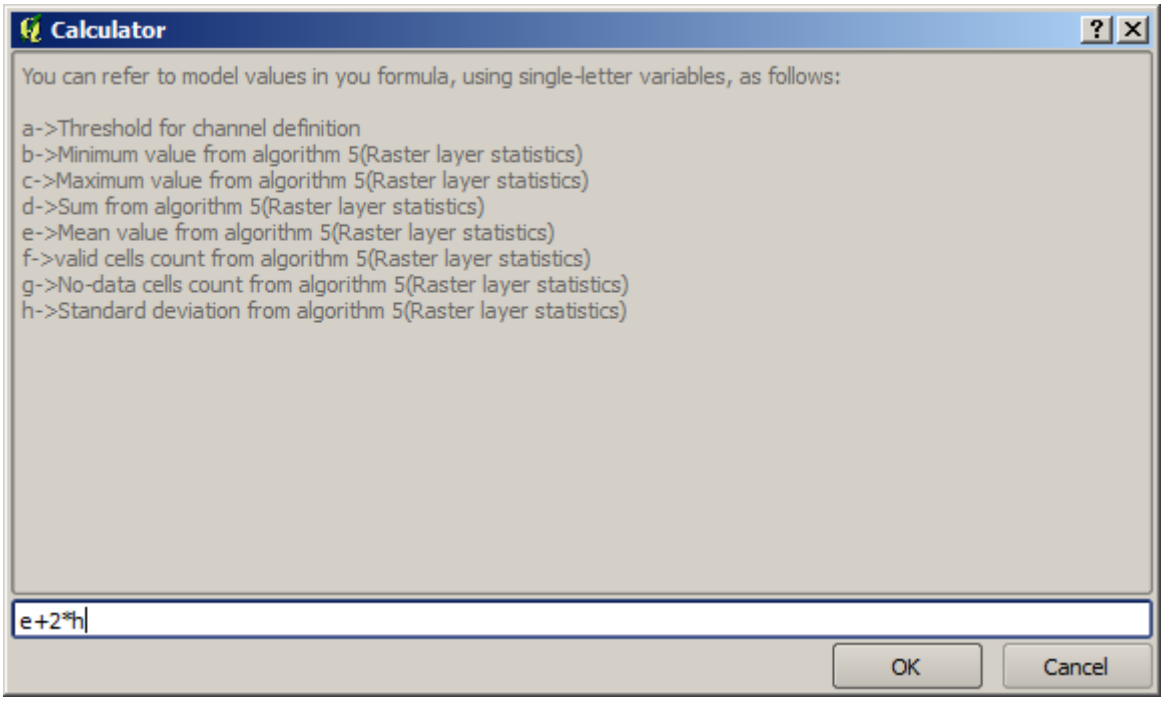
If you edit the *Channel network* algorithm, as we did in the last lesson, you will see now that you have other options apart from the numeric input that you added.



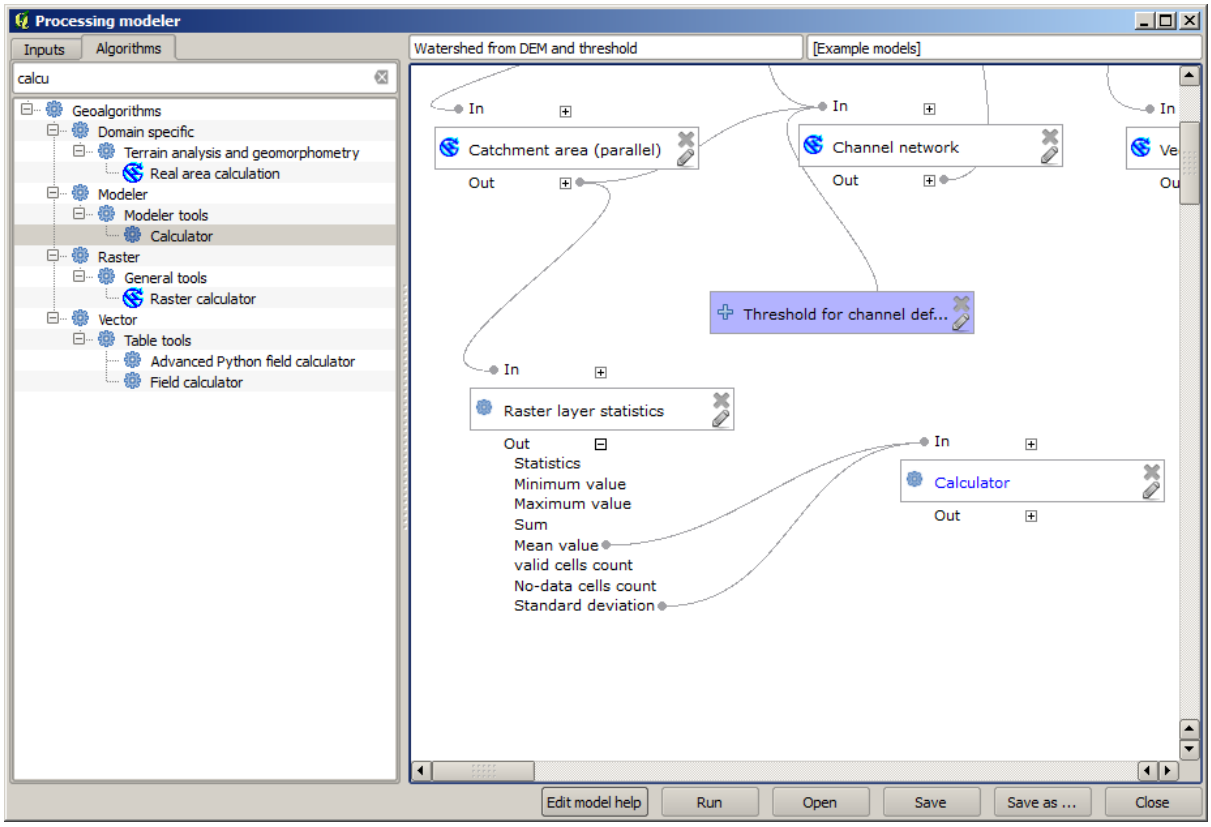
However, none of this values is suitable for being used as a valid threshold, since they will result in channel networks that will not be very realistic. We can, instead, derive some new parameter based on them, to get a better result. For instance, we can use the mean plus 2 times the standard deviation.

To add that arithmetical operation, we can use the calculator that you will find in the *Geoalgorithms/modeler/modeler-tools* group. This group contains algorithms that are not very useful outside of the modeler, but that provide some useful functionality when creating a model.

The parameters dialog of the calculator algorithm looks like this:



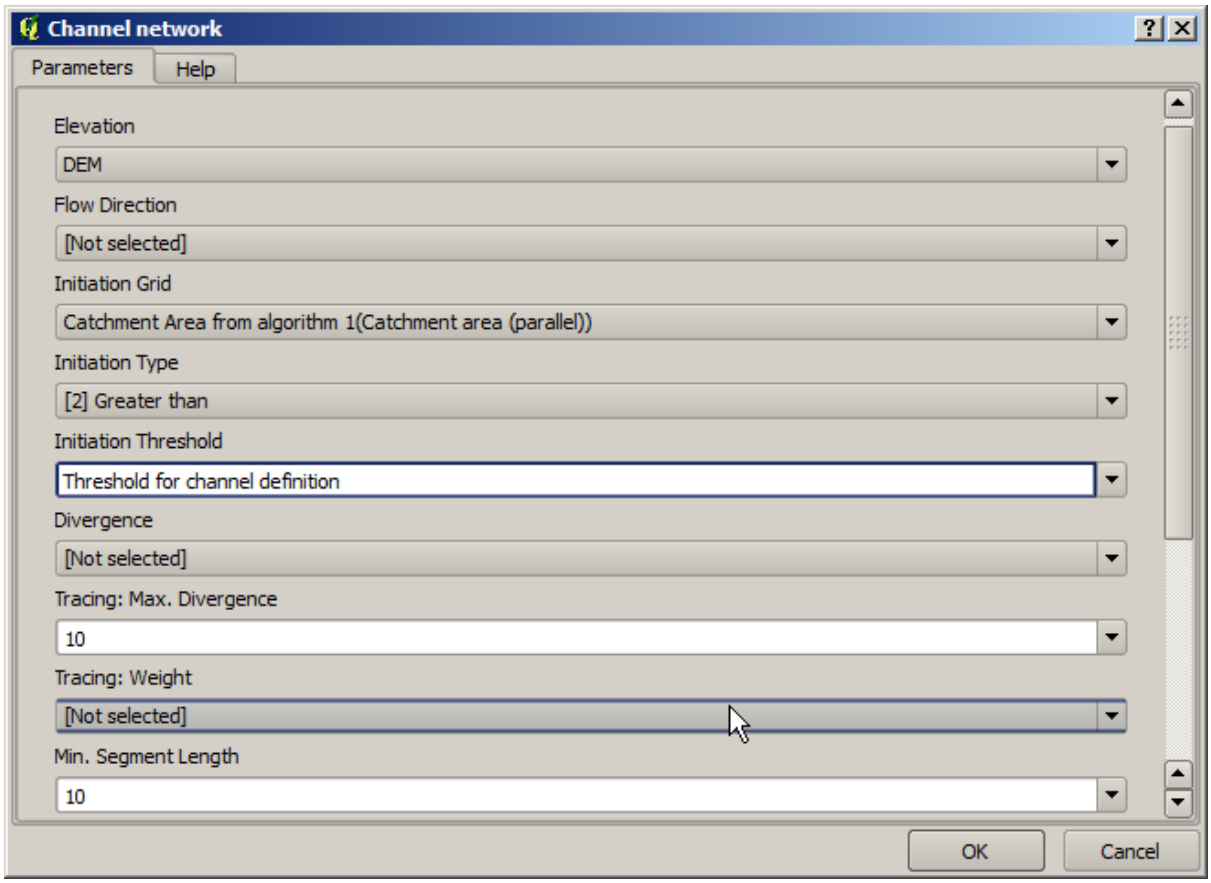
As you can see, the dialog is different to the other ones we have seen, but you have in there the same variables that were available in the *Threshold* field in the *Channel network* algorithm. Enter the above formula and click on *OK* to add the algorithm.



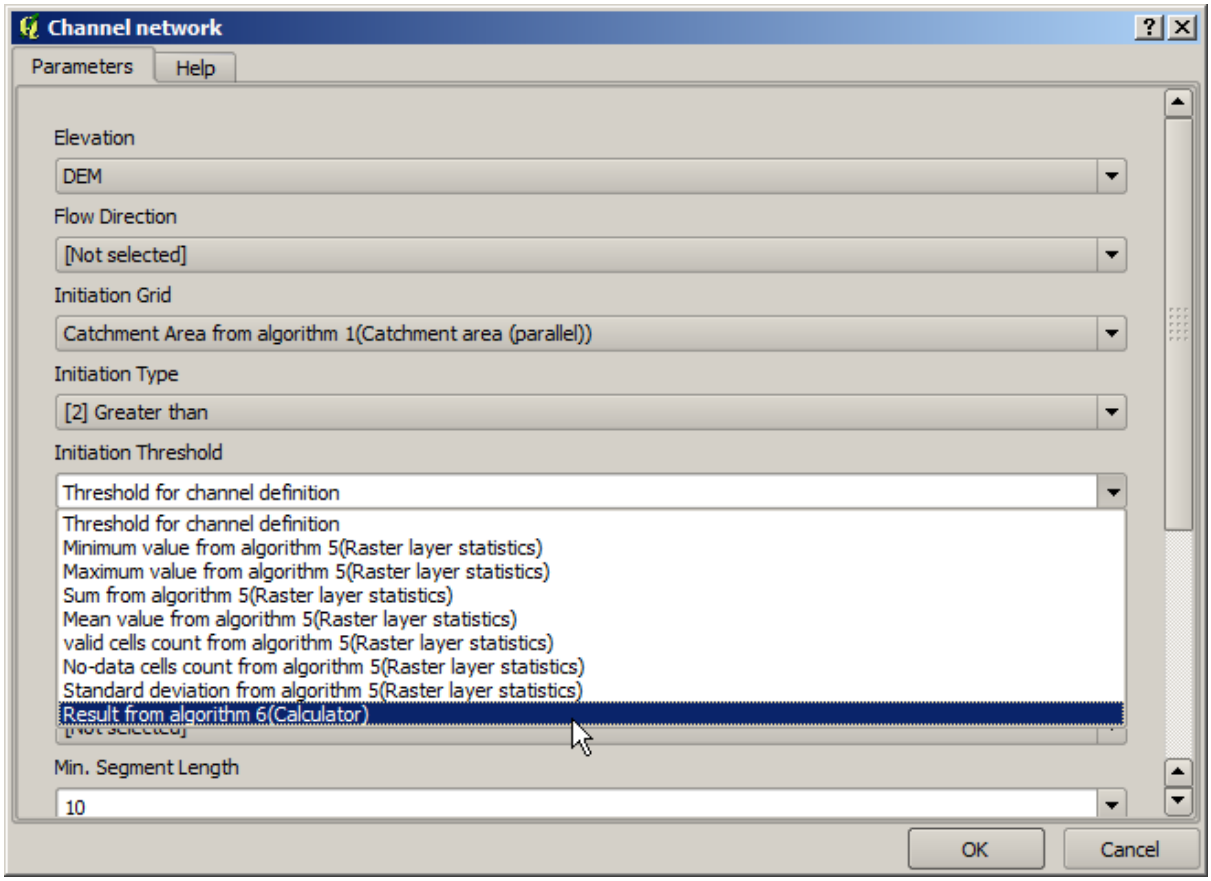
If you expand the outputs entry, as shown above, you will see that the model is connected to two of the values, namely the mean and the standard deviation, which are the ones that we have used in the formula.

Adding this new algorithm will add a new numeric value. If you go again to the *Channel network* algorithm, you can

now select that value in the *Threshold* parameter.



Click on *OK* and your model should look like this.



We are not using the numeric input that we added to the model, so it can be removed. Right–click on it and select *Remove*

Предупреждение: todo: Add image

Our new model is now finished.

17.20 A model within a model

Предупреждение: Beware, this chapter is not well tested, please report any issue; images are missing

Примечание: In this lesson we will see how to use a model within a bigger model.

We have already created a few models, and in this lesson we are going to see how we can combine them on a single bigger one. A model behaves like any other algorithm, which means that you can add a model that you have already created as part of another one that you create after that.

In this case, we are going to expand our hydrological model, by adding the mean TWI value in each of the basins that it generates as result. To do that, we need to calculate the TWI, and to compute the statistics. Since we have already created a model to calculate TWI from a DEM, it is a good idea to reuse that model instead of adding the algorithms it contains individually.

Let's start with the model we used as starting point for the last lesson.

Предупреждение: todo: Add image

First, we will add the TWI model. For it to be available, it should have been saved on the models folder, since otherwise it will not be shown in the toolbox or the algorithms list in the modeler. Make sure you have it available.

Add it to the current model and use the input DEM as its input. The output is a temporary one, since we just want the TWI layer to compute the statistics. The only output of this model we are creating will still be the vector layer with the watersheds.

Here is the corresponding parameters dialog:

Предупреждение: todo: Add image

Now we have a TWI layer that we can use along with the watersheds vector layer, to generate a new one which contains the values of the TWI corresponding to each watershed.

This calculation is done using the *Grid statistics in polygons* algorithm. Use the layers mentioned above as input, to create the final result.

Предупреждение: todo: Add image

The output of the *Vectorize grid classes* algorithm was originally our final output, but now we just want it as an intermediate result. To change that, we have to edit the algorithm. Just double–click on it to see its parameters dialog, and delete the name of the output. That will make it a temporary output, as it is by default.

Предупреждение: todo: Add image

This is how the final model should look like:

Предупреждение: todo: Add image

As you see, using a model in another model is nothing special, and you can add it just like you add another algorithm, as long as the model is saved in the models folder and is available in the toolbox.

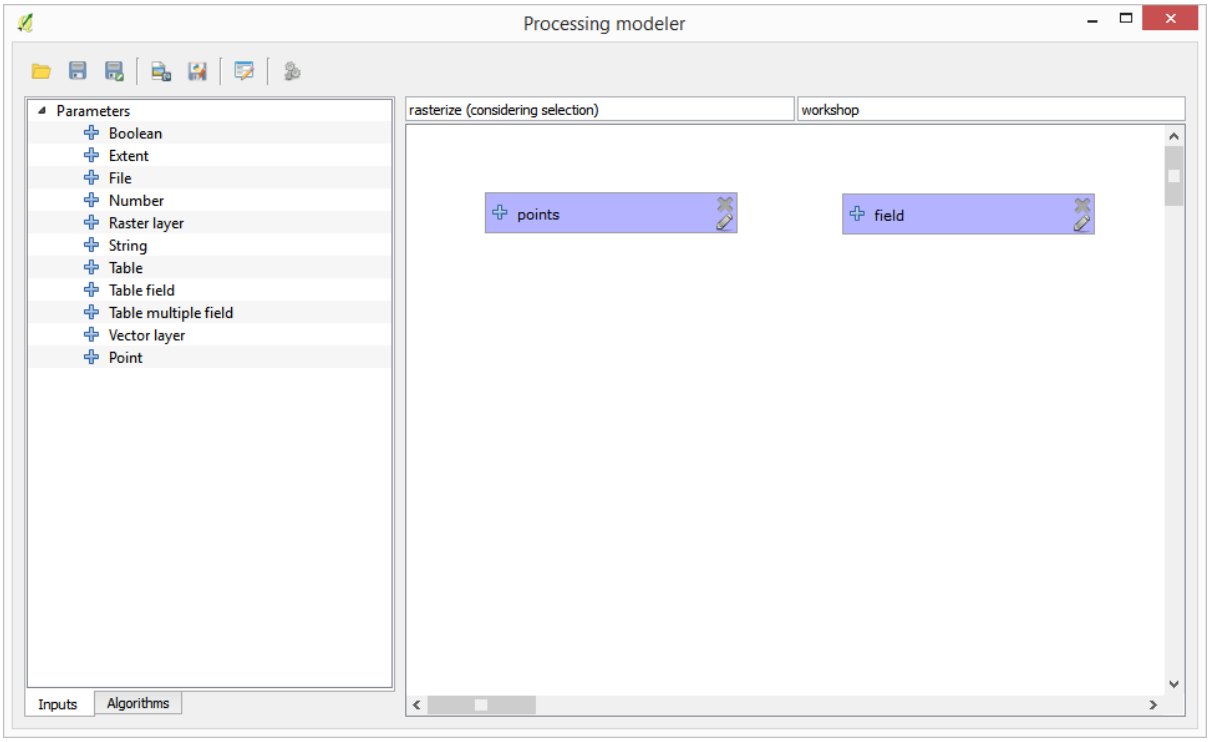
17.21 Using modeler-only tools for creating a model

Примечание: This lesson shows how to use some algorithms that are only available in the modeler, to provide additional functionality to models.

The goal of this lesson is to use the modeler to create an interpolation algorithm that takes into account the current selection, not just to use only selected features, but to use the extent of that selection to create the interpolated raster layer.

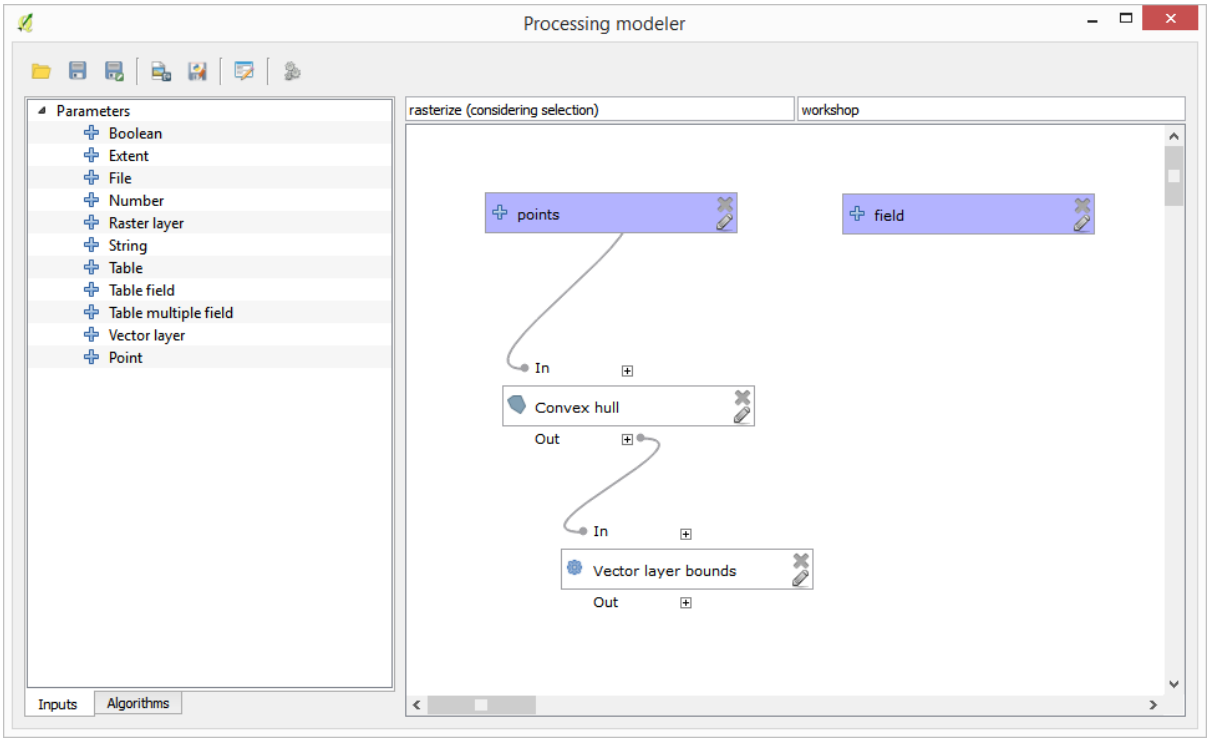
The interpolation process involves two steps, as it has been already explained in previous lessons: rasterizing the points layer and fill the no-data values that appear in the rasterized layer. In case the points layer has a selection, only selected points will be used, but if the output extent is set to be automatically adjusted, the full extent of the layer will be used. That is, the extent of the layer is always considered to be the full extent of all features, not the one computed from just the selected ones. We will try to fix that by using some additional tools into our model.

Open the modeler and start the model by adding the required inputs. In this case we need a vector layer (restricted to points) and an attribute from it, with the values that we will use for rasterizing.



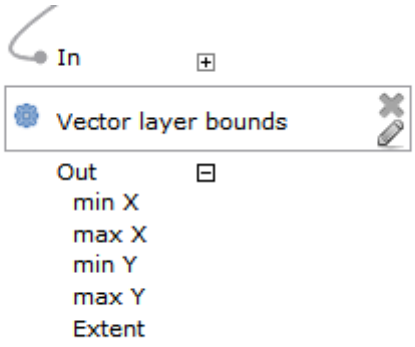
The next step is to compute the extent of the selected features. That's where we can use the model-only tool called *Vector layer bounds*. First, we will have to create a layer that has the extent of those selected features. Then, we can use this tool on that layer.

An easy way of creating a layer with the extent of the selected features is to compute a convex hull of the input points layer. It will use only the selected point, so the convex hull will have the same bounding box as the selection. Then we can add the *Vector layer bounds* algorithm, and use the convex hull layer as input. It should look this in the modeler canvas:



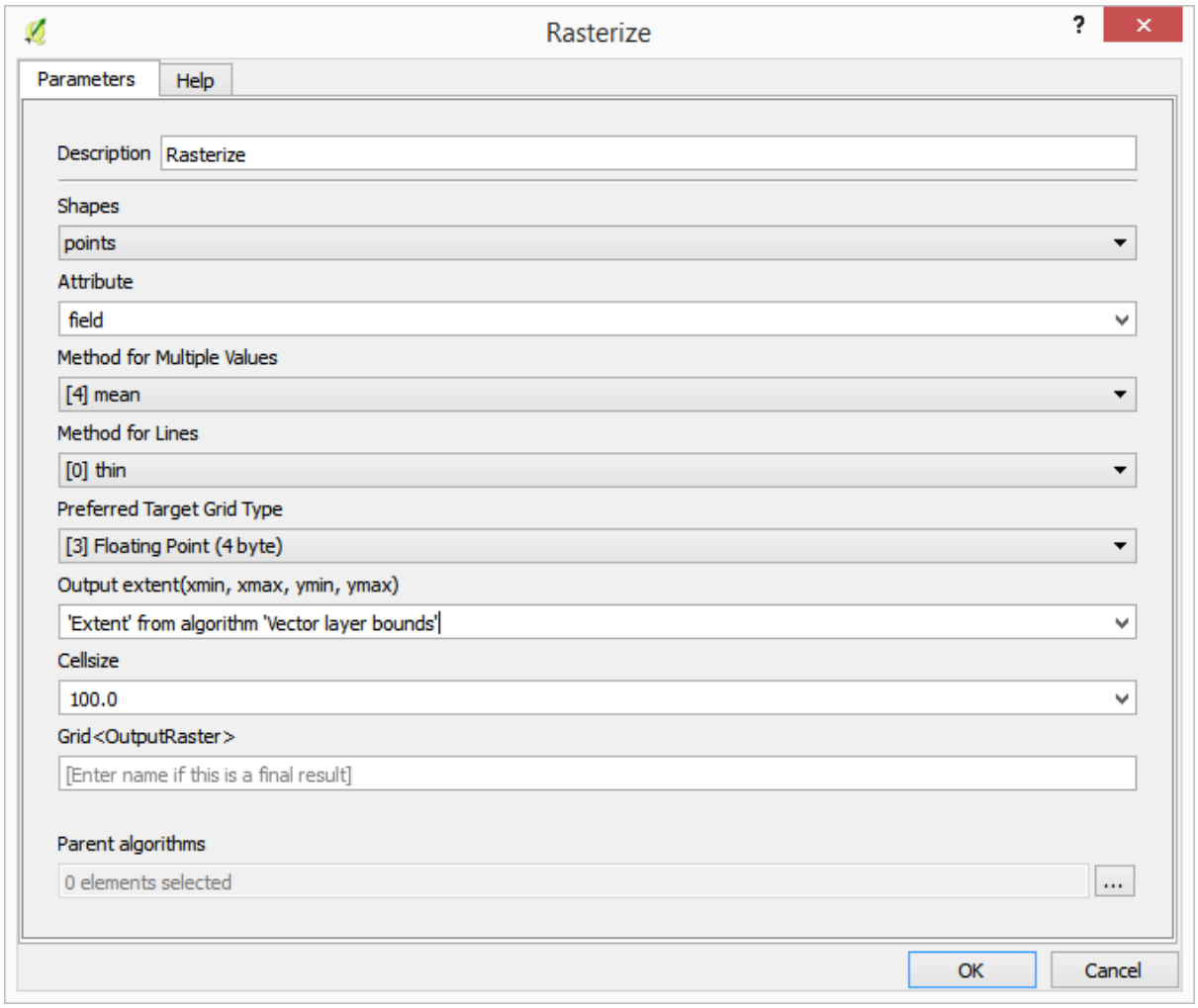
The result from the *Vector layer bounds* is a set of four numeric values and a extent object. We will use both the

numeric outputs and the extent for this exercise.

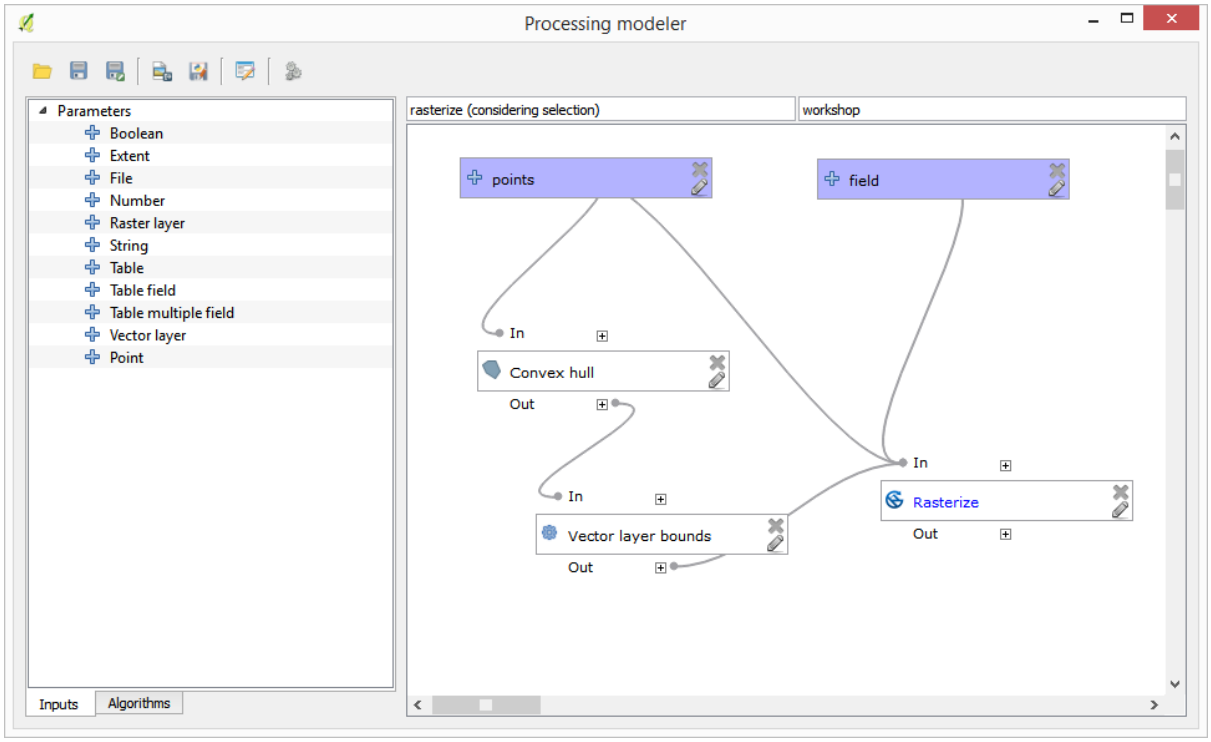


We can now add the algorithm that rasterizes the vector layer, using the extent from the *Vector layer bounds* algorithm as input.

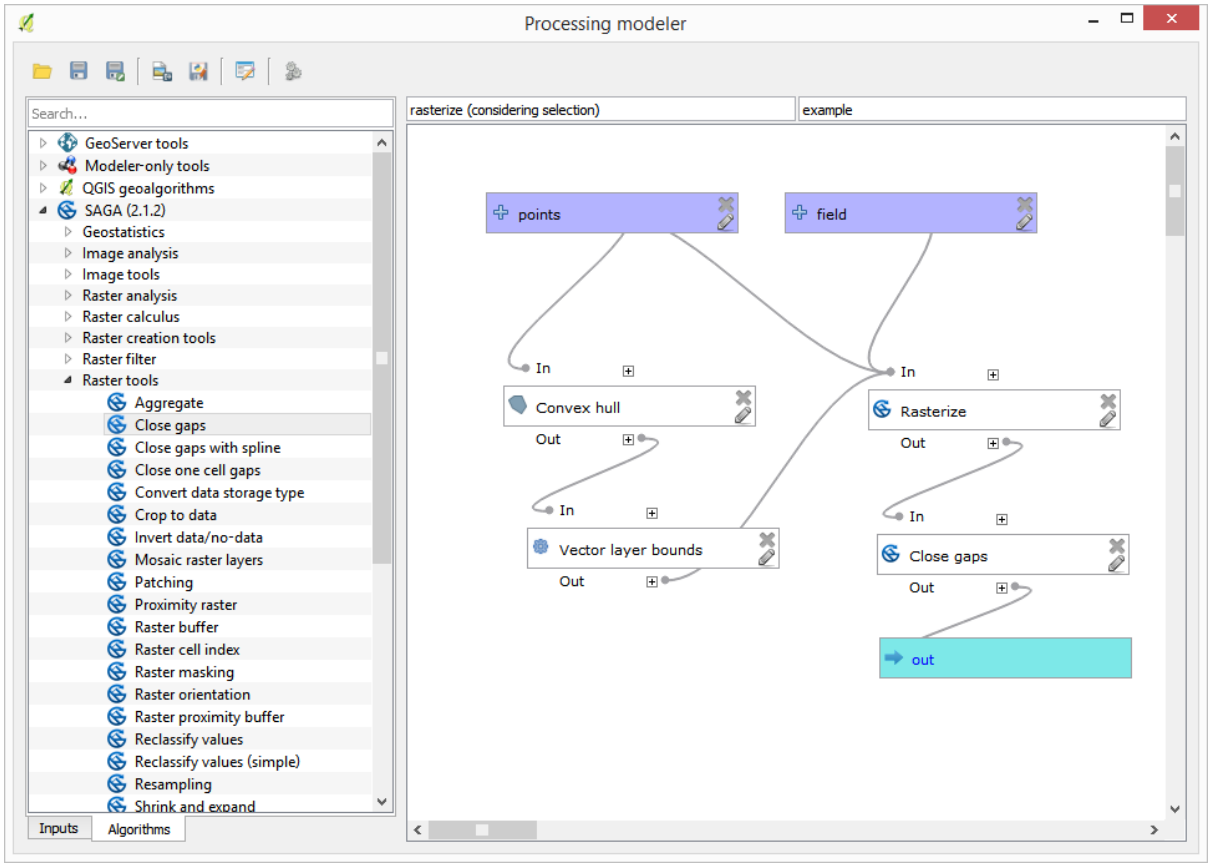
Fill the parameters of the algorithm as shown next:



The canvas should now look like.



Finally, fill the no-data values of the raster layer using the *Close gaps* algorithm.

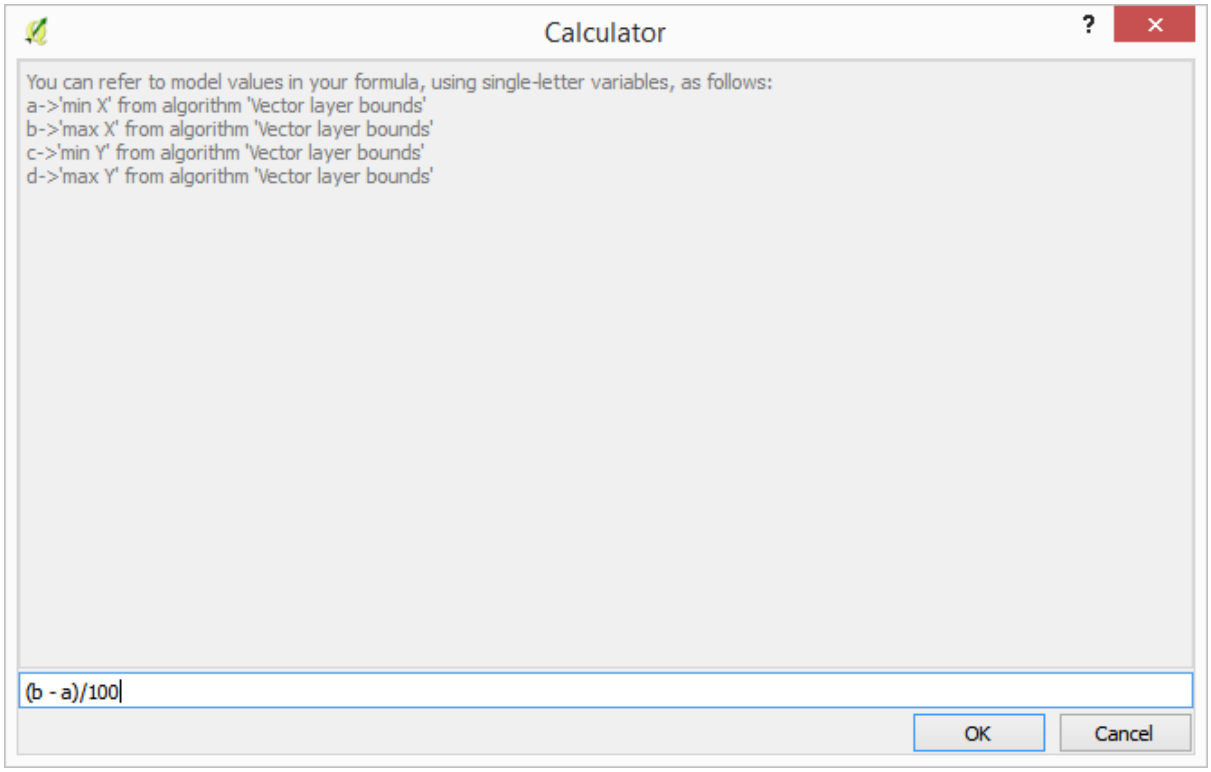


The algorithm is now ready to be saved and added to the toolbox. You can run it and it will generate a raster layer from interpolating the selected points in the input layer, and the layer will have the same extent as the selection.

Here's an improvement to the algorithm. We have used a harcoded value for the cellsize when rasterizing. This value

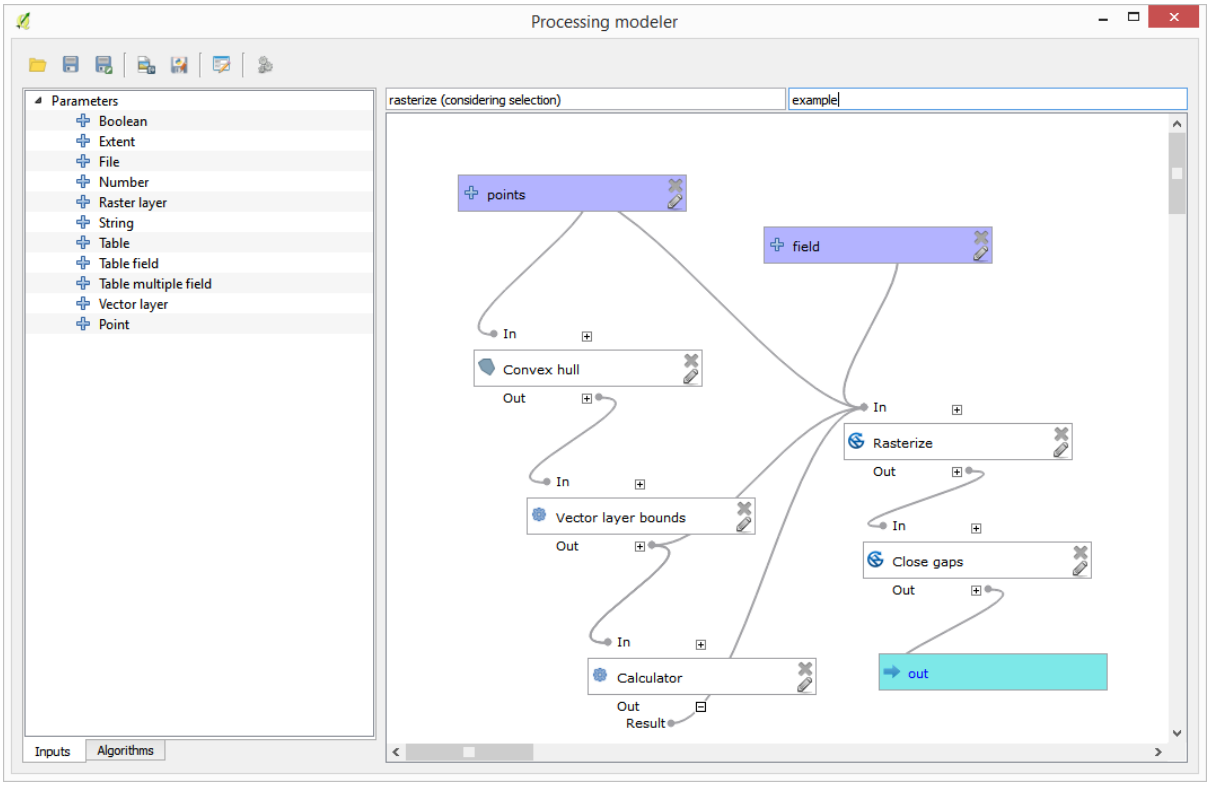
is fine for our test input layer, but might not be for other cases. We could add a new parameter, so the user enters the desired value, but a much better approach would be to have that value automatically computed.

We can use the modeler-only calculator, and compute that value from the extent coordinates. For instance, to create a layer with a fixed width of 100 pixels, we can use the following formula in the calculator.



Now we have to edit the rasterize algorithm, so it uses the output of the calculator instead of the hardcoded value.

The final algorithm should look like this:

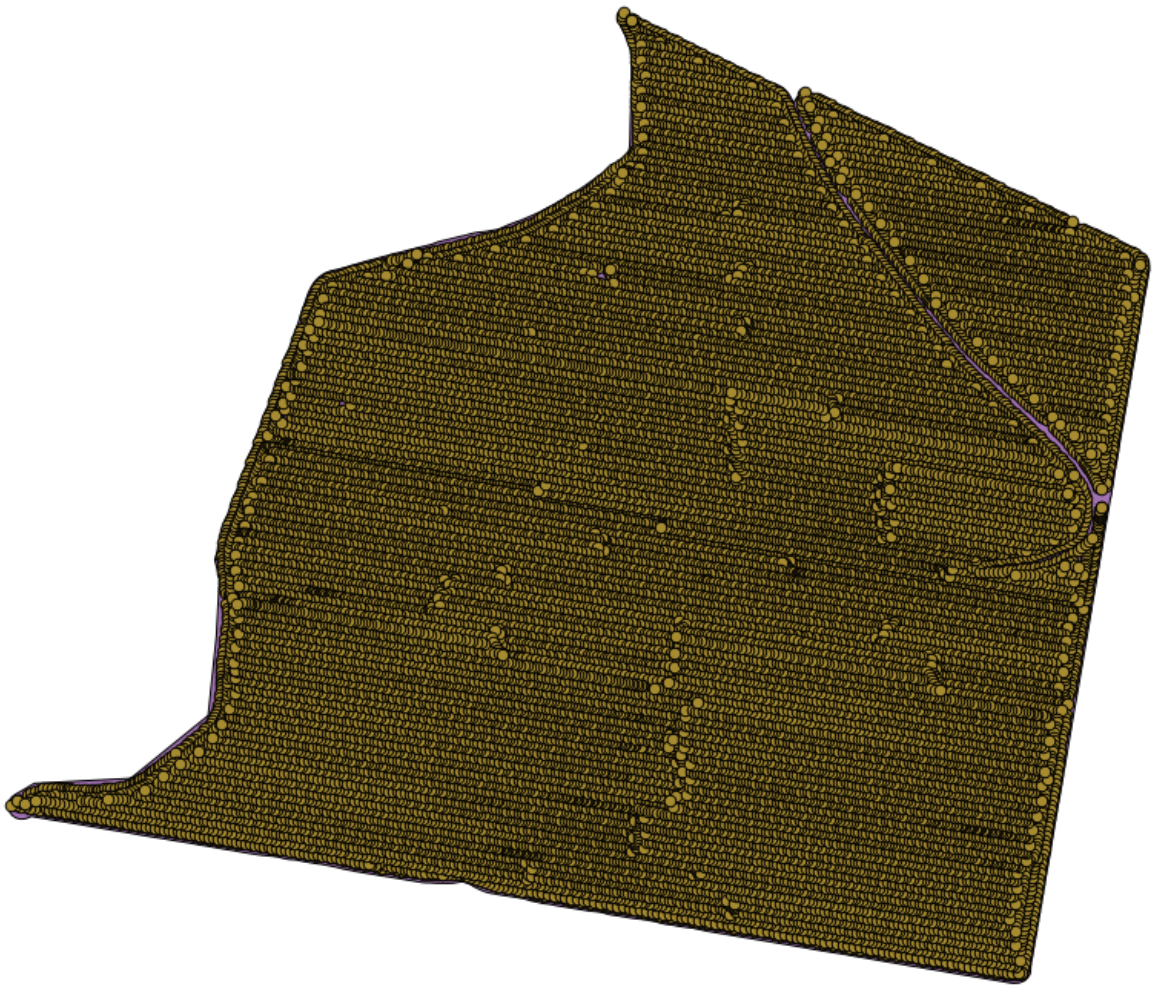


17.22 Interpolation

Примечание: This chapter shows how to interpolate point data, and will show you another real example of performing spatial analysis

In this lesson, we are going to interpolate points data to obtain a raster layer. Before doing it, we will have to do some data preparation, and after interpolating we will add some extra processing to modify the resulting layer, so we will have a complete analysis routine.

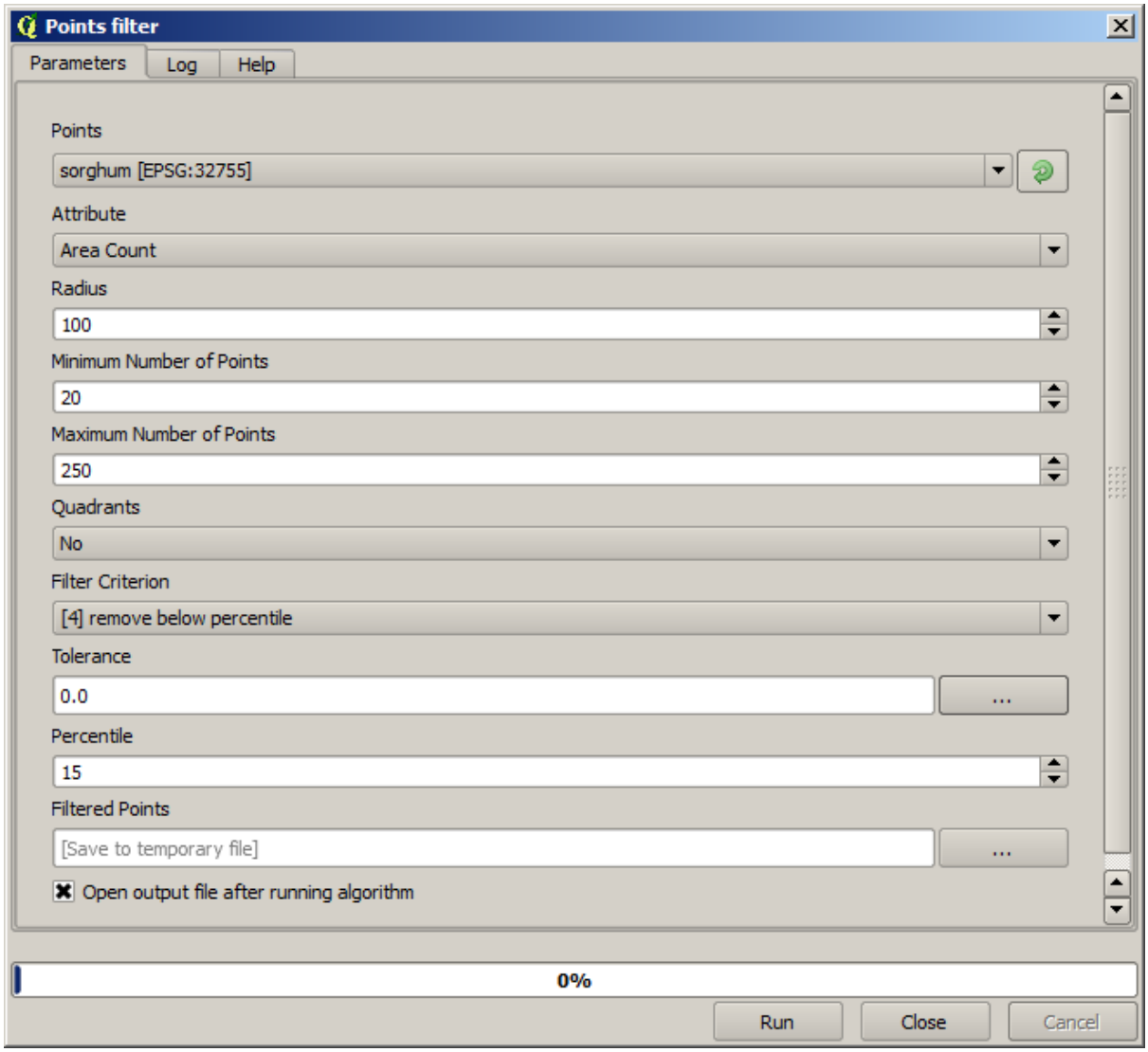
Open the example data for this lesson, which should look like this.



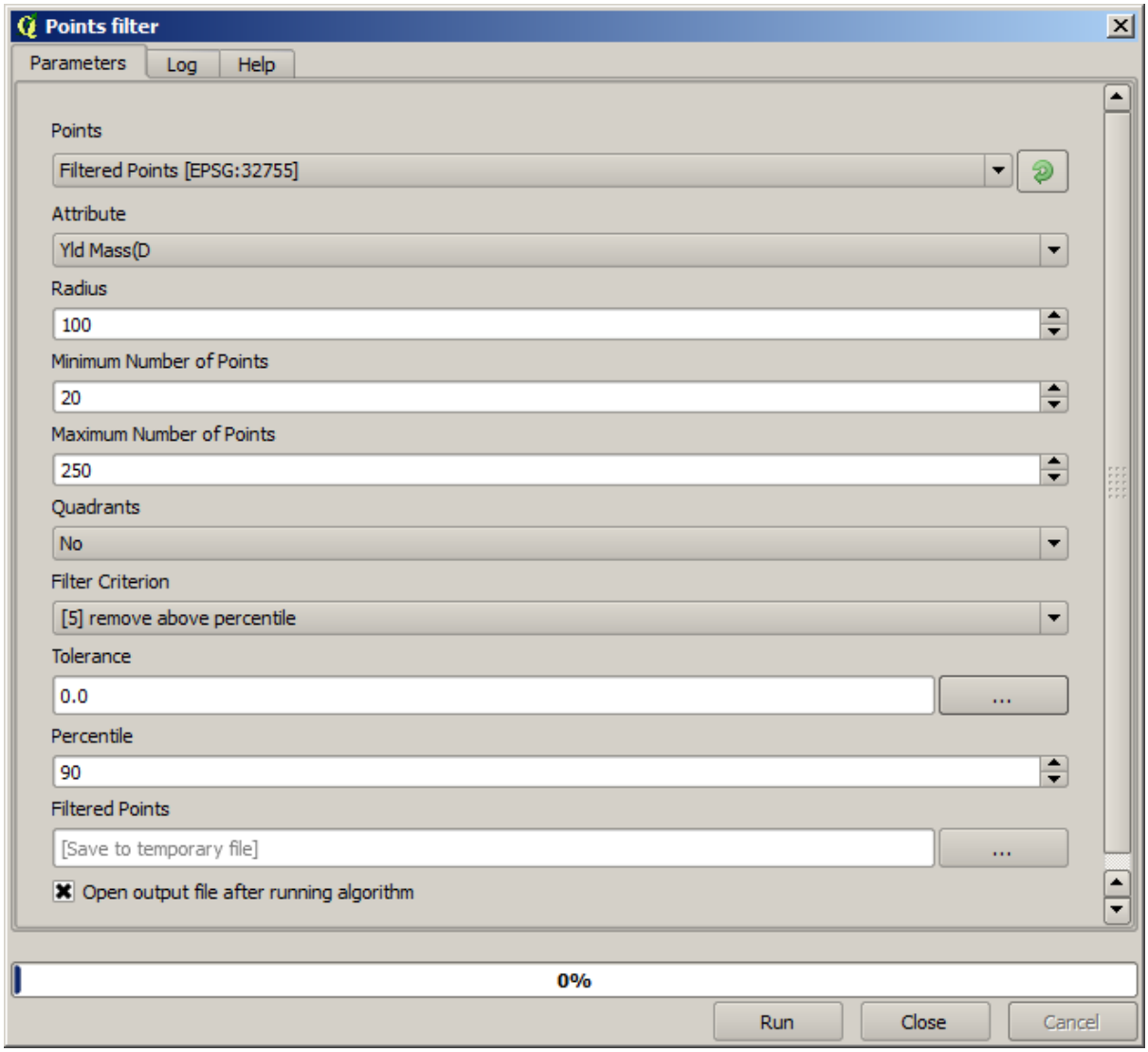
The data correspond to crop yield data, as produced by a modern harvester, and we will use it to get a raster layer of crop yield. We do not plan to do any further analysis with that layer, but just to use it as a background layer for easily identifying the most productive areas and also those where productivity can be improved.

The first thing to do is to clean–up the layer, since it contains redundant points. These are caused by the movement of the harvester, in places where it has to do a turn or it changes its speed for some reason. The *Points filter* algorithm will be useful for this. We will use it twice, to remove points that can be considered outliers both in the upper and lower part of the distribution.

For the first execution, use the following parameter values.



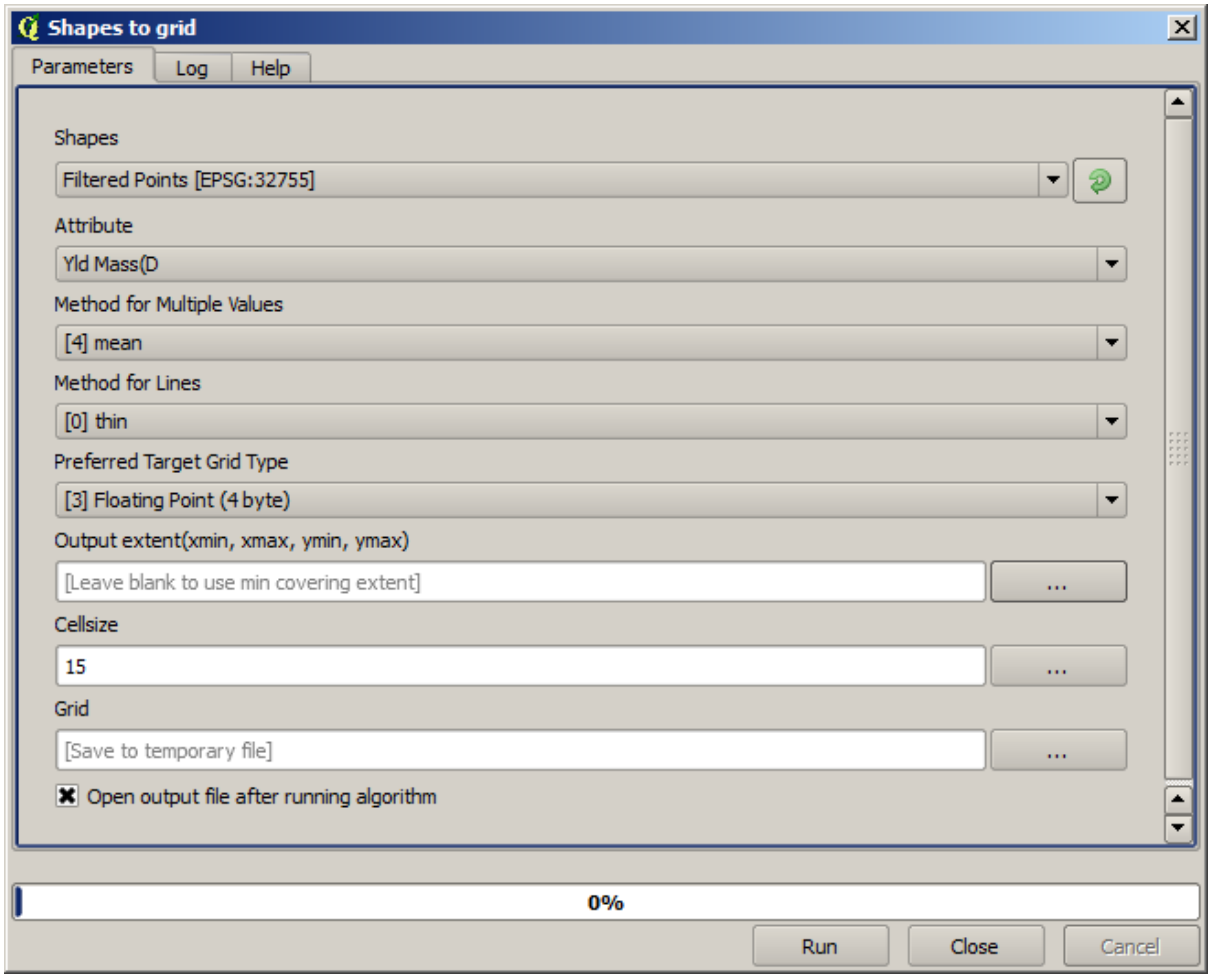
Now for the next one, use the configuration shown below.



Notice that we are not using the original layer as input, but the output of the previous run instead.

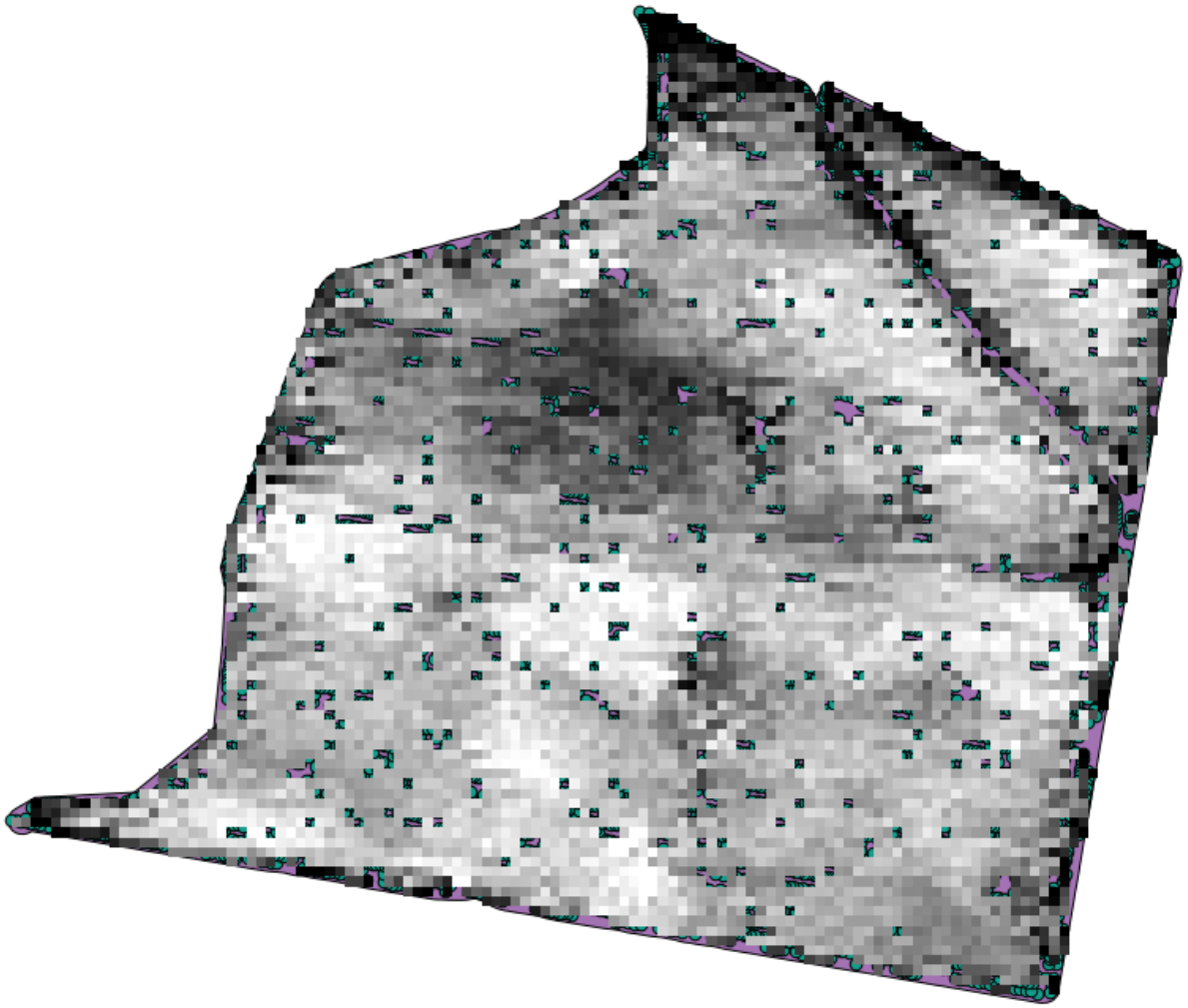
The final filter layer, with a reduced set of points, should look similar to the original one, but it contains a smaller number of points. You can check that by comparing their attribute tables.

Now let's rasterize the layer using the *Rasterize* algorithm.

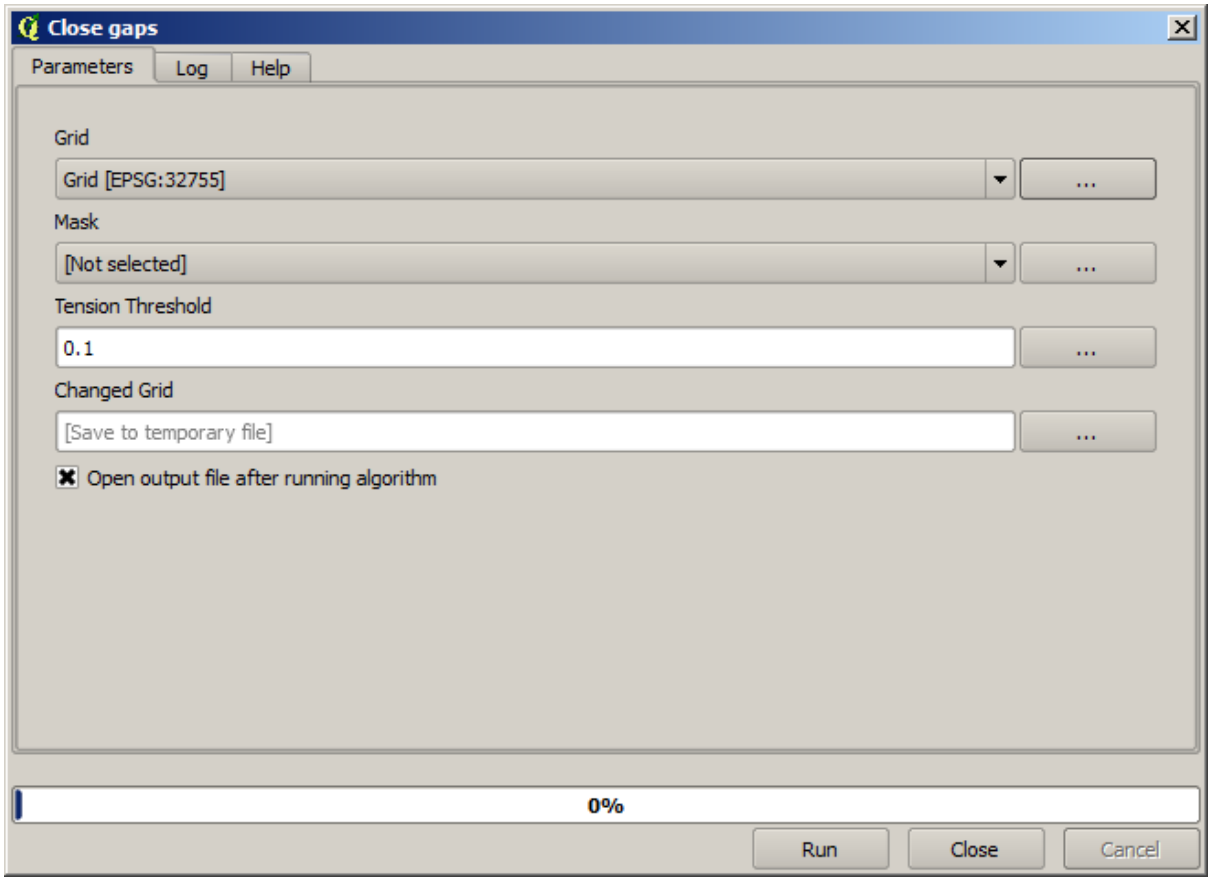


The *Filtered points* layer refers to the resulting one of the second filter. It has the same name as the one produced by the first filter, since the name is assigned by the algorithm, but you should not use the first one. Since we will not be using it for anything else, you can safely remove it from your project to avoid confusion, and leave just the last filtered layer.

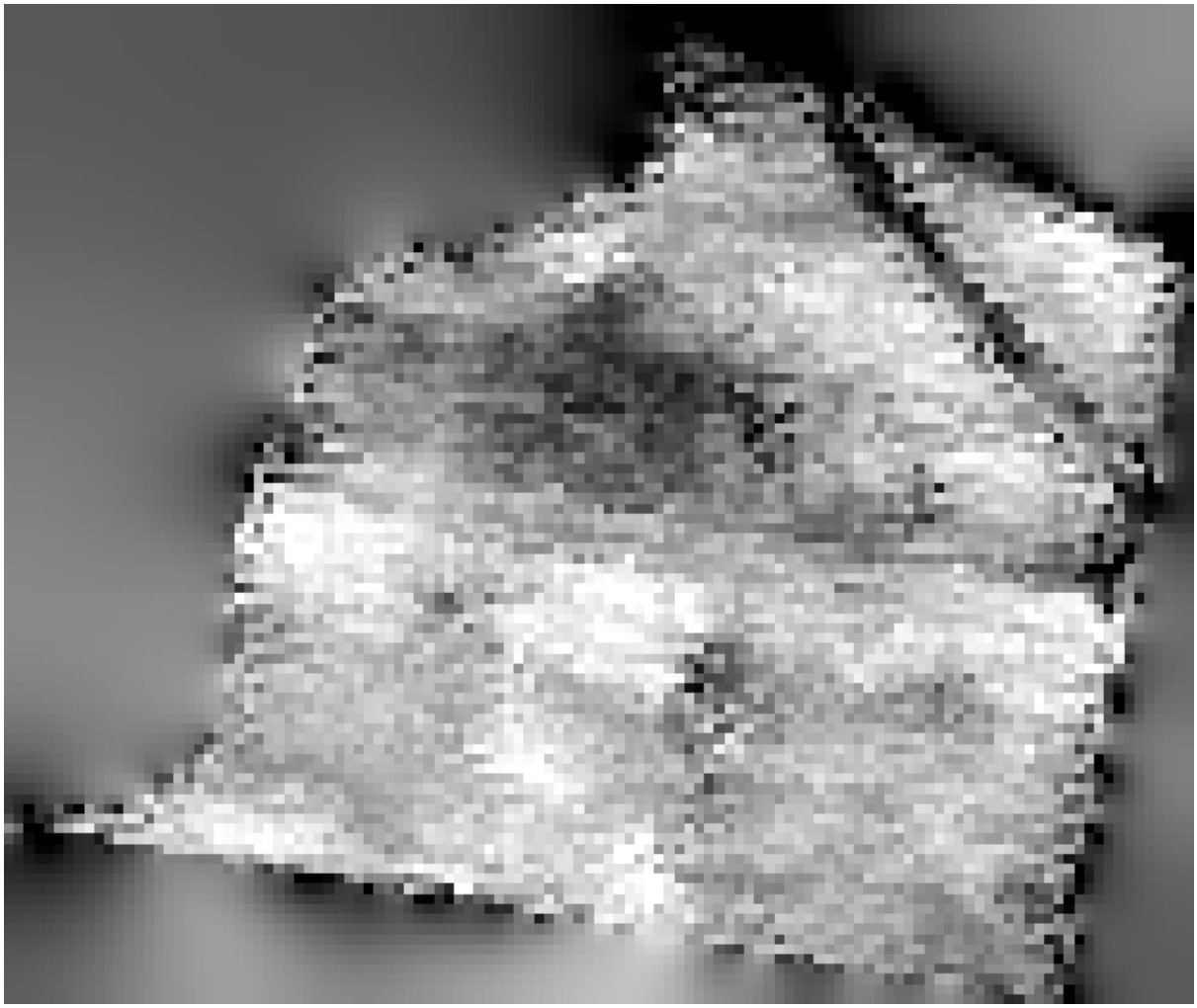
The resulting raster layer looks like this.



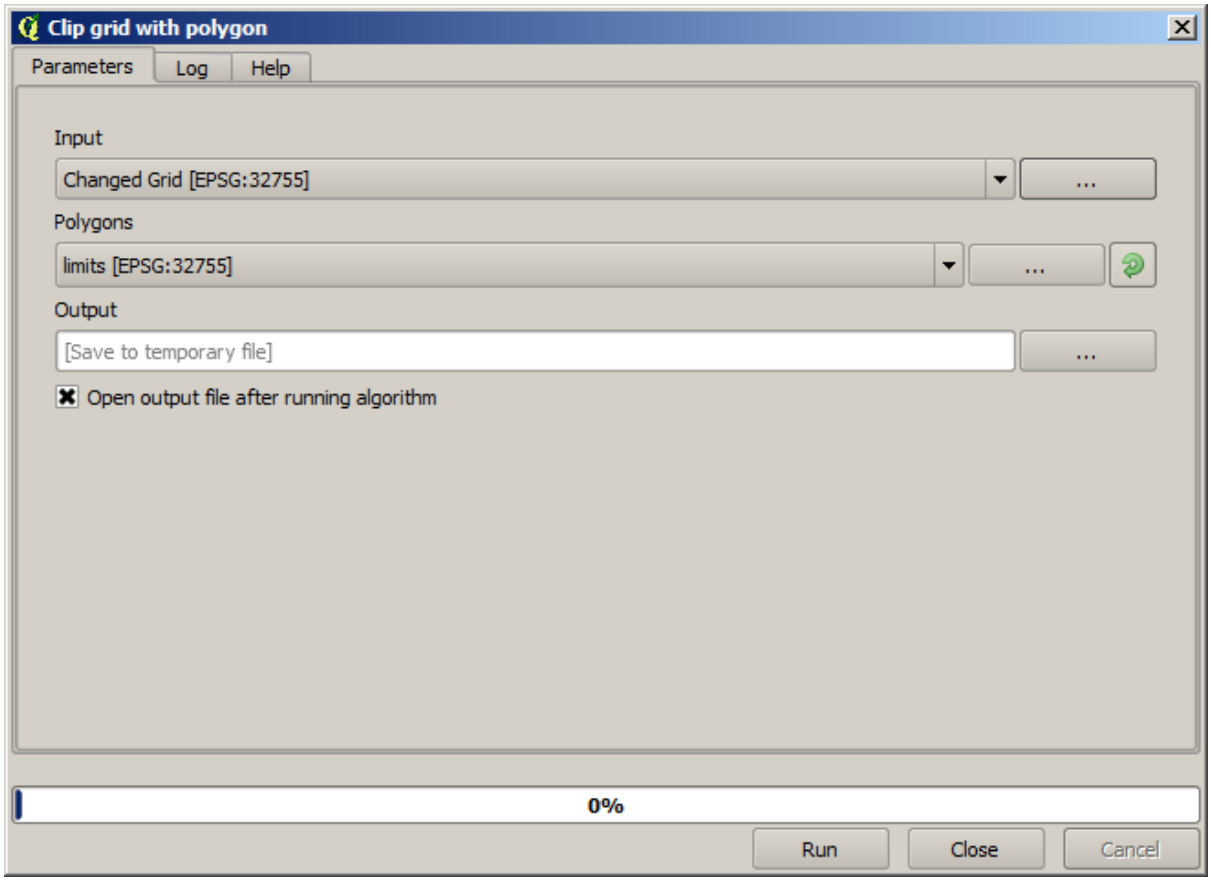
It is already a raster layer, but it is missing data in some of its cells. It only contain valid values in those cells that contained a point from the vector layer that we have just rasterized, and a no–data value in all the other ones. To fill the missing values, we can use the *Close gaps* algorithm.



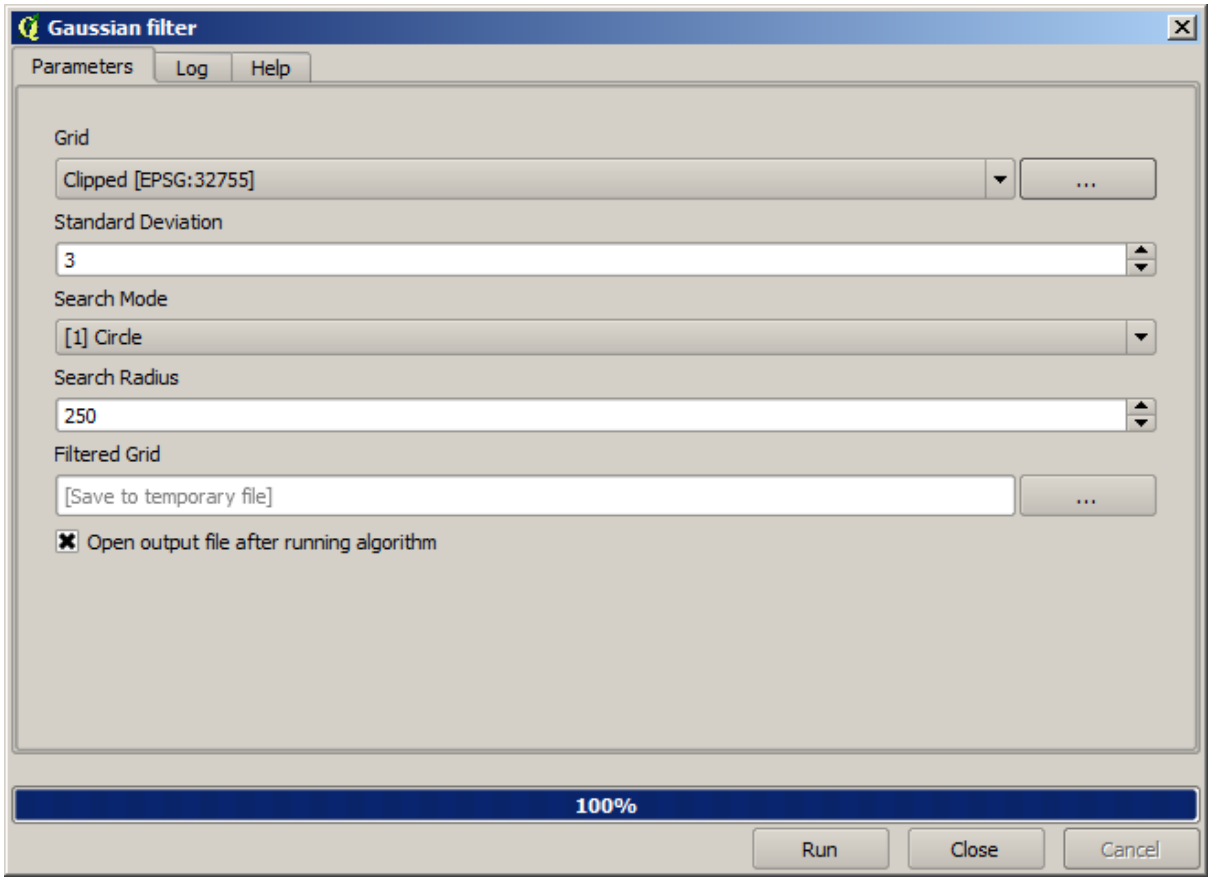
The layer without no–data values looks like this.



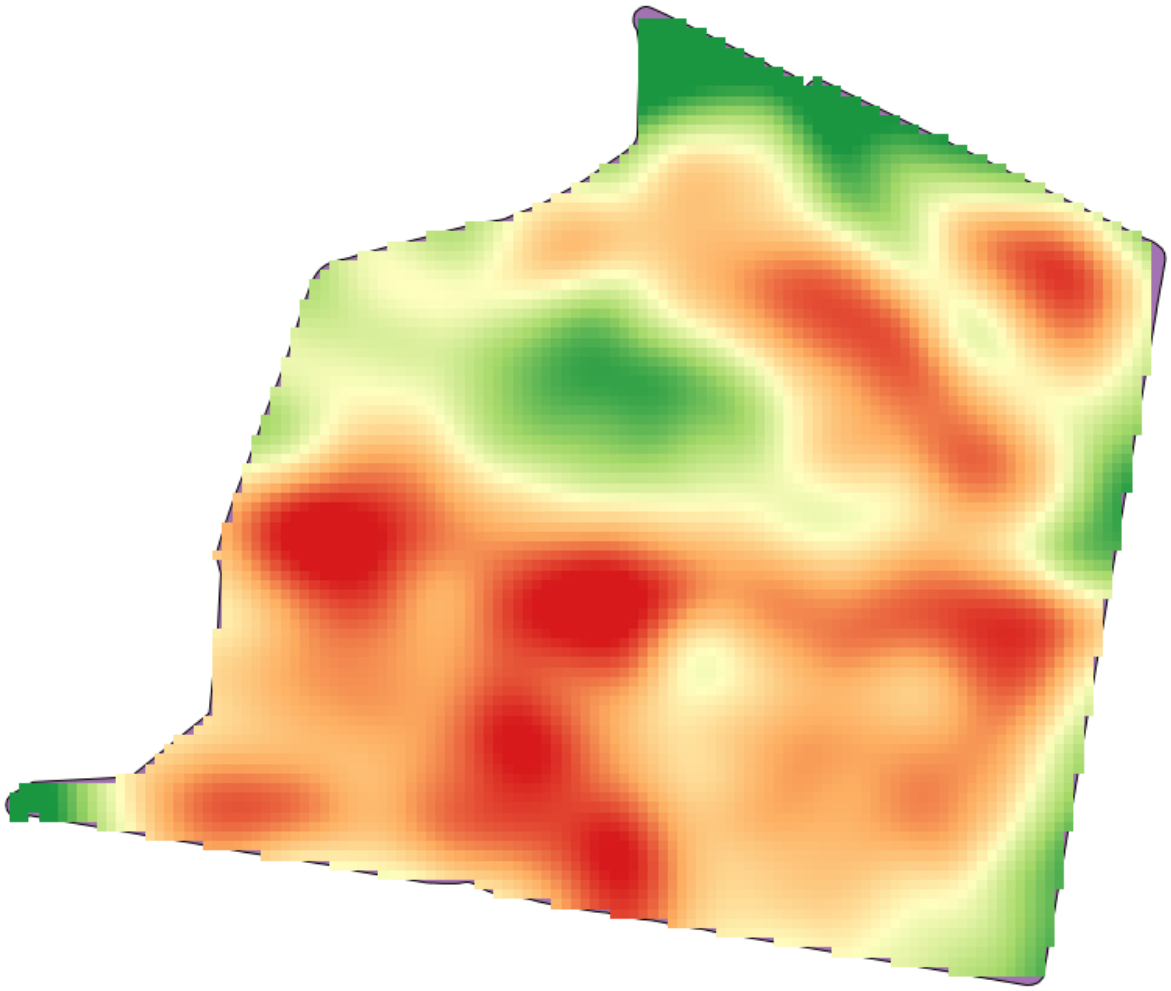
To restrict the area covered by the data to just the region where crop yield was measured, we can clip the raster layer with the provided limits layer.



And for a smoother result (less accurate but better for rendering in the background as a support layer), we can apply a *Gaussian filter* to the layer.



With the above parameters you will get the following result



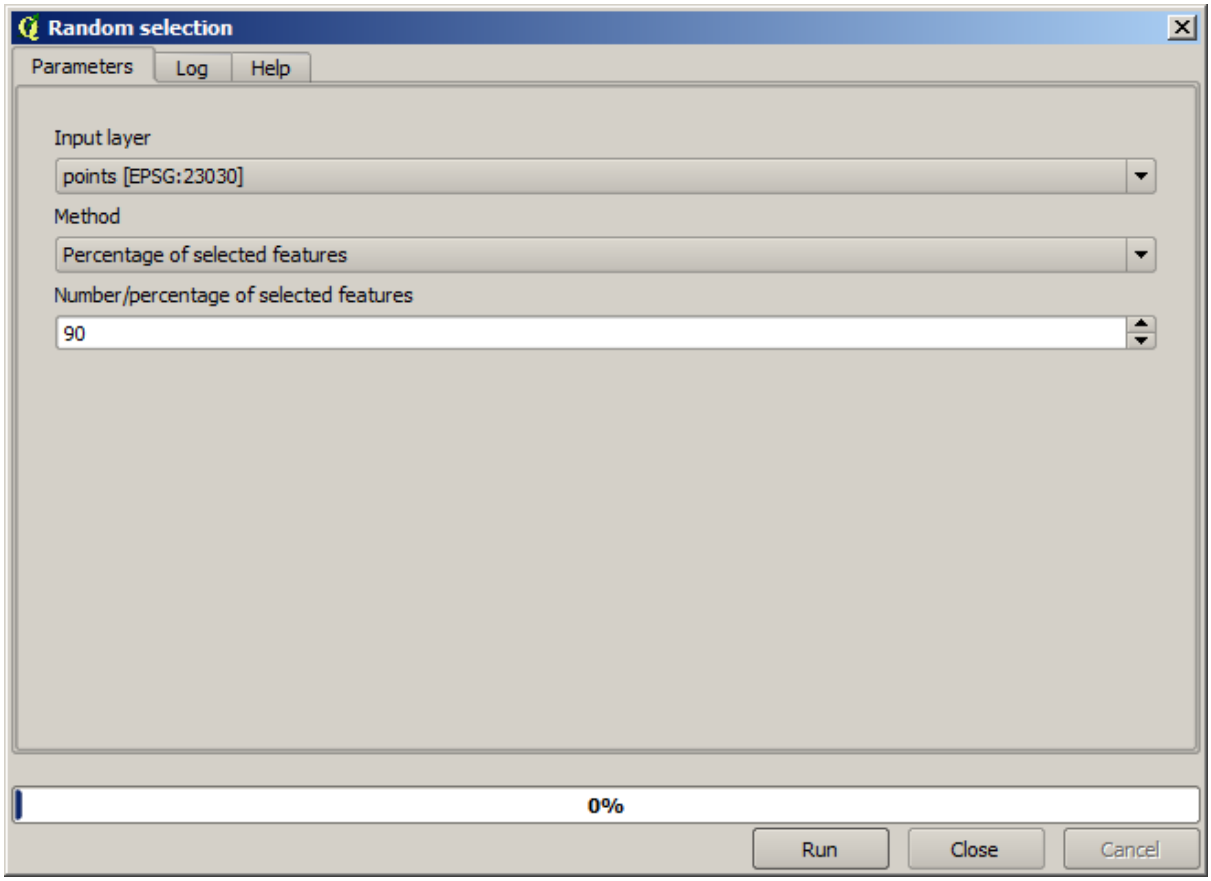
17.23 More interpolation

Примечание: This chapter shows another practical case where interpolation algorithms are used.

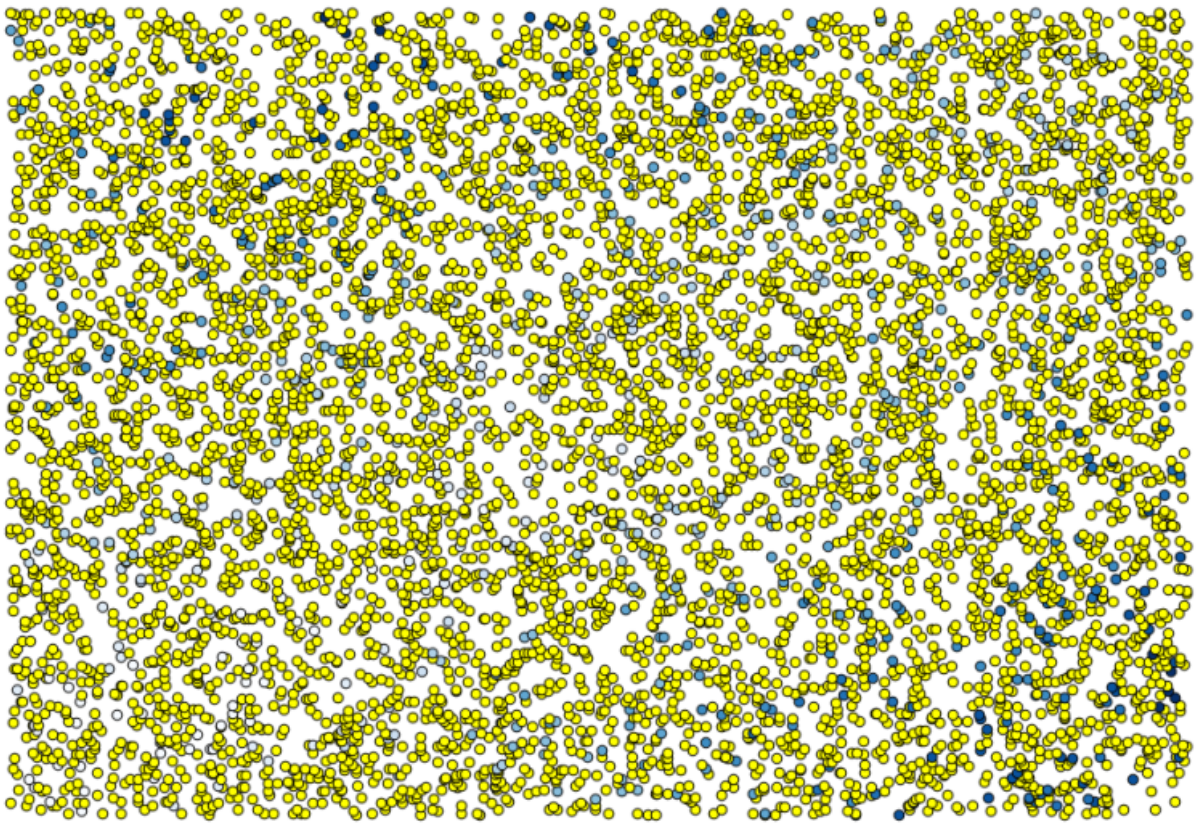
Interpolation is a common technique, and it can be used to demonstrate several techniques that can be applied using the QGIS processing framework. This lesson uses some interpolation algorithms that were already introduced, but has a different approach.

The data for this lesson contains also a points layer, in this case with elevation data. We are going to interpolate it much in the same way as we did in the previous lesson, but this time we will save part of the original data to use it for assessing the quality of the interpolation process.

First, we have to rasterize the points layer and fill the resulting no–data cells, but using just a fraction of the points in the layer. We will save 10% of the points for a later check, so we need to have 90% of the points ready for the interpolation. To do so, we could use the *Split shapes layer randomly* algorithm, which we have already used in a previous lesson, but there is a better way to do that, without having to create any new intermediate layer. Instead of that, we can just select the points we want to use for the interpolation (the 90% fraction), and then run the algorithm. As we have already seen, the rasterizing algorithm will use only those selected points and ignore the rest. The selection can be done using the *Random selection* algorithm. Run it with the following parameters.

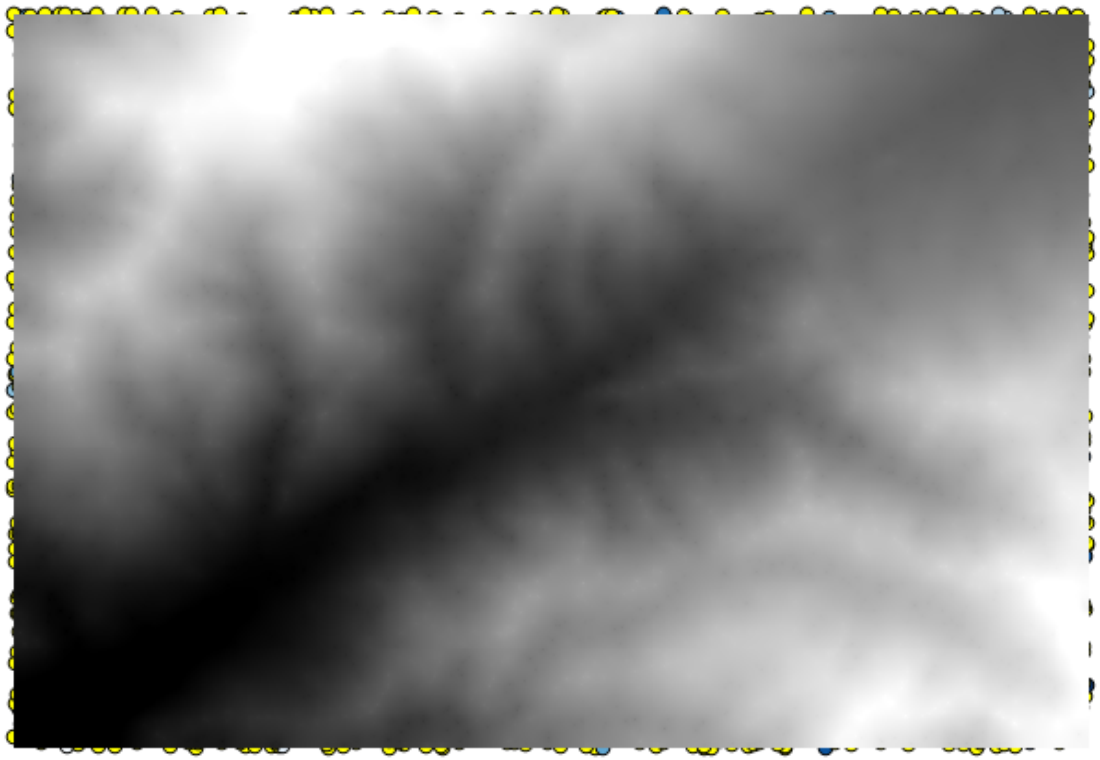


That will select 90% of the points in the layer to rasterize



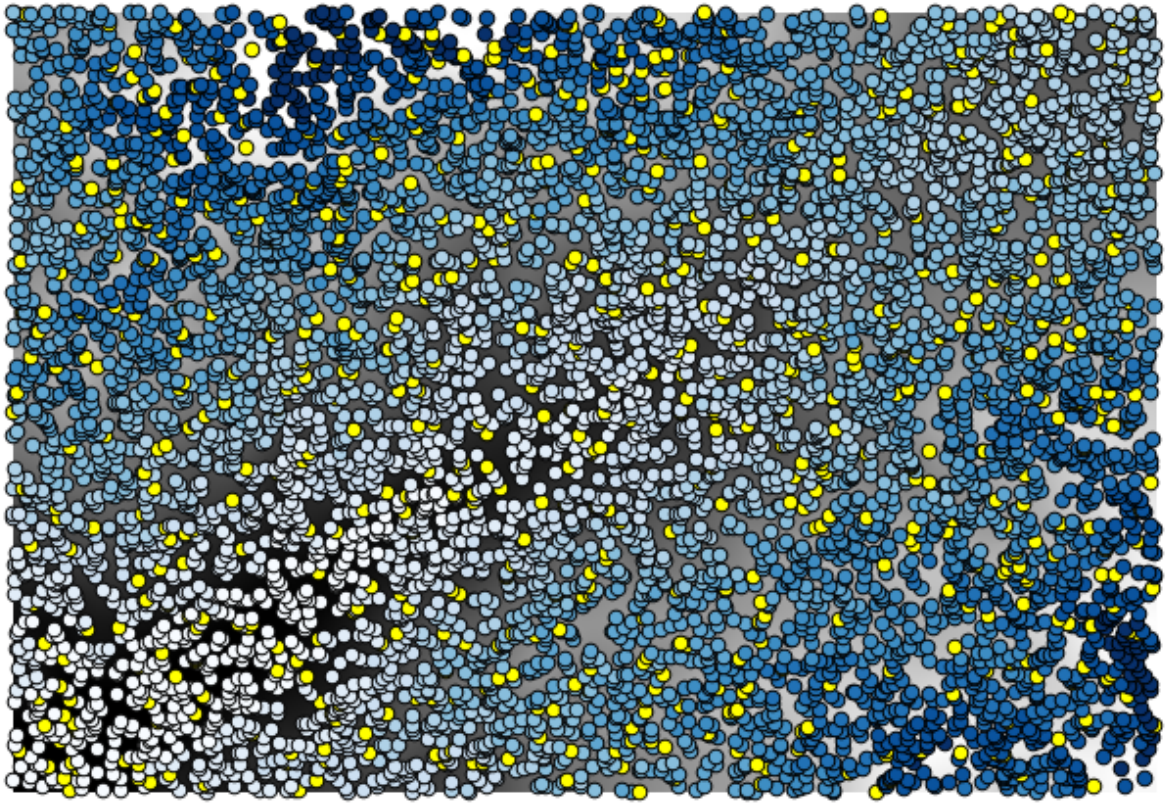
The selection is random, so your selection might differ from the selection shown in the above image.

Now run the *Rasterize* algorithm to get the first raster layer, and then run the *Close gaps* algorithm to fill the no–data cells [Cell resolution: 100 m].

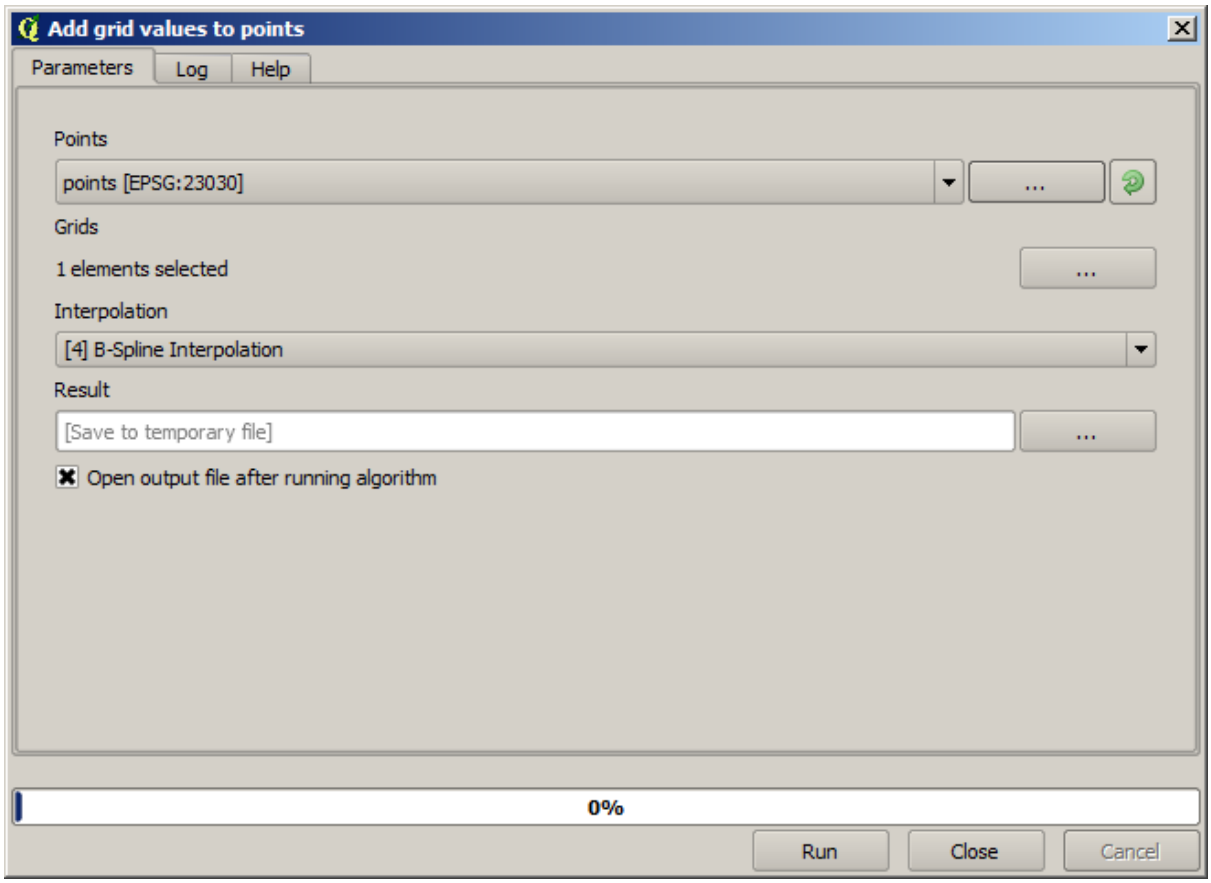


To check the quality of the interpolation, we can now use the points that are not selected. At this point, we know the real elevation (the value in the points layer) and the interpolated elevation (the value in the interpolated raster layer). We can compare the two by computing the differences between those values.

Since we are going to use the points that are not selected, first, let's invert the selection.

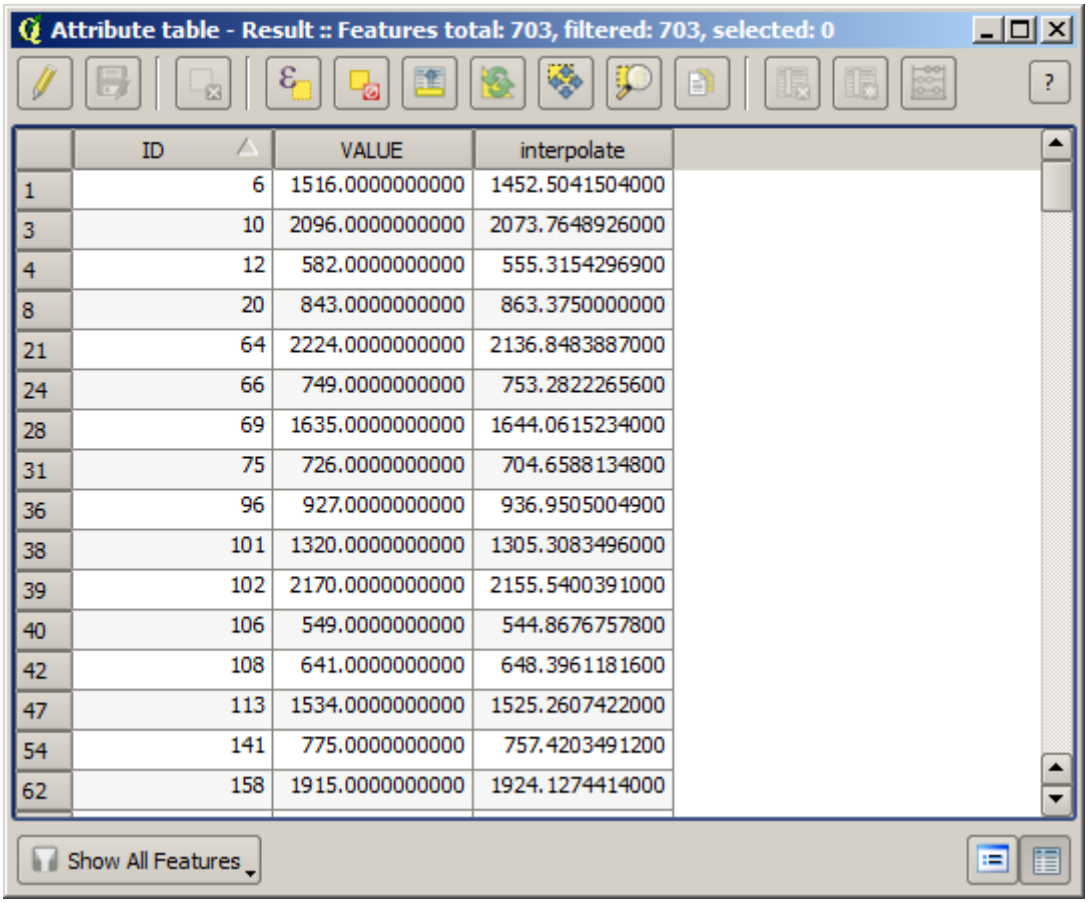


The points contain the original values, but not the interpolated ones. To add them in a new field, we can use the *Add raster values to points* algorithm



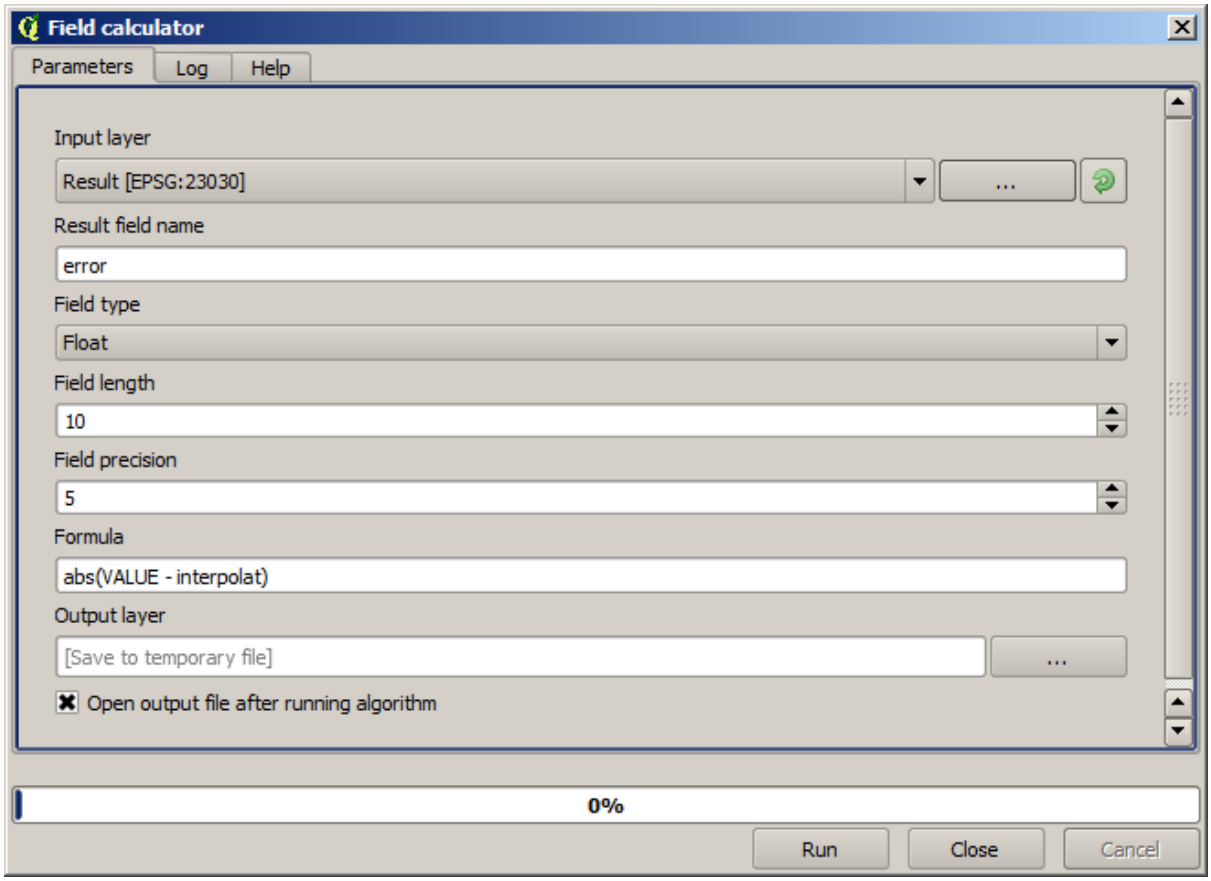
The raster layer to select (the algorithm supports multiple raster, but we just need one) is the resulting one from the interpolation. We have renamed it to *interpolate* and that layer name is the one that will be used for the name of the field to add.

Now we have a vector layer that contains both values, with points that were not used for the interpolation.



	ID	VALUE	interpolate
1	6	1516.0000000000	1452.5041504000
3	10	2096.0000000000	2073.7648926000
4	12	582.0000000000	555.3154296900
8	20	843.0000000000	863.3750000000
21	64	2224.0000000000	2136.8483887000
24	66	749.0000000000	753.2822265600
28	69	1635.0000000000	1644.0615234000
31	75	726.0000000000	704.6588134800
36	96	927.0000000000	936.9505004900
38	101	1320.0000000000	1305.3083496000
39	102	2170.0000000000	2155.5400391000
40	106	549.0000000000	544.8676757800
42	108	641.0000000000	648.3961181600
47	113	1534.0000000000	1525.2607422000
54	141	775.0000000000	757.4203491200
62	158	1915.0000000000	1924.1274414000

Now, we will use the fields calculator for this task. Open the *Field calculator* algorithm and run it with the following parameters.



If your field with the values from the raster layer has a different name, you should modify the above formula accordingly. Running this algorithm, you will get a new layer with just the points that we haven't used for the interpolation, each of them containing the difference between the two elevation values.

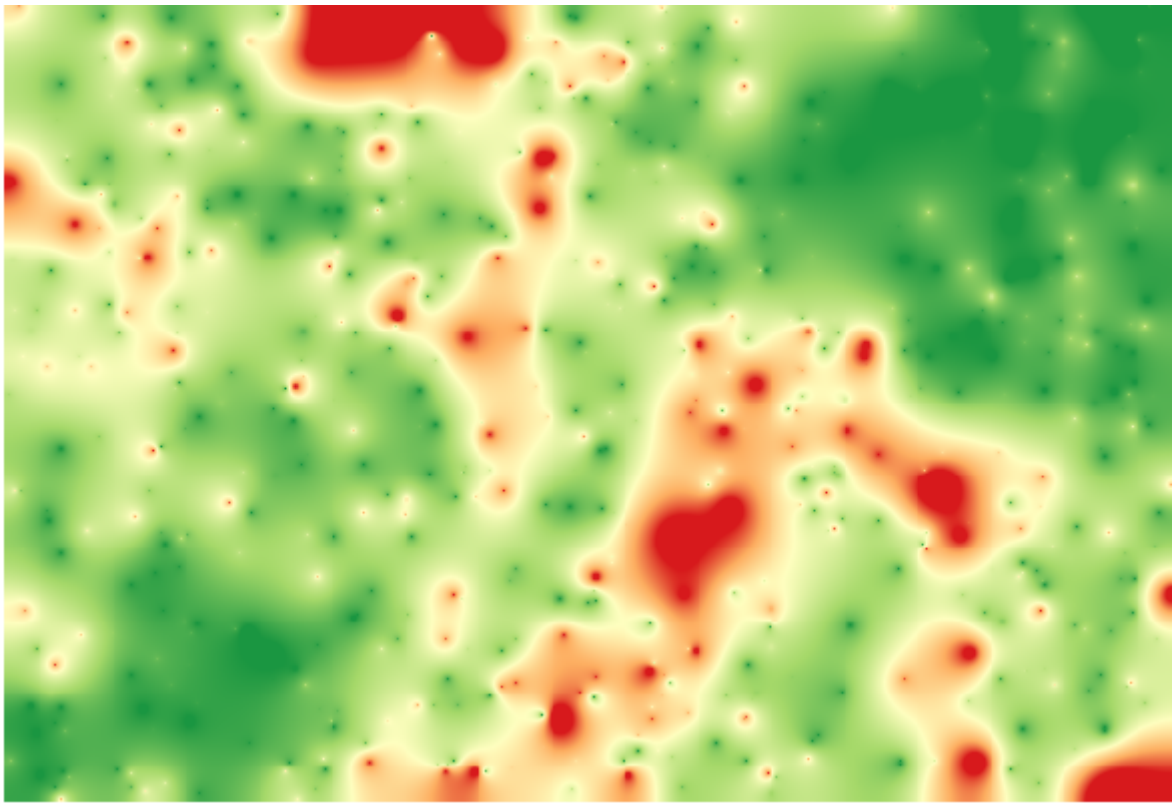
Representing that layer according to that value will give us a first idea of where the largest discrepancies are found.

Attribute table - Output layer :: Features total: 703, filtered: 703, selected: 0

	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

Show All Features

Interpolating that layer will get you a raster layer with the estimated error in all points of the interpolated area.



You can also get the same information (difference between original point values and interpolated ones) directly with *GRASS ► v.sample*.

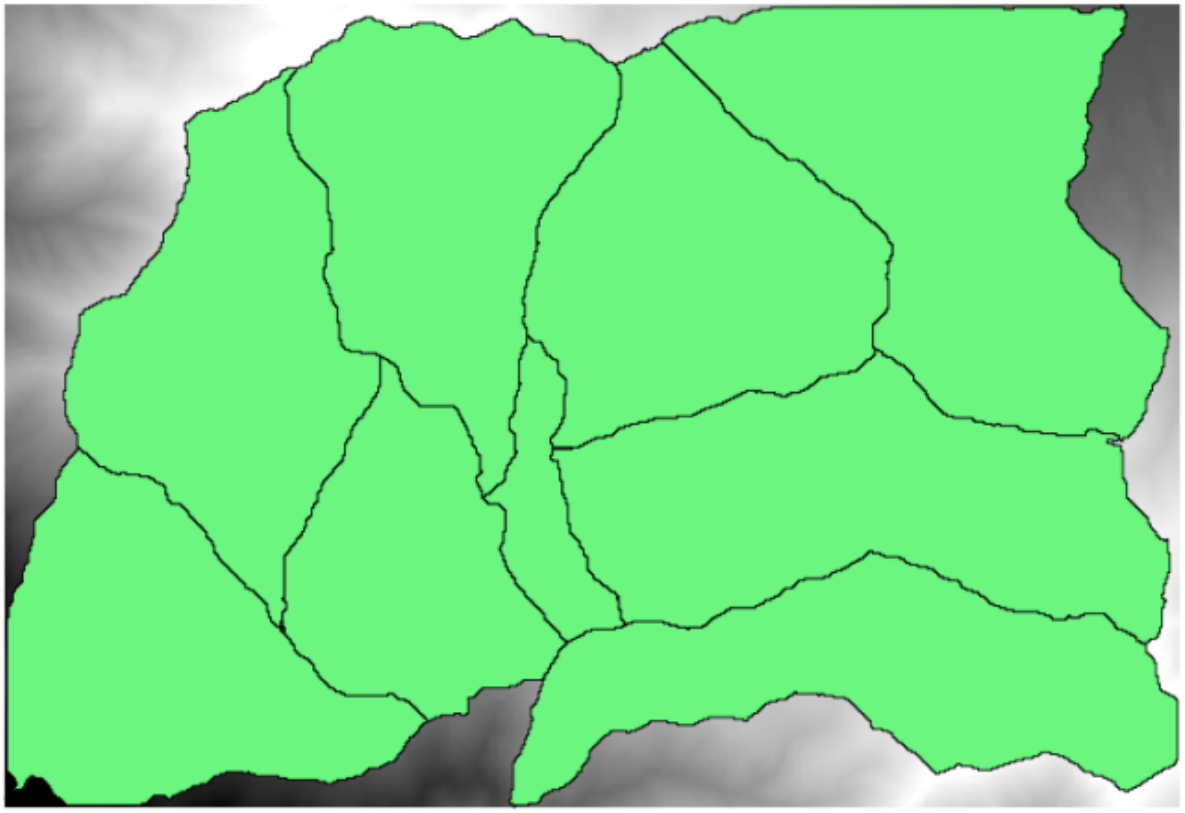
Your results might differ from these ones, since there is a random component introduced when running the random selection, at the beginning of this lesson.

17.24 Iterative execution of algorithms

Примечание: This lesson shows a different way of executing algorithms that use vector layers, by running them repeatedly, iterating over the features in an input vector layer

We already know the graphical modeler, which is one way of automating processing tasks. However, in some situations, the modeler might not be what we need to automate a given task. Let's see one of those situations and how to easily solve it using a different functionality: the iterative execution of algorithms.

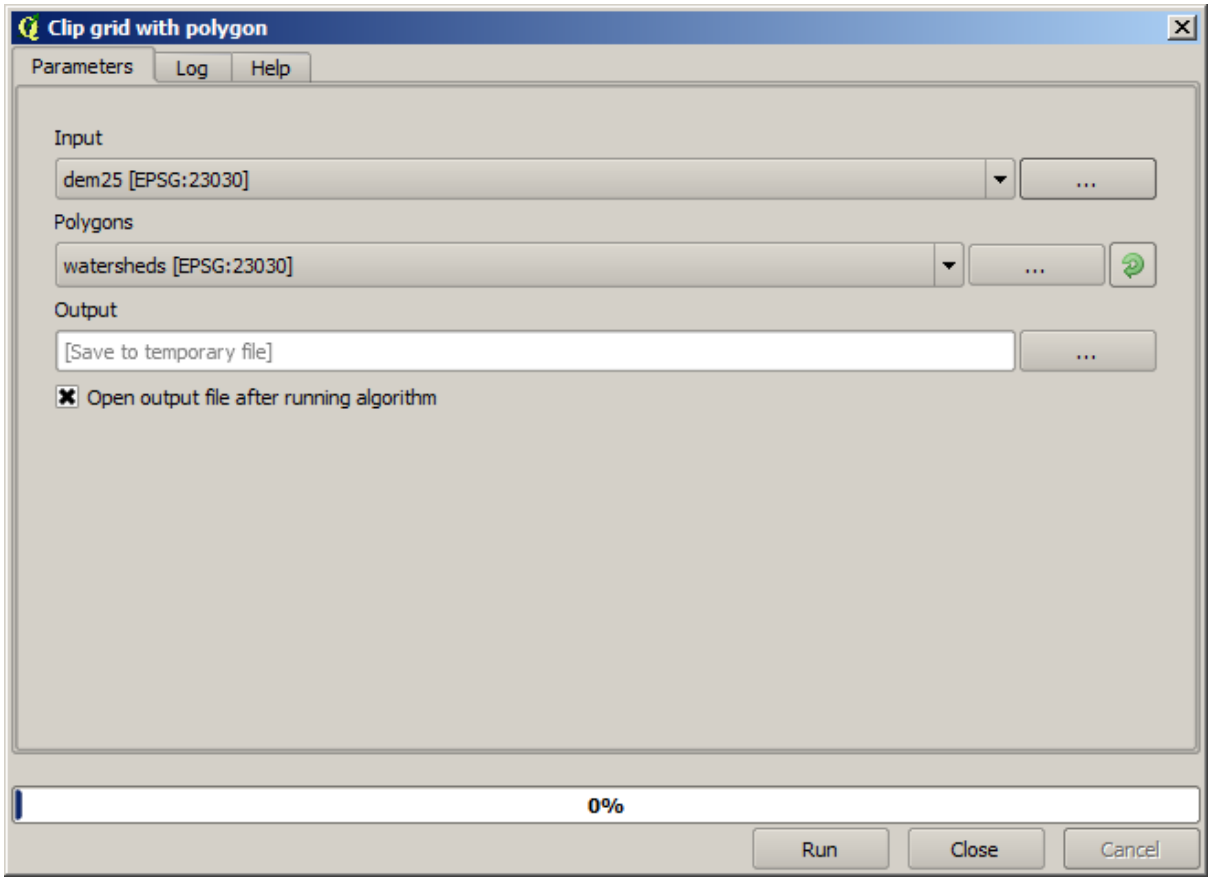
Open the data corresponding to this chapter. It should look like this.



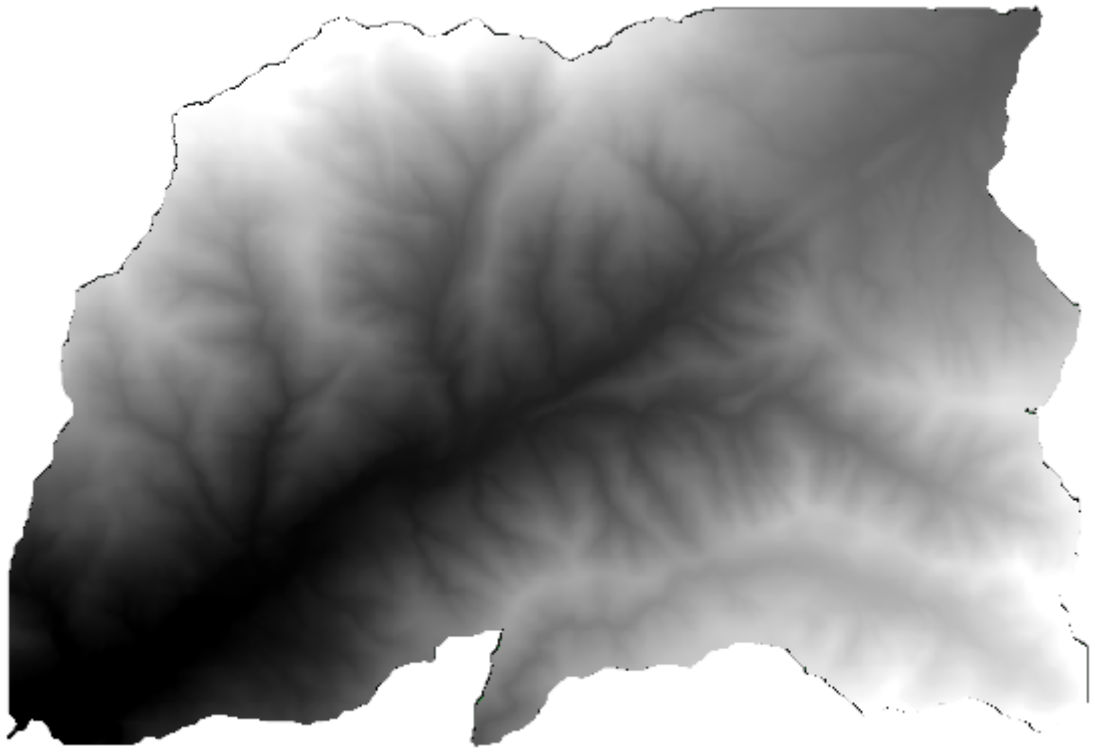
You will recognize our well-known DEM from previous chapters and a set of watersheds extracted from it. Imagine that you need to cut the DEM into several smaller layers, each of them containing just the elevation data corresponding to a single watershed. That will be useful if you later want to calculate some parameters related to each watershed, such as its mean elevation or it hypsographic curve.

This can be a lengthy and tedious task, especially if the number of watersheds is large. However, it is a task that can be easily automated, as we will see.

The algorithm to use for clipping a raster layer with a polygon layer is called *Clip raster with polygons*, and has the following parameters dialog.

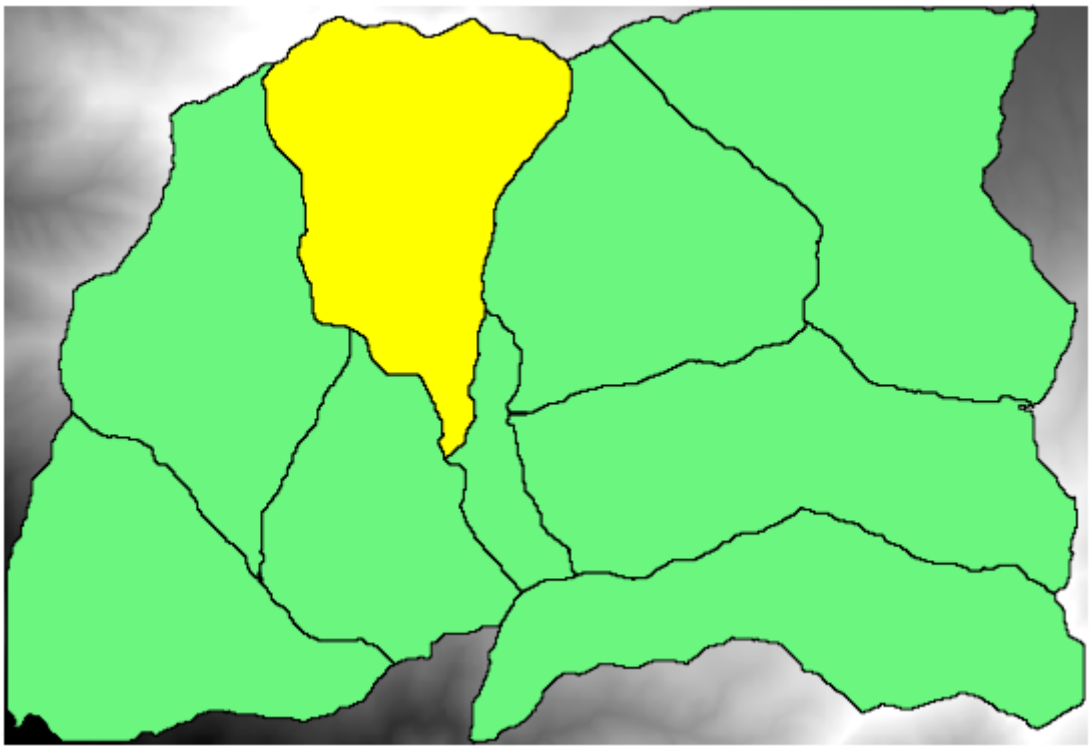


You can run it using the watersheds layer and the DEM as input, and you will get the following result.

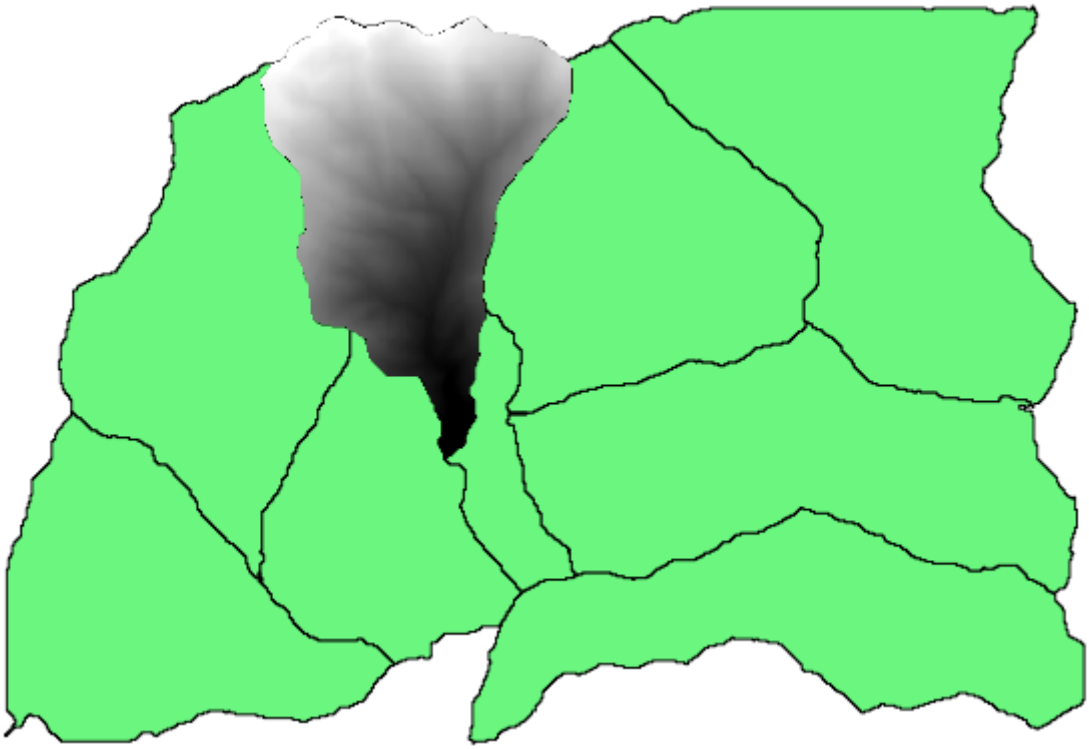


As you can see, the area covered by all the watershed polygons is used.

You can have the DEM clipped with just a single watershed by selecting the desired watershed and then running the algorithm as we did before.

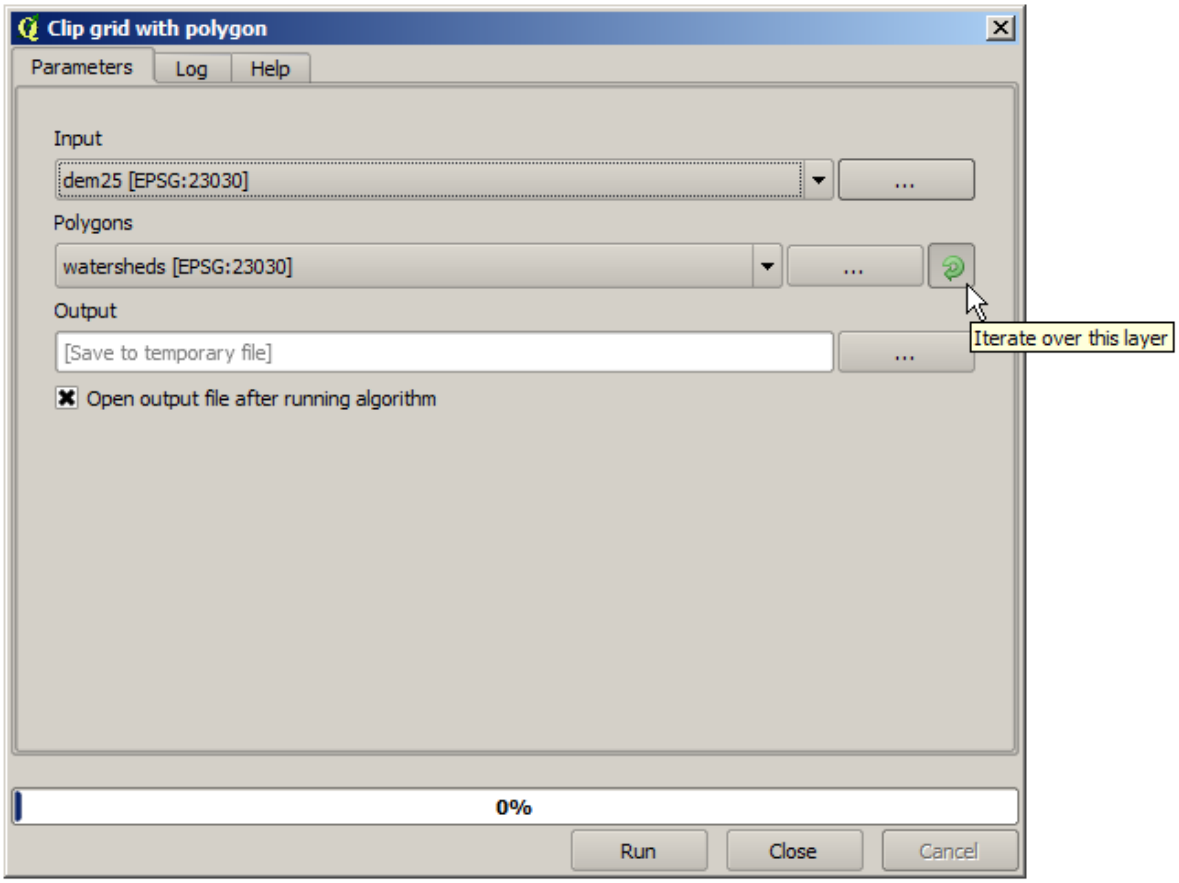


Since only selected features are used, only the selected polygon will be used to crop the raster layer.



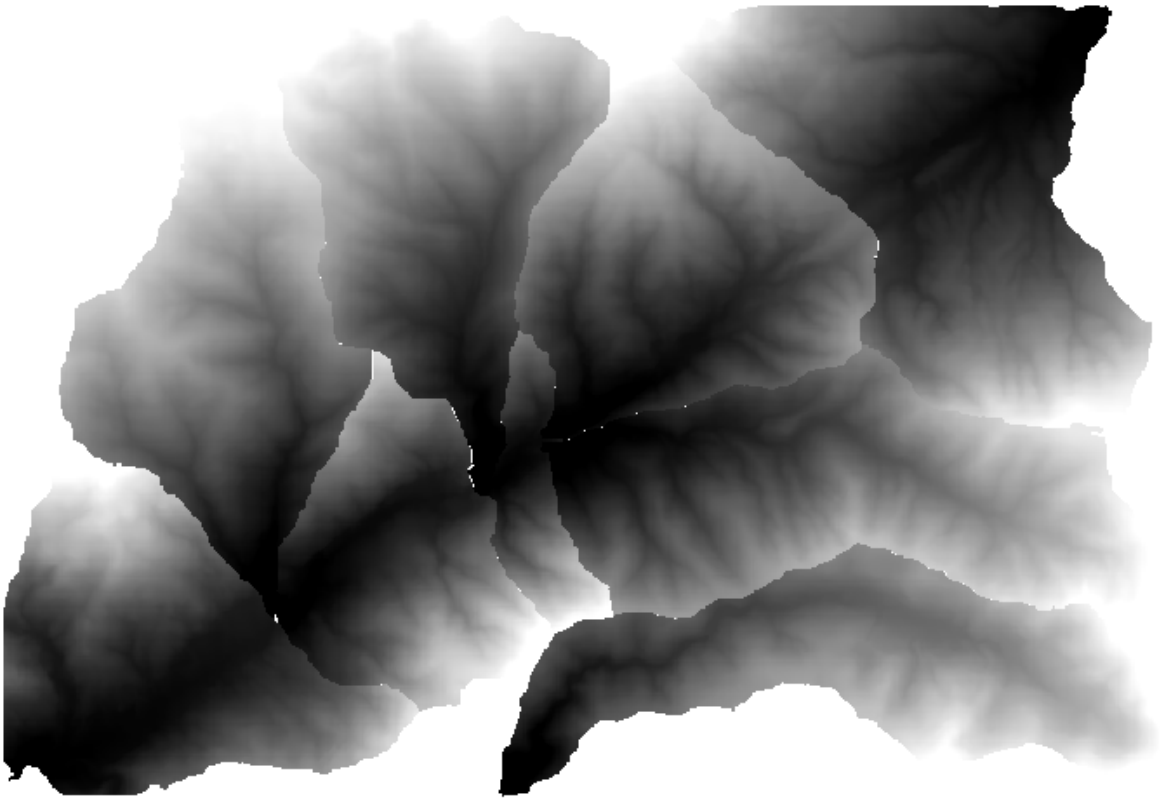
Doing this for all the watersheds will produce the result we are looking for, but it doesn't look like a very practical way of doing it. Instead, let's see how to automate that *select and crop* routine.

First of all, remove the previous selection, so all polygons will be used again. Now open the *Clip raster with polygon* algorithm and select the same inputs as before, but this time click on the button that you will find in the right–hand side of the vector layer input where you have selected the watersheds layer.



This button will cause the selected input layer to be split into as many layer as feature are found in it, each of them containing a single polygon. With that, the algorithm will be called repeatedly, one for each one of those single-polygon layers. The result, instead of just one raster layer in the case of this algorithm, will be a set of raster layers, each one of them corresponding to one of the executions of the algorithm.

Here's the result that you will get if you run the clipping algorithm as explained.



For each layer, the black and white color palette, (or whatever palette you are using), is adjusted differently, from its minimum to its maximum values. That's the reason why you can see the different pieces and the colors do not seem to match in the border between layers. Values, however, do match.

If you enter an output filename, resulting files will be named using that filename and a number corresponding to each iteration as suffix.

17.25 More iterative execution of algorithms

Примечание: This lesson shows how to combine the iterative execution of algorithms with the modeler to get more automation.

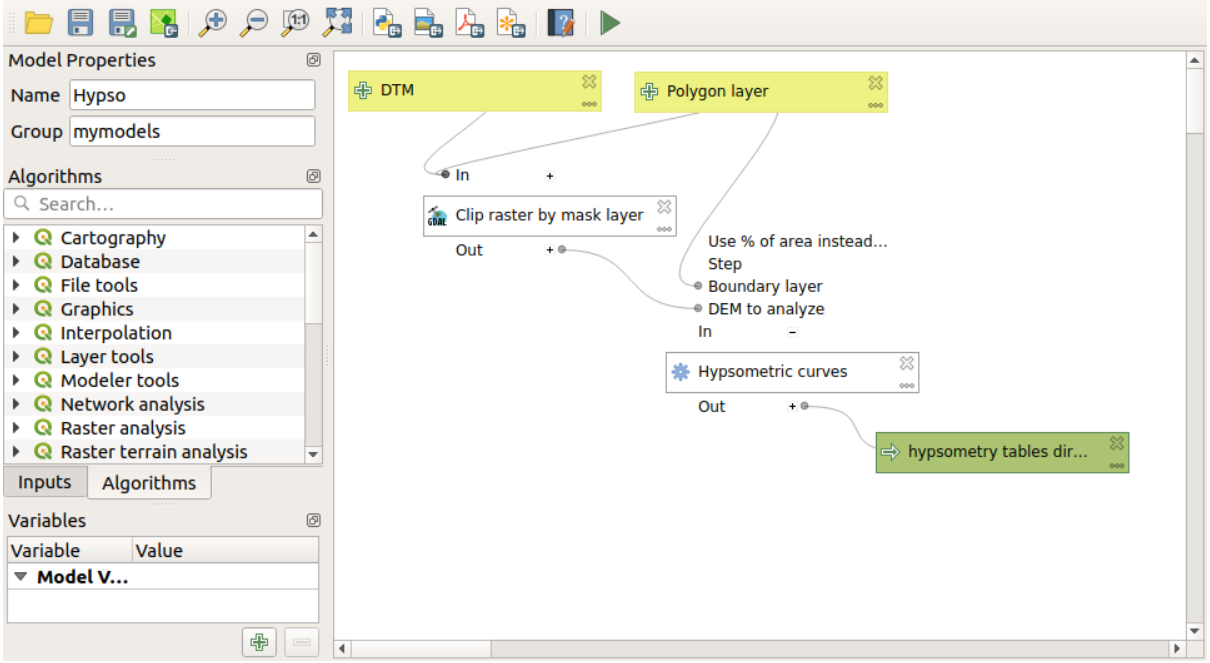
The iterative execution of algorithms is available not just for built-in algorithms, but also for the algorithms that you can create yourself, such as models. We are going to see how to combine a model and the iterative execution of algorithms, so we can obtain more complex results with ease.

The data the we are going to use for this lesson is the same one that we already used for the last one. In this case, instead of just clipping the DEM with each watershed polygon, we will add some extra steps and calculate a hypsometric curve for each of them, to study how elevation is distributed within the watershed.

Since we have a workflow that involves several steps (clipping + computing the hypsometric curve), we should go to the modeler and create the corresponding model for that workflow.

You can find the model already created in the data folder for this lesson, but it would be good if you first try to create it yourself. The clipped layer is not a final result in this case, since we are just interested in the curves, so this model will not generated any layers, but just a table with the curve data.

The model should look like this:

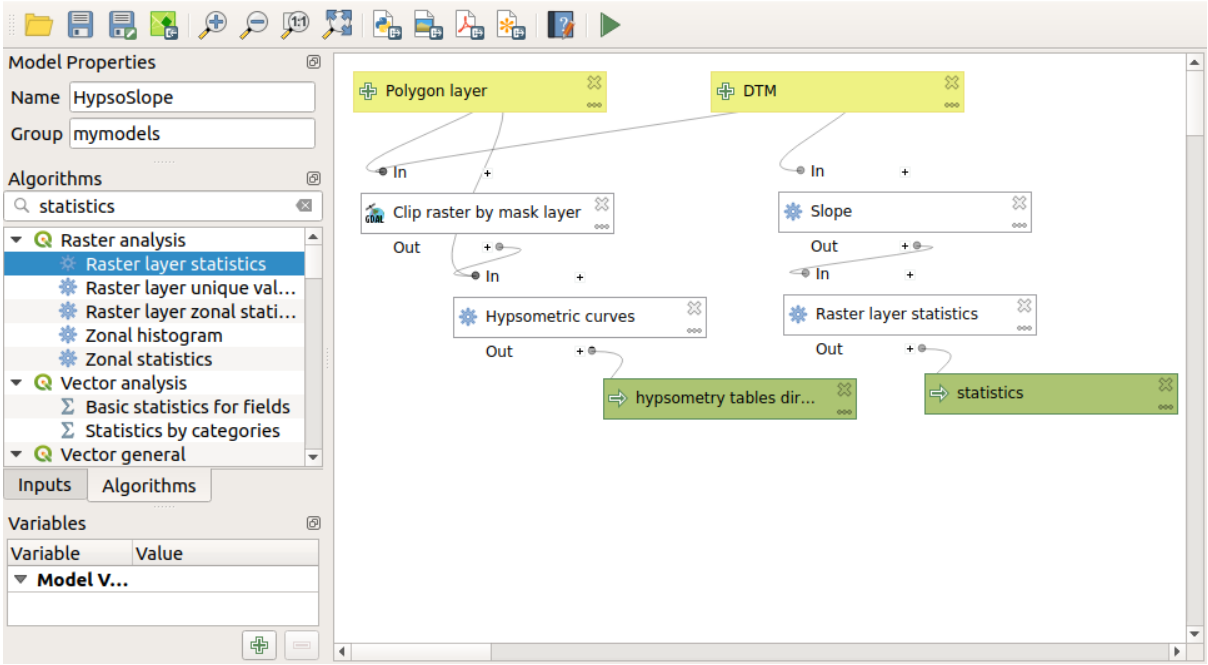


Add the model to you models folder, so it is available in the toolbox, and execute it.

Select the DEM and watersheds basins.

The algorithm will generate tables for all the basins and place them in the output directory.

We can make this example more complex by extending the model and computing some slope statistics. Add the *Slope* algorithm to the model, and then the *Raster statistics* algorithm, which should use the slope output as its only input.



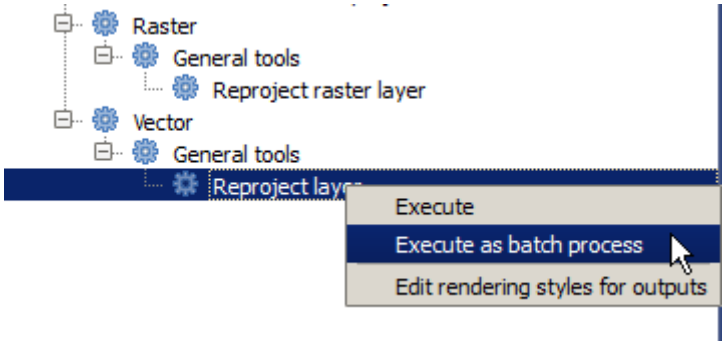
If you now run the model, apart from the tables you will get a set of pages with statistics. These pages will be available in the results dialog.

17.26 Интерфейс пакетной обработки

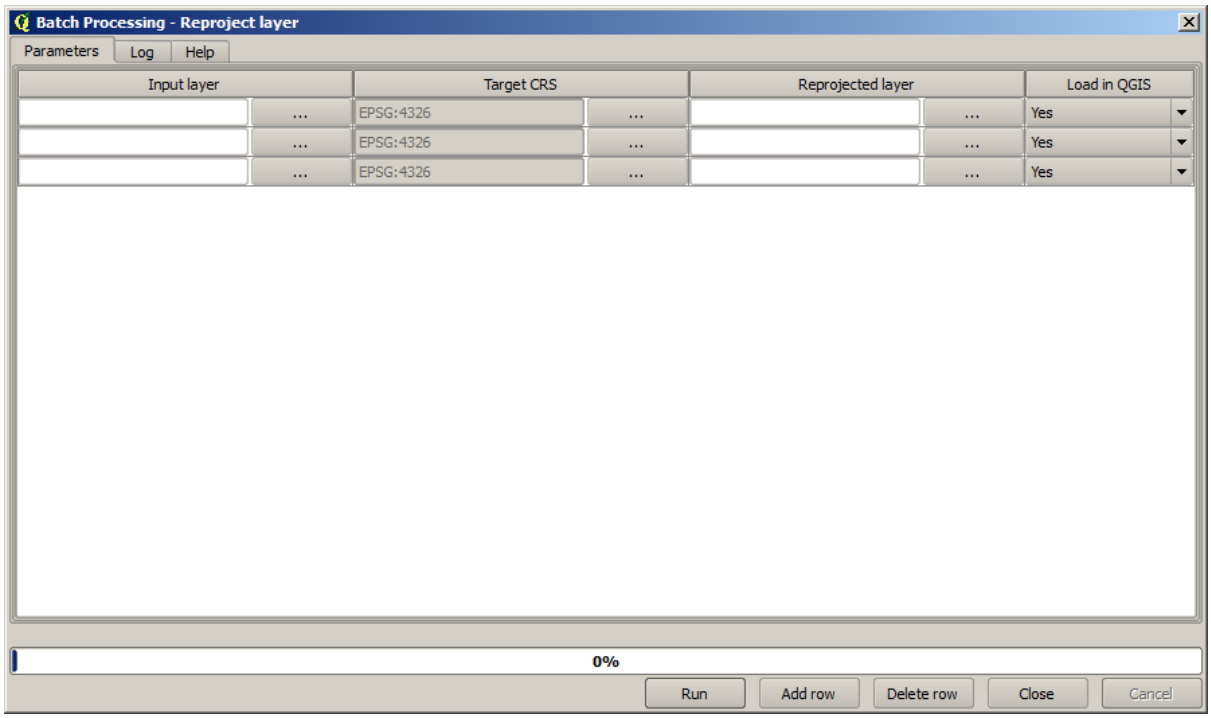
Примечание: This lesson introduces the batch processing interface, which allows to execute a single algorithm with a set of different input values.

Sometimes a given algorithm has to be executed repeatedly with different inputs. This is, for instance, the case when a set of input files have to be converted from one format to another, or when several layers in a given projection must be converted into another projection.

In that case, calling the algorithm repeatedly on the toolbox is not the best option. Instead, the batch processing interface should be used, which greatly simplifies performing a multiple execution of a given algorithm. To run an algorithm as a batch process, find it in the toolbox, and instead of double–clicking on it, right–click on it and select *Run as batch process*.



For this example, we will use the *Reproject algorithm*, so find it and do as described above. You will get to the following dialog.

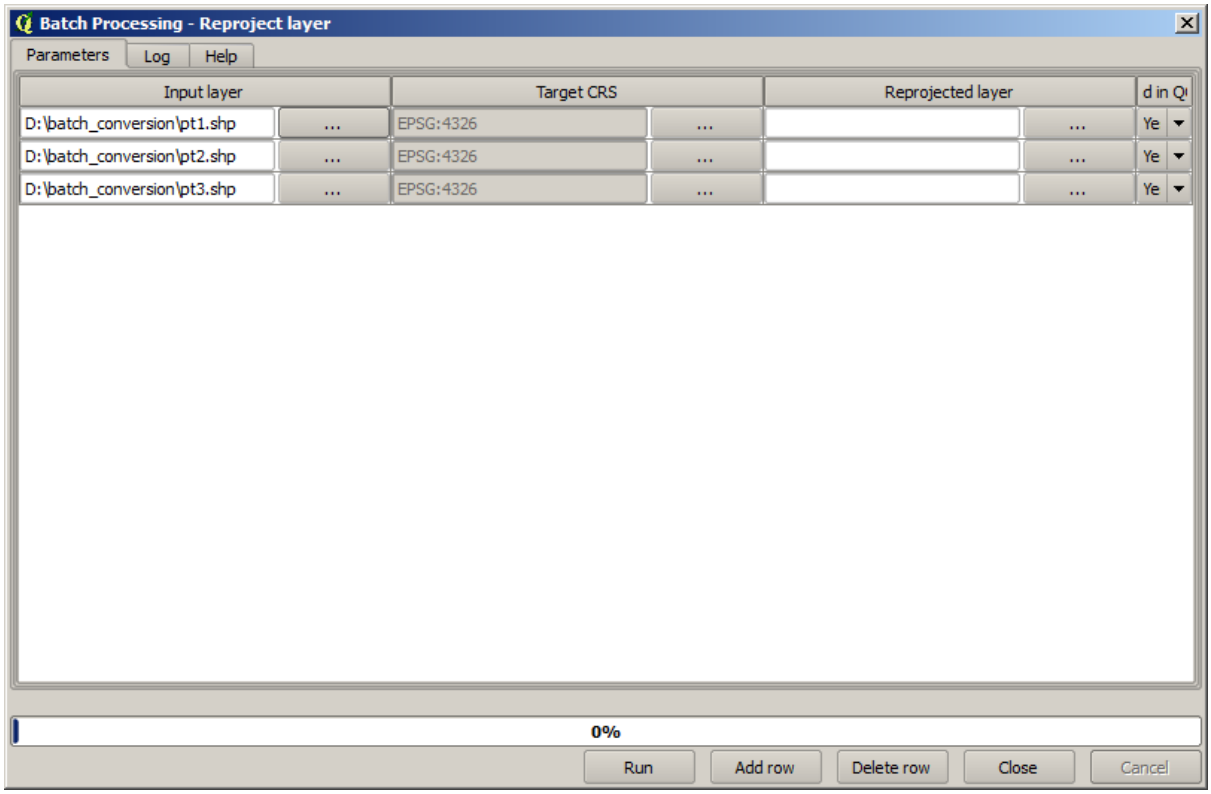


If you have a look at the data for this lesson, you will see that it contains a set of three shapefiles, but no QGIS project file. This is because, when an algorithm is run as a batch process, layer inputs can be selected either from the current QGIS project or from files. That makes it easier to process large amounts of layers, such as, for instance, all the layers in a given folder.

Each row in the table of the batch processing dialog represents a single execution of the algorithm. Cells in a row correspond to the parameter needed by the algorithm, which are not arranged one above the other, as in the normal single–execution dialog, but horizontally in that row.

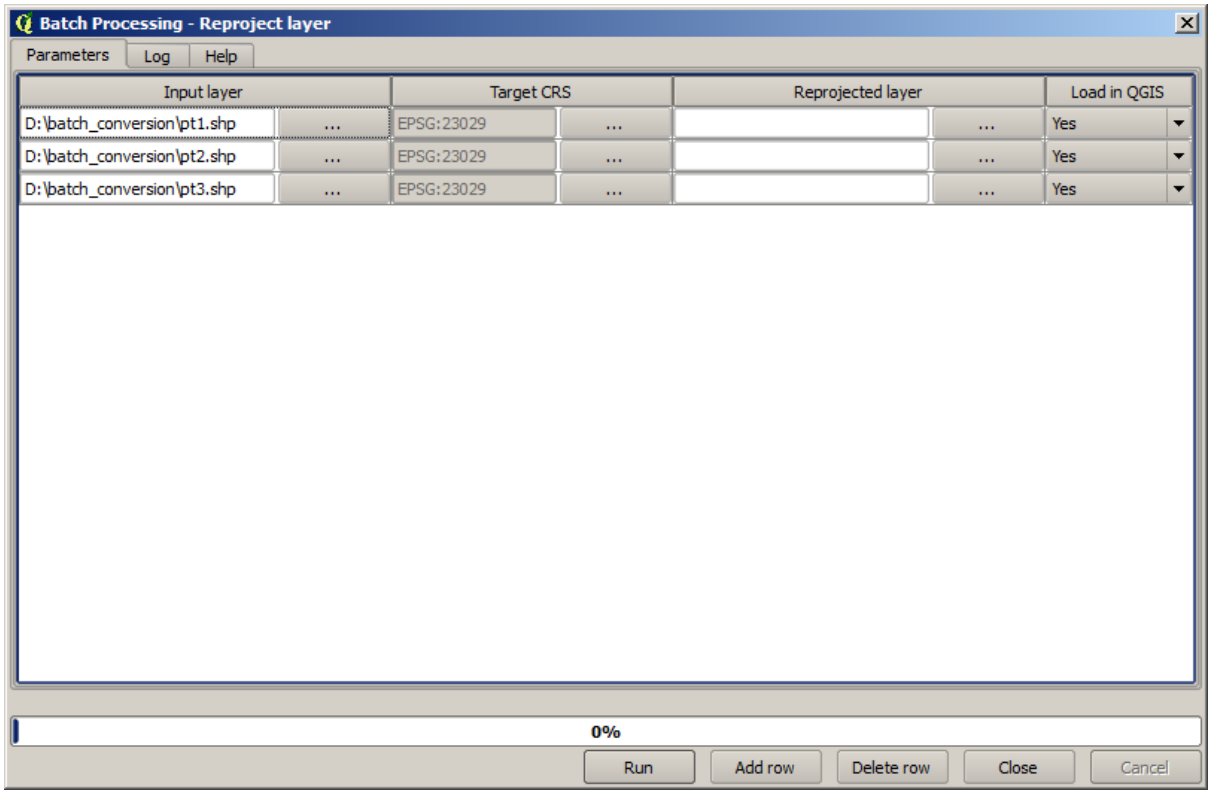
Defining the batch process to run is done by filling the table with the corresponding values, and the dialog itself contains several tools to make this task easier.

Let's start filling the fields one by one. The first column to fill is the *Input layer* one. Instead of entering the names of each one of the layers we want to process, you can select all of them and let the dialog put one in each row. Click on the button in the upper-left cell, and in the file selection dialog that will popup, select the three files to reproject. Since only one of them is needed for each row, the remaining ones will be used to fill the rows underneath.



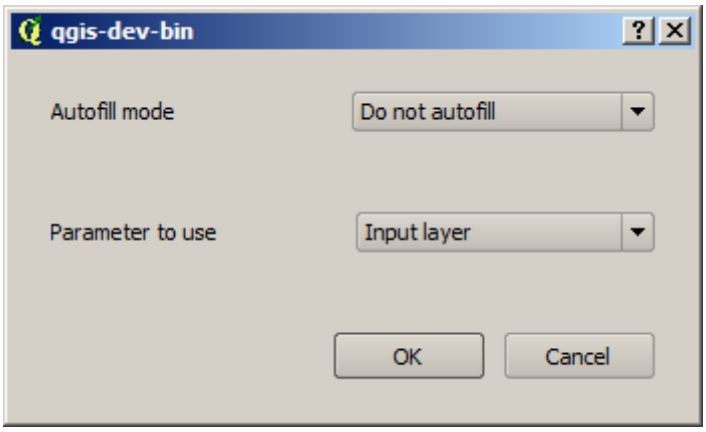
The default number of rows is 3, which is exactly the number of layers we have to convert, but if you select more layers, new rows will be added automatically. If you want to fill the entries manually, you can add more rows using the *Add row* button.

We are going to convert all those layers to the EPSG:23029 CRS, so we have to select that CRS in the second field. We want the same on for all rows, but we do not have to do it for every single row. Instead, set that CRS for the first row (the one at the top) using the button in the corresponding cell, and then double click on the column header. That causes all the cells in the column to be filled using the value of the top cell.

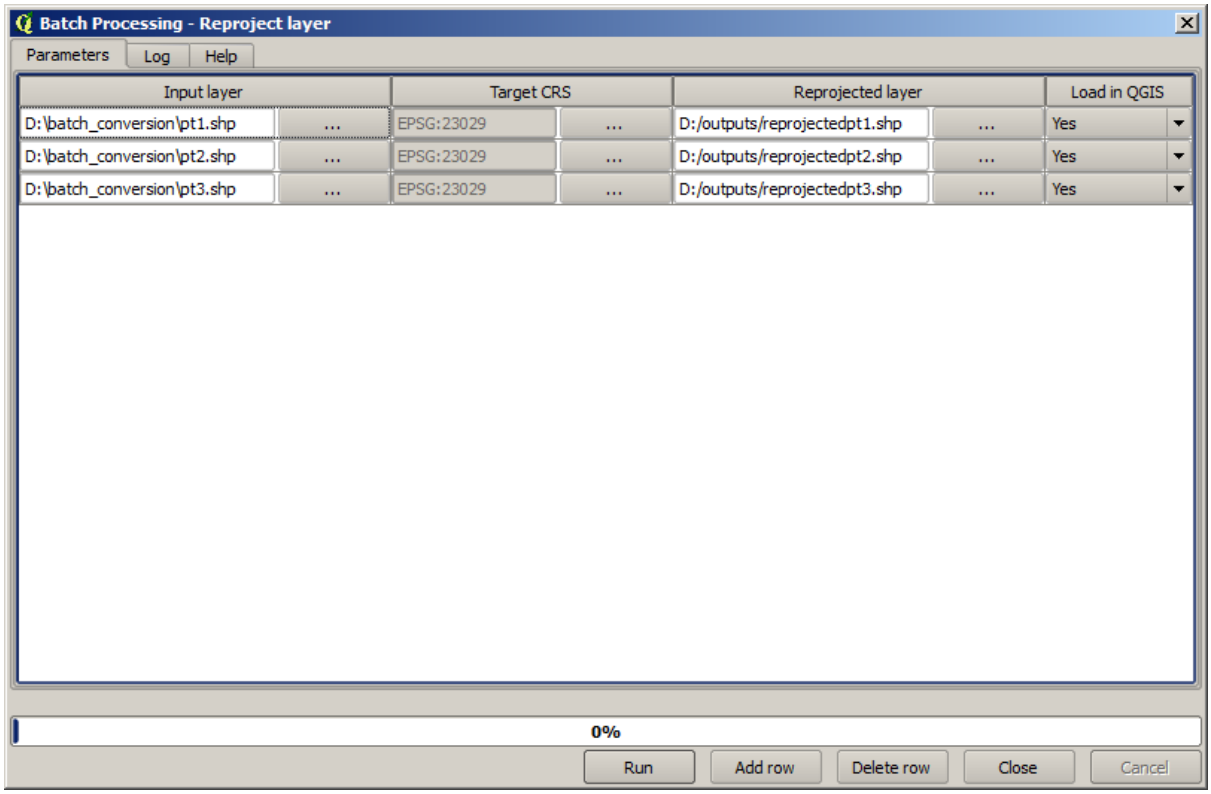


Finally, we have to select an output file for each execution, which will contain the corresponding reprojected layer. Once again, let's do it just for the first row. Click on the button in the upper cell and, in a folder where you want to put your output files, enter a filename (for instance, `reprojected.shp`).

Now, when you click *OK* on the file selection dialog, the file does not automatically gets written to the cell, but an input box like the following one is shown instead.



If you select the first option, only the current cell will be filled. If you select any of the other ones, all the rows below will be filled with a given pattern. In this case, we are going to select the *Fill with parameter value* option, and then the *Input Layer* value in the drop down menu below. That will cause the value in the *Input Layer* (that is, the layer name) to be added to the filename we have added, making each output filename different. The batch processing table should now look like this.



The last column sets whether or not to add the resulting layers to the current QGIS project. Leave the default *Yes* option, so you can see your results in this case.

Click on *OK* and the batch process will be run. If everything went fine, all your layers will have been processed, and 3 new layers would have been created.

17.27 Models in the batch processing interface

Предупреждение: Beware, this chapter is not well tested, please report any issue; images are missing

Примечание: This lesson shows another example of the batch processing interface, but this time using a model instead of a built-in algorithm

Models are just like any other algorithm, and they can be used in the batch processing interface. To demonstrate that, here is a brief example that we can do using our already well-known hydrological model.

Make sure you have the model added to your toolbox, and then run it in batch mode. This is what the batch processing dialog should look like.

Предупреждение: todo: Add image

Add rows up to a total of 5. Select the DEM file corresponding to this lesson as the input for all of them. Then enter 5 different threshold values as shown next.

Предупреждение: todo: Add image

As you see the batch processing interface can be run not just to run the same process on different datasets but also on the same dataset with different parameters.

Click on *OK* and you should get 5 new layers with watersheds corresponding to the specified 5 threshold values.

17.28 Хуки пред- и постобработки

Примечание: This lesson shows how to use pre- and post-execution hooks, which allow to perform additional operations before and after actual processing.

Pre- and post-execution hooks are Processing scripts that run before and after actual data processing is performed. This can be used to automate tasks that should be performed whenever an algorithm is executed.

The syntax of the hooks is identical to the syntax of Processing scripts, see the corresponding chapter in the QGIS User Guide for more details.

In addition to all scripts features, in hooks you can use a special global variable named `alg`, which represents the algorithm that has just been (or is about to be) executed.

Here is an example post-execution script. By default, Processing stores analysis results in temporary files. This script will copy outputs to a specific directory, so they won't be deleted after closing QGIS.

```
import os
import shutil
from processing.core.outputs import OutputVector, OutputRaster, OutputFile

MY_DIRECTORY = '/home/alex/outputs'

for output in alg.outputs:
    if isinstance(output, (OutputVector, OutputRaster, OutputFile)):
        dirname = os.path.split(output.value)[0]
        shutil.copytree(dirname, MY_DIRECTORY)
```

In the first two lines we import the required Python packages: `os` — for path manipulations, e.g. extracting file name, and `shutil` — for various filesystem operations like copying files. In the third line we import Processing outputs. This will be explained in more detail later in this lesson.

Then we define a `MY_DIRECTORY` constant, which is the path to the directory where we want to copy analysis results.

At the end of the script, we have the main hook code. In the loop we iterate over all algorithm outputs and check if this output is a file-based output and can be copied. If so, we determine top-level directory in which output files are located and then copy all files to our directory.

To activate this hook we need to open the Processing options, find the entry named *Post-execution script file* in the *General* group, and specify the filename of the hook script there. the specified hook will be executed after each Processing algorithm.

In a similar way, we can implement pre-execution hooks. For example, let's create a hook to check input vectors for geometry errors.

```
from qgis.core import QgsGeometry, QgsFeatureRequest
from processing.core.parameters import ParameterVector

for param in alg.parameters:
    if isinstance(param, ParameterVector):
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        layer = processing.getObject(param.value)
        for f in layer.getFeatures(QgsFeatureRequest().setSubsetOfAttributes([])):
            errors = f.geometry().validateGeometry()
            if len(errors) > 0:
                progress.setInfo('One of the input vectors contains invalid␣
↪geometries!')
```

As in the previous example, first we import required QGIS and Processing packages.

Then we iterate over all the algorithm parameters and if a ParameterVector parameter is found, we get the corresponding vector layer object from it. We loop over all the features of the layer and check them for geometry errors. If at least one feature contains an invalid geometry, we print a warning message.

To activate this hook we need enter its filename in the *Pre-execution script file* option in the Processing configuration dialog. The hook will be executed before running any Processing algorithm.

17.29 Other programs

Module contributed by Paolo Cavallini - Faunalia

Примечание: This chapter shows how to use additional programs from inside Processing. To complete it, you must have installed, with the tools of your operating system, the relevant packages.

17.29.1 GRASS

GRASS is a free and open source GIS software suite for geospatial data management and analysis, image processing, graphics and maps production, spatial modeling, and visualization.

It is installed by default on Windows through the OSGeo4W standalone installer (32 and 64 bit), and it is packaged for all major Linux distributions.

17.29.2 R

R is a free and open source software environment for statistical computing and graphics.

It has to be installed separately, together with a few necessary libraries (**LIST**). To enable the use of R in QGIS, the *Processing R Provider* plugin must also be installed.

The beauty of Processing implementation is that you can add your own scripts, simple or complex ones, and they may then be used as any other module, piped into more complex workflows, etc.

Test some of the preinstalled examples, if you have R already installed (remember to activate R modules from the General configuration of Processing).

17.29.3 Others

LASTools is a set of mixed, free and proprietary commands to process and analyze LiDAR data. Availability in various operating system varies.

More tools are available through additional plugins, e.g.:

- LecoS: a suite for land cover statistics and landscape ecology
- lwgeom: formerly part of PostGIS, this library brings a few useful tools for geometry cleanup
- Animove: tools to analyse the home range of animals.

More will come.

17.29.4 Comparison among backends

Buffers and distances

Let's load `points.shp` and type `buf` in the filter of the Toolbox, then double click on:

- *Fixed distance buffer*: Distance 10000
- *Variable distance buffer*: Distance field SIZE
- *v.buffer.distance*: distance 10000
- *v.buffer.column*: bufcolumn SIZE
- *Shapes Buffer*: fixed value 10000 (dissolve and not), attribute field (with scaling)

See how speed is quite different, and different options are available.

Exercise for the reader: find the differences in geometry output between different methods.

Now, raster buffers and distances:

- first, load and rasterize the vector `rivers.shp` with *GRASS* ► *v.to.rast.value*; **beware:** cell size must be set to 100 m, otherwise the computation time will be enormous; resulting map will have 1 and NULLs
- same, with *SAGA* ► *Shapes to Grid* ► *COUNT* (resulting map: 6 to 60)
- then, *proximity* (value= 1 for GRASS, a list of rivers ID for SAGA), *r.buffer* with parameters 1000,2000,3000, *r.grow.distance* (the first of the two maps; the second will show the areas pertaining to each river, if done on the SAGA raster).

Dissolve

Dissolve features based on a common attribute:

- *GRASS* ► *v.dissolve* `municipalities.shp` on PROVINCIA
- *QGIS* ► *Dissolve* `municipalities.shp` on PROVINCIA
- *OGR* ► *Dissolve* `municipalities.shp` on PROVINCIA
- *SAGA* ► *Polygon Dissolve* `municipalities.shp` on PROVINCIA (**NB:** *Keep inner boundaries* must be unselected)

Примечание: The last one is broken in SAGA <=2.10

Exercise for the reader: find the differences (geometry and attributes) between different methods.

17.30 Interpolation and contouring

Module contributed by Paolo Cavallini - Faunalia

Примечание: This chapter shows how to use different backends to calculate different interpolations.

17.30.1 Interpolation

The project shows a gradient in rainfall, from south to north. Let's use different methods for interpolation, all based on vector `points.shp`, parameter `RAIN`:

Предупреждение: Set cell size to `500` for all analyses.

- *GRASS* ► *v.surf.rst*
- *SAGA* ► *Multilevel B-Spline Interpolation*
- *SAGA* ► *Inverse Distance Weighted* [Inverse distance to a power; Power: 4; Search radius: Global; Search range: all points]
- *GDAL* ► *Grid (Inverse Distance to a power)* [Power:4]
- *GDAL* ► *Grid (Moving average)* [Radius1&2: 50000]

Then measure variation among methods and correlate it with distance to points:

- *GRASS* ► *r.series* [Unselect Propagate NULLs, Aggregate operation: stddev]
- *GRASS* ► *v.to.rast.value* on `points.shp`
- *GDAL* ► *Proximity*
- *GRASS* ► *r.covar* to show the correlation matrix; check the significance of the correlation e.g. with http://vassarstats.net/rsig.html.

Thus, areas far from points will have less accurate interpolation.

17.30.2 Contour

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:** in some older SAGA versions, output shp is not valid, known bug]

17.31 Vector simplification and smoothing

Module contributed by Paolo Cavallini - Faunalia

Примечание: This chapter shows how simplify vectors, and smooth out sharp corners.

Sometimes we need a simplified version of a vector, to have a smaller file size and get rid of unnecessary details. Many tools do this in a very rough way, and miss the adjacency and sometimes the topological correctness of polygons. GRASS is the ideal tool for this: being a topological GIS, adjacency and correctness are preserved even at

very high simplification levels. In our case, we have a vector resulting from a raster, thus showing a «saw» pattern at borders. Applying a simplification results in straight lines:

- *GRASS* ► *v.generalize* [Maximal tolerance value: 30 m]

We can also do the reverse, and make a layer more complex, smoothing out sharp corners:

- *GRASS* ► *v.generalize* [method: chaiken]

Try to apply this second command both to original vector and to the one from the first analysis, and see the difference. Note that adjacency is not lost.

This second option can be applied e.g. to contour lines resulting from a coarse raster, to GPS tracks with sparse vertices, etc.

17.32 Planning a solar farm

Module contributed by Paolo Cavallini - Faunalia

Примечание: This chapter shows how to use several criteria to locate the areas suitable for installing a photovoltaic power station

First of all, create an aspect map from DTM:

- *GRASS* ► *r.aspect* [Data type: int; cell size:100]

In GRASS, aspect is calculated in degrees, counterclockwise starting from East. To extract only South facing slopes (270 degrees +- 45), we can reclassify it:

- *GRASS* ► *r.reclass*

with the following rules:

```
225 thru 315 = 1 south
* = NULL
```

You can use the text file `reclass_south.txt` provided. Note that with these simple text files we can create also very complex reclassifications.

We want to build a large farm, so we select only large (> 100 ha) contiguous areas:

- *GRASS* ► *r.reclass.greater*

Finally, we convert to a vector:

- *GRASS* ► *r.to.vect* [Feature type: area; Smooth corners: yes]

Exercise for the reader: repeat the analysis, replacing GRASS commands with analogous from other programs.

17.33 Use R scripts in Processing

Module contributed by Matteo Ghetta - funded by Scuola Superiore Sant'Anna

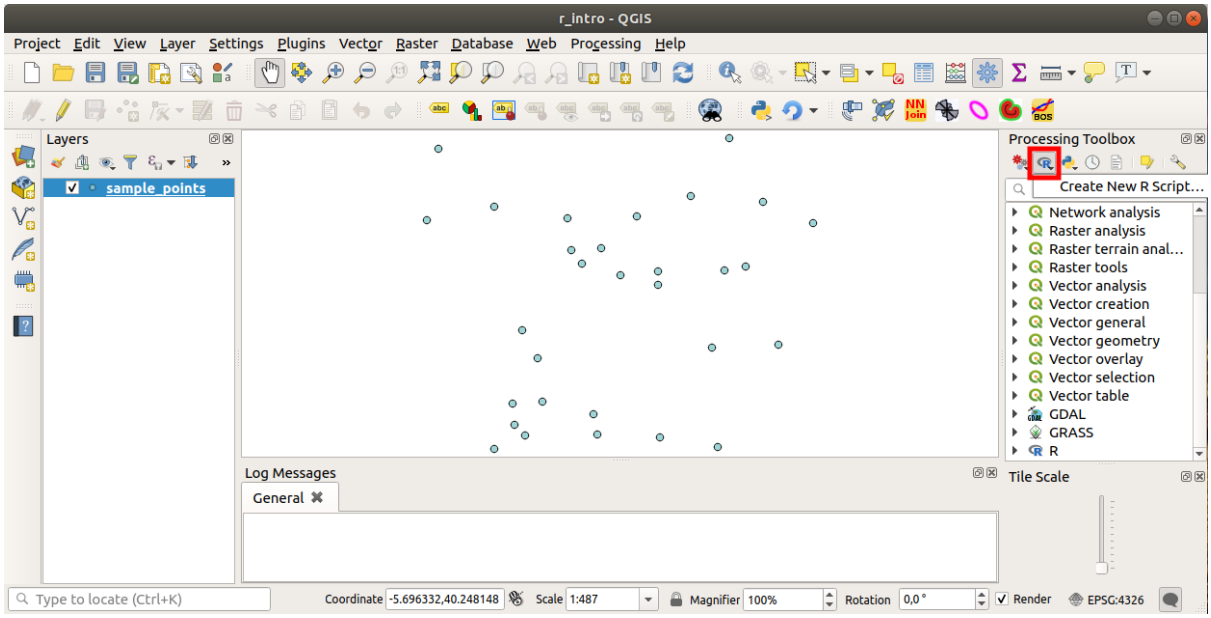
Processing (with the `Processing R Provider` plugin) makes it possible to write and run R scripts inside QGIS.

Предупреждение: R has to be installed on your computer and the PATH has to be correctly set up. Moreover Processing just calls the external R packages, it is not able to install them. So be sure to install external packages directly in R. See the related chapter in the user manual.

Примечание: If you have *package* problems, it may be related to missing *mandatory* packages required by Processing, like `sp`, `rgdal` and `raster`.

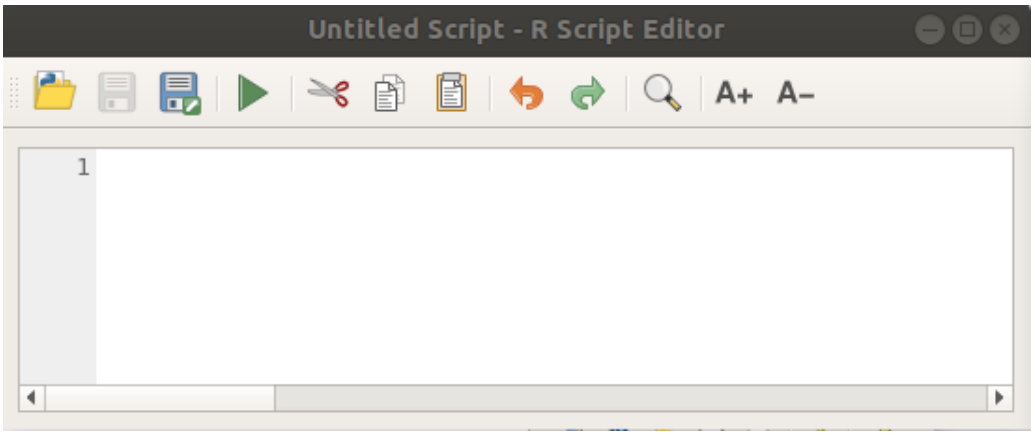
17.33.1 Adding scripts

Adding a script is simple. The easiest way is to open the Processing toolbox and choose *Create new R script…* from the R menu (labelled with an R icon) at the top of the Processing Toolbox. You can also create the script in for instance a text editor and save it in your R scripts folder (`processing/rscripts`). When it has been saved there, it will be available for editing by right-clicking on the script name in the processing toolbox and then choose *Edit Script…*).



Примечание: If you cannot see R in Processing, you have to activate it in *Processing* ► *Options* ► *Providers*

It opens a *script editor window* in which you have to specify some parameters before you can add the script body.



17.33.2 Creating plots

In this tutorial we are going to create a **boxplot** of a vector layer field.

Open the `r_intro.qgs` QGIS project under the `exercise_data/processing/r_intro/` folder.

Script parameters

Open the editor and start writing at the beginning of it.

You **must** specify some parameters **before** the script body:

1. The name of the group (*plots* in this case) in which you want to put your script (if the group does not exist, it will be created):

   ```
   ##plots=group
   ```

   You will find your script in the **plots** R group in the Processing toolbox.

2. You have to tell Processing that you want to display a plot (in this example):

   ```
   ##showplots
   ```

   You will then find a link to the plot in the **Result Viewer** panel (can be turned on / off in *View* ► *Panels* and with *Processing* ► *Results Viewer*).

3. You also need to tell Processing about your input data. In this example we want to create a plot from a field of a vector layer:

   ```
   ##Layer=vector
   ```

   Processing now knows that the input is a vector. The name *Layer* is not important, what matters is the **vector** parameter.

4. Finally, you have to specify the input field of the vector layer (using the name you have provided above - *Layer*):

   ```
   ##X=Field Layer
   ```

   Processing now knows that you need a field of *Layer*, and that you will call it **X**.

5. It is also possible to define the name of your script using `name`:

   ```
   ##My box plot script=name
   ```

   If not defined, the file name will be used as the name of the script.

Script body

Now that you have set up the *heading* of the script you can add the function:

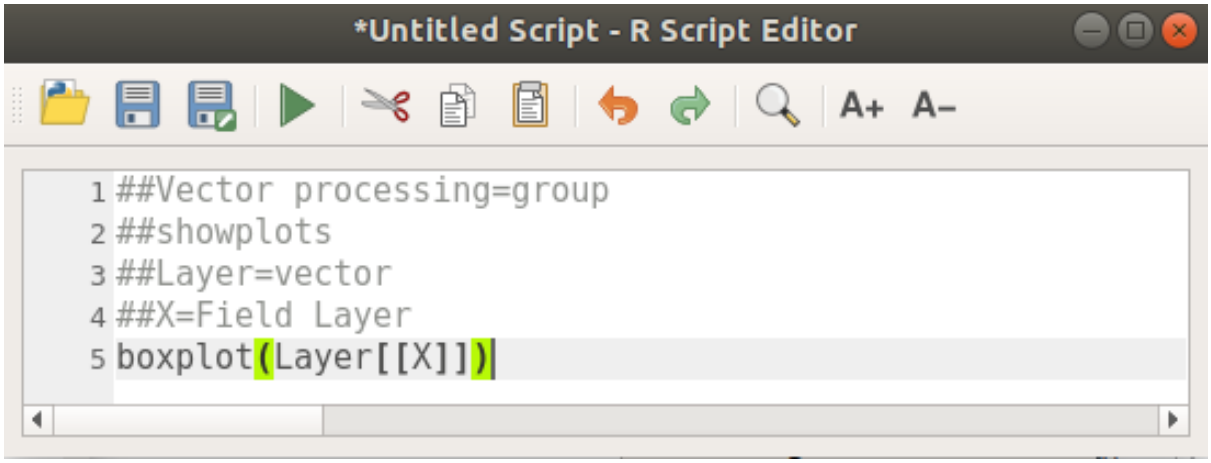
```
boxplot(Layer[[X]])
```

boxplot is the name of the R function, the parameter **Layer** is the name that you have defined for the input dataset and **X** is the name you have defined for the field of that dataset.

> **Предупреждение:** The parameter **X** has to be within double square brackets (`[[]]`).

The final script looks like this:

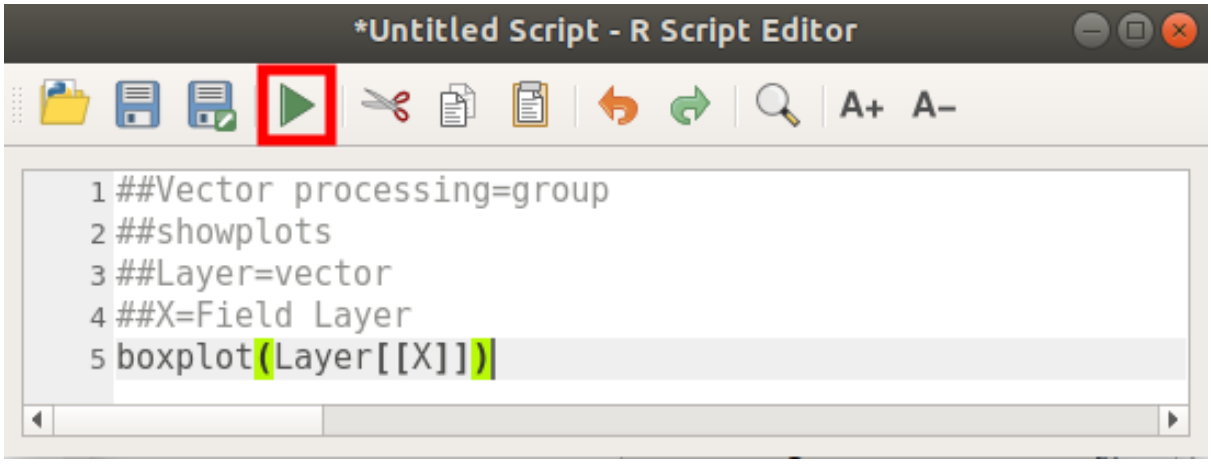
```
##Vector processing=group
##showplots
##Layer=vector
##X=Field Layer
boxplot(Layer[[X]])
```



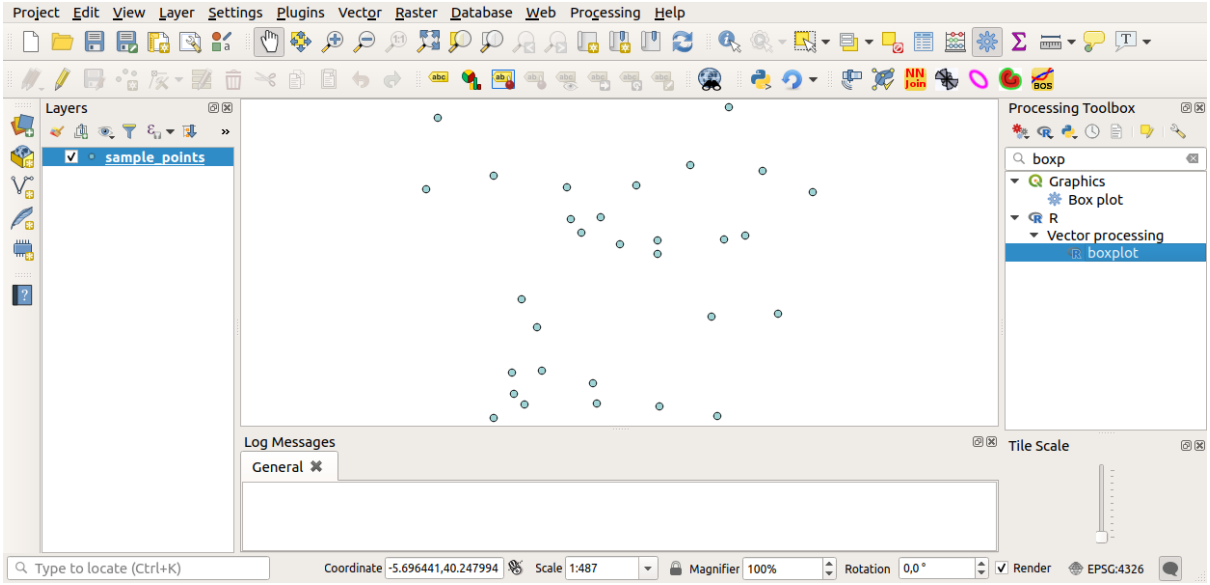
Save the script in the default path suggested by Processing (processing/rscripts). If you have not defined a `name` in the script heading, the file name you choose will become the name of the script in the Processing toolbox.

Примечание: You can save the script wherever you like, but Processing will then not be able to include it in the processing toolbox automatically, so you have to upload it manually.

Now just run it using the button on the top of the editor window:



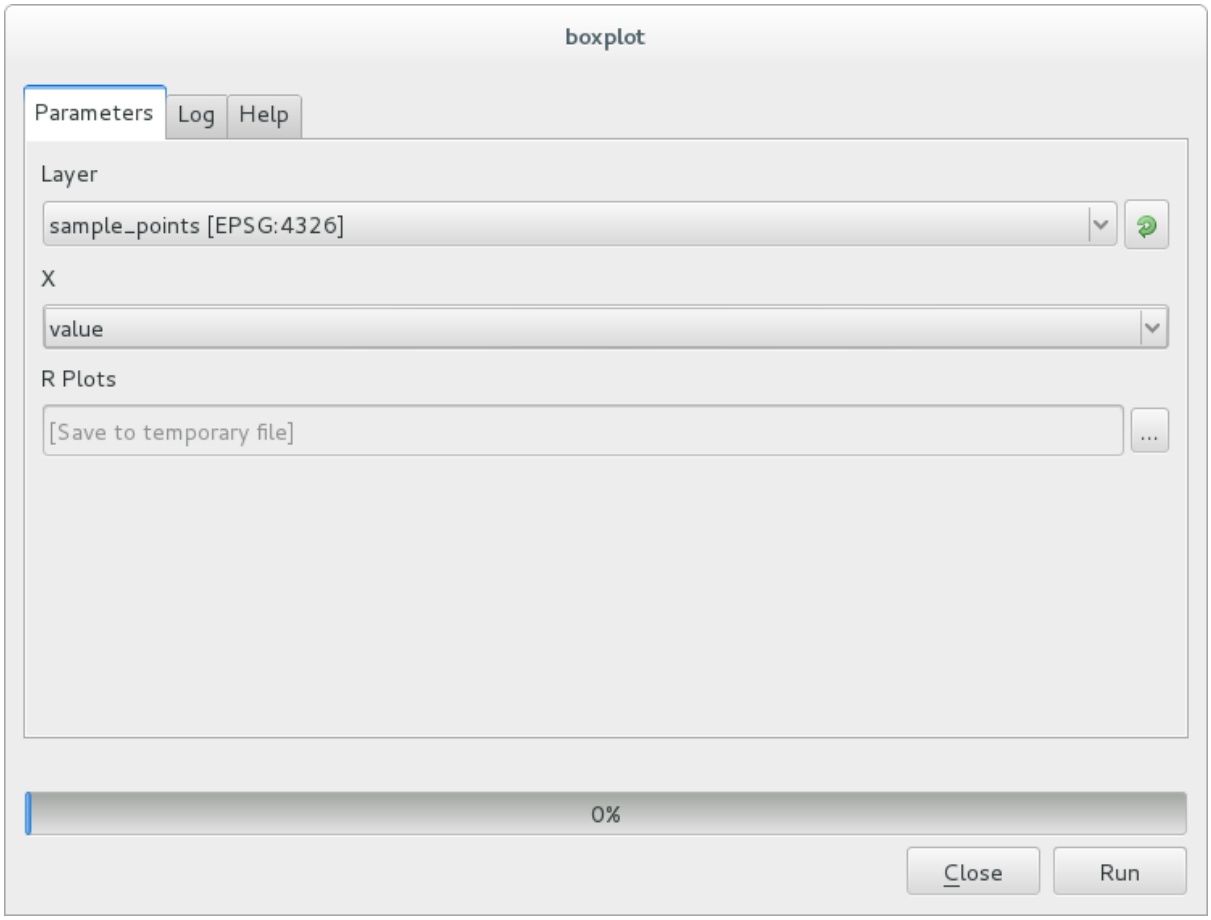
Once the editor window has been closed, use the text box of Processing to find your script:



You can now fill the parameters required in the Processing algorithm window:

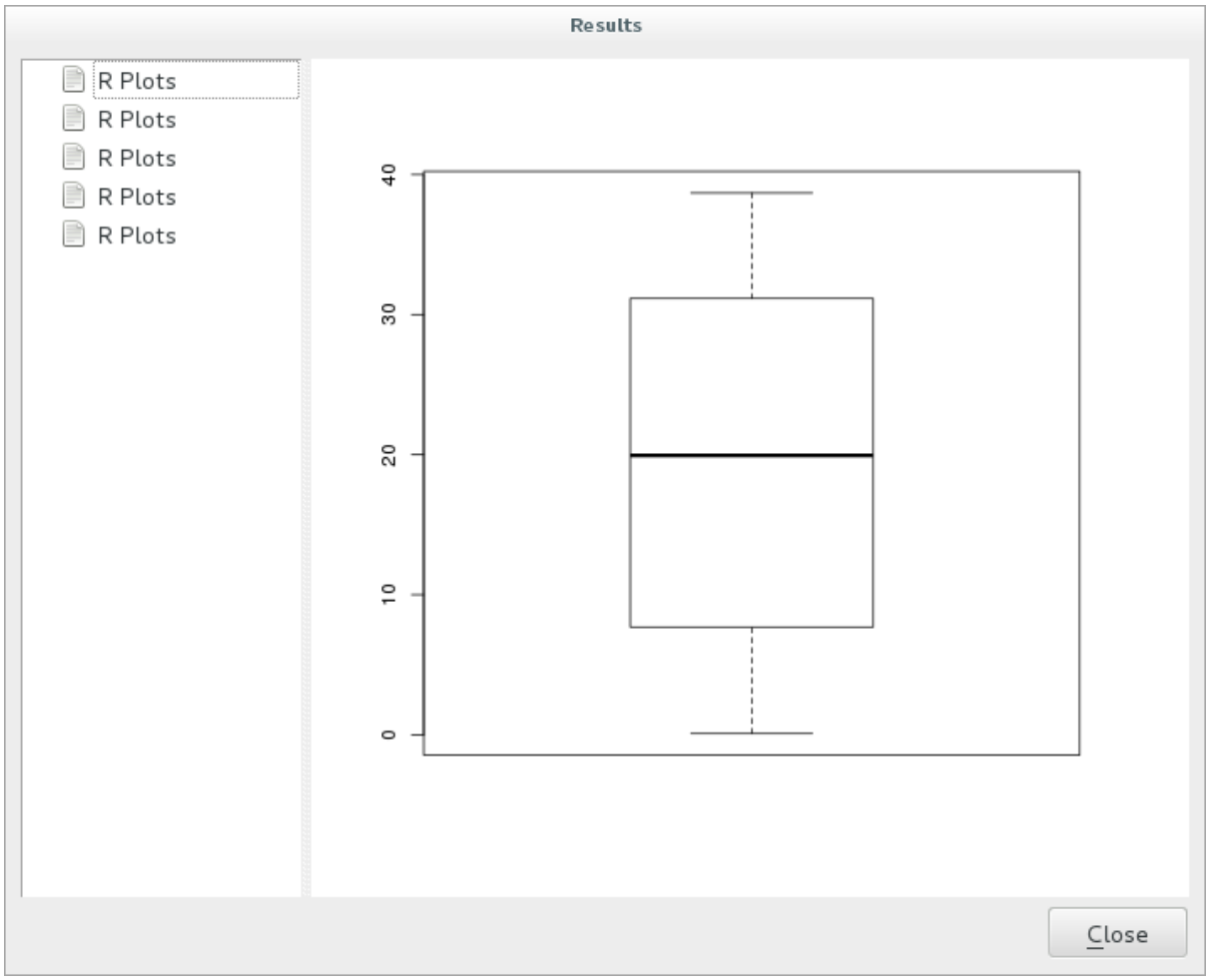
- for **Layer** choose *sample_points*
- for the **X** field choose *value*

Click on **Run**.



The **Result window** should be automatically opened, if not, just click on *Processing* ► *Result Viewer*….

Click on the link in the viewer and you will see:



Примечание: You can open, copy and save the image by right clicking on the plot.

17.33.3 Create a vector

You can also create a vector layer and have it automatically loaded into QGIS.

The following example has been taken from the `Random sampling grid` script that can be found in the online collection of R scripts (the scripts in this online collection can be found in https://github.com/qgis/QGIS-Processing/tree/master/rscripts).

The aim of this exercise is to create a random point vector layer using an input vector layer to restrict the extent using the `spsample` function of the `sp` package.

Script parameters

As before we have to set some parameters before the script body:

1. Specify the name of the group in which you want to put your script, in this case *Point pattern analysis*:

```
##Point pattern analysis=group
```

2. Define an input parameter (a vector layer) that will constrain the placement of the random points:

```
##Layer=vector
```

3. Set an input parameter for the number of points that are going to be created (`Size`, with a default value of `10`):

```
##Size=number 10
```

Примечание: Since a default value (`10`) is defined, the user can change this number or can leave the parameter without a number.

---

4. Specify that there is an output vector layer (called `Output`):

```
##Output=output vector
```

Script body

Now you can add the body of the function:

1. Use the `spsample` function:

```
pts=spsample(Layer, Size, type="random")
```

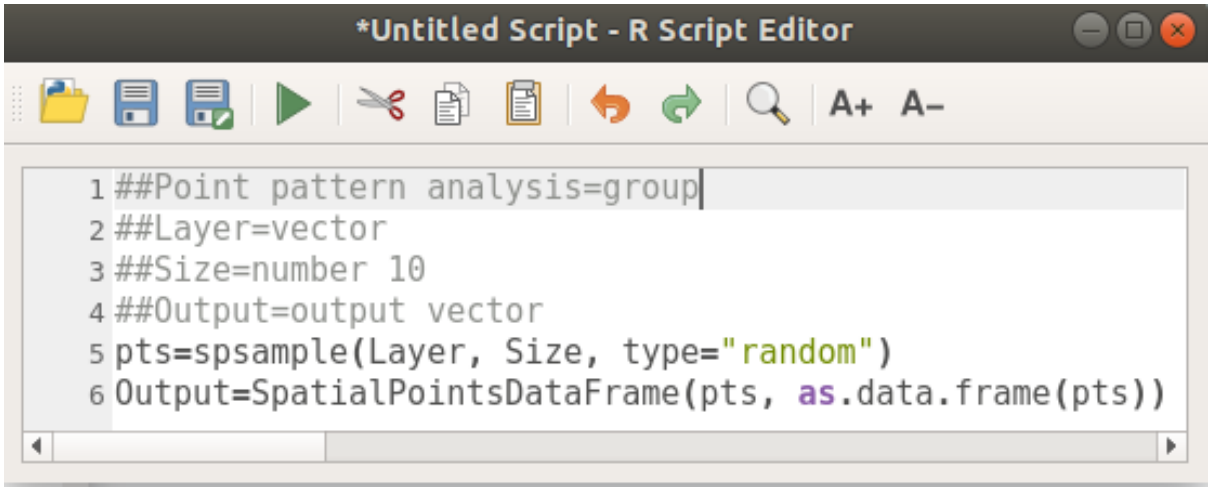
The function uses the *Layer* to constrain the placement of the points (if it is a line layer, a points will have to be on one of the lines in the layer, if it is a polygon layer, a point will have to be within a polygon). The number of points is taken from the *Size* parameter. The sampling method is *random*.

2. Generate the output (the `Output` parameter):

```
Output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```

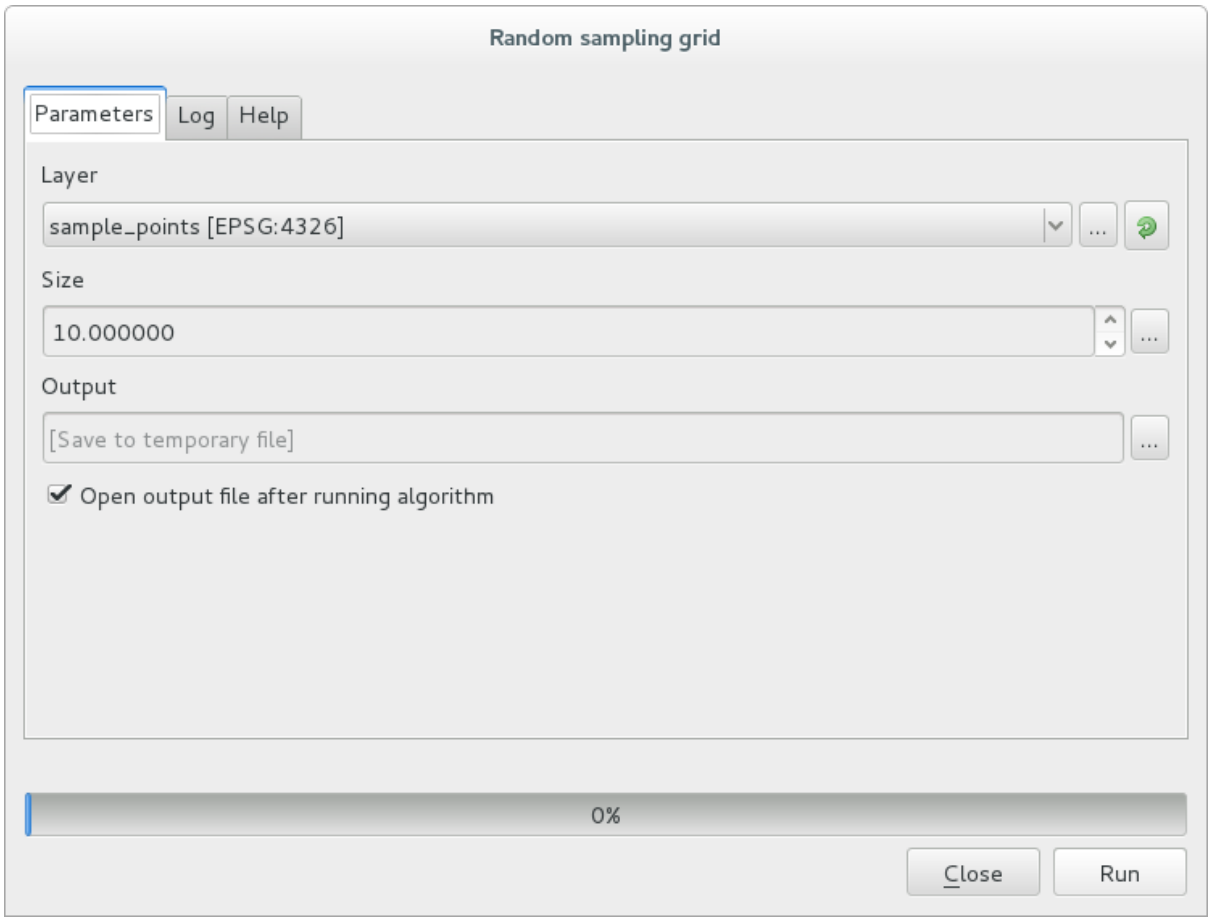
The final script looks like this:

```
##Point pattern analysis=group
##Layer=vector
##Size=number 10
##Output=output vector
pts=spsample(Layer, Size, type="random")
Output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```



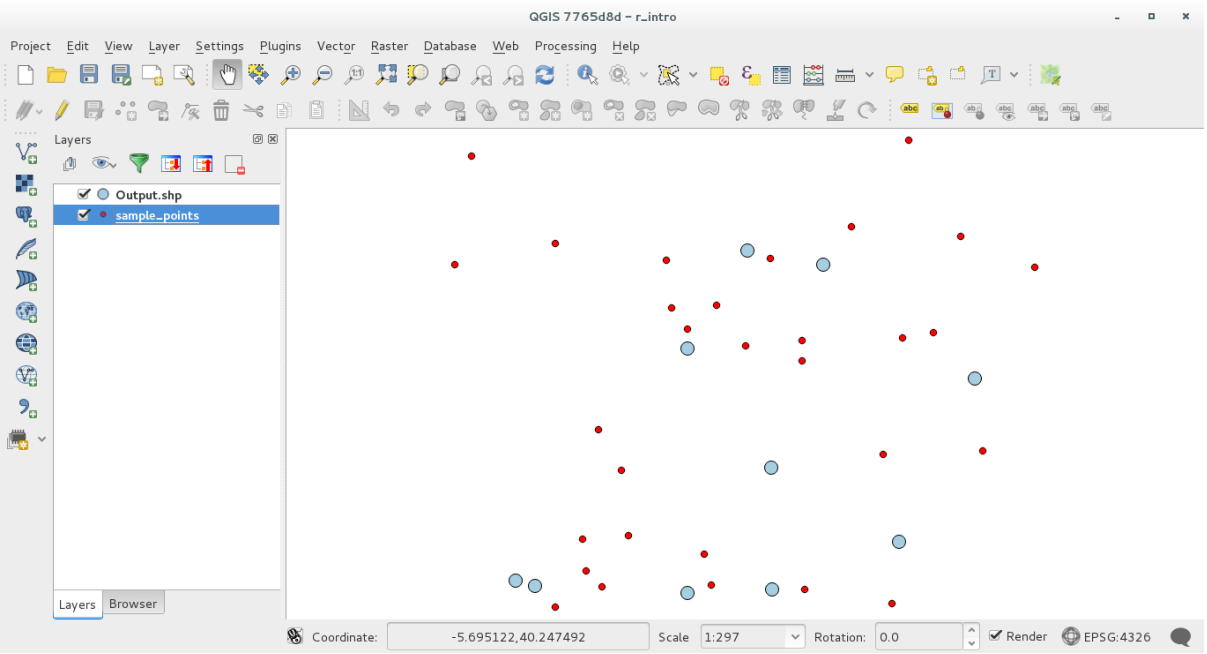
Save it and run it, clicking on the run button.

In the new window type in the right parameters:



and click on run.

The result layer will be added to the table of contents and its points will be displayed on the map canvas:



17.33.4 Text and graph output from R - syntax

Processing (with the `Processing R Provider plugin`) uses special syntax to get the results out of R:

- `>` before your command, as in `>lillie.test(Layer[[Field]])` means the result should be sent to R output (Result viewer)
- `+` after a plot enables overlay plots. For example `plot(Layer[[X]], Layer[[Y]]) + abline(h=mean(Layer[[X]]))`

17.34 Predicting landslides

Module contributed by Paolo Cavallini - Faunalia

Примечание: This chapter shows how to create an oversimplified model to predict the probability of landslides.

First, we calculate slope (choose among various backends; the interested reader can calculate the difference between the outputs):

- *GRASS* ► *r.slope*
- *SAGA* ► *Slope, Aspect, Curvature*
- *GDAL Slope*

Then we create a model of predicted rainfall, based on the interpolation of rainfall values at meteo stations:

- *GRASS* ► *v.surf.rst* (resolution: 500 m)

The probability of a landslide will be very roughly related to both rainfall and slope (of course a real model will use more layers, and appropriate parameters), let's say `(rainfall * slope )/100`:

- *SAGA* ► *Raster calculator* rain, slope: `(a*b)/100` (or: *GRASS* ► *r.mapcalc*)
- then let's calculate what are the municipalities with the greatest predicted risk of rainfall: *SAGA* ► *Raster statistics with polygons* (the parameters of interest are *Maximum* and *Mean*)

ГЛАВА 18

Module: Using Spatial Databases in QGIS

In this module you will learn about how to use Spatial Databases with QGIS to manage, display and manipulate data in the database as well as performing analysis by querying. We will primarily use PostgreSQL and PostGIS (which were covered in previous sections), but the same concepts are applicable to other spatial database implementations including SpatiaLite.

18.1 Lesson: Working with Databases in the QGIS Browser

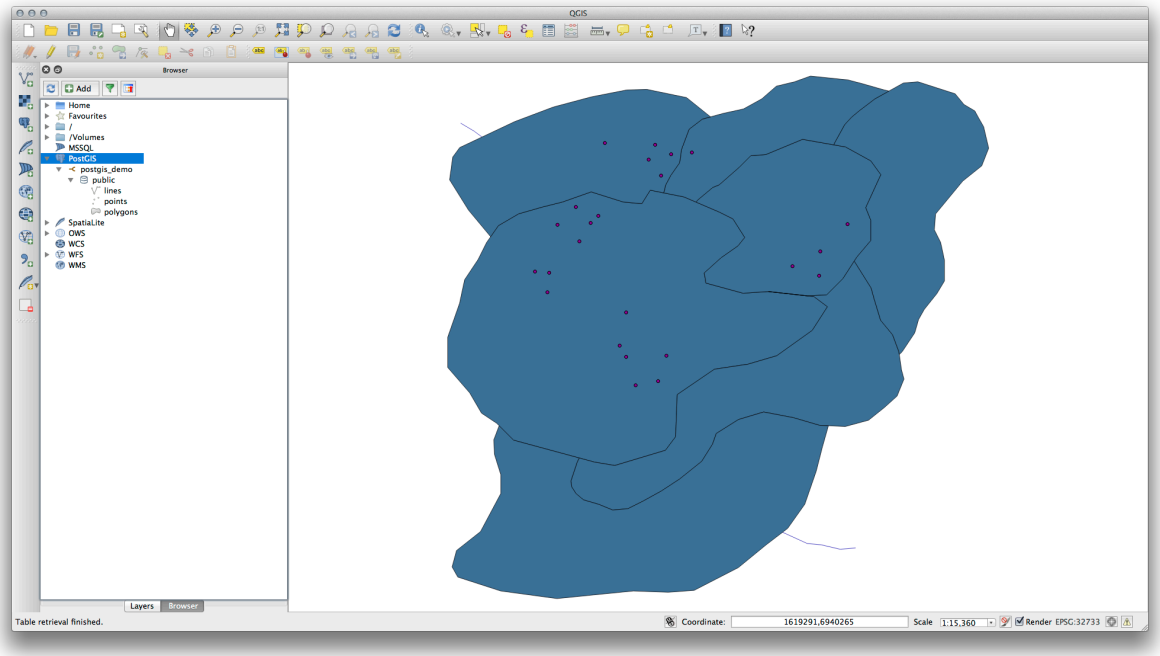
In the previous 2 modules we looked at the basic concepts, features and functions of relational databases as well as extensions that let us store, manage, query and manipulate spatial data in a relational database. This section will dive deeper into how to effectively use spatial databases in QGIS.

The goal for this lesson: To learn how to interact with spatial databases using the QGIS Browser interface.

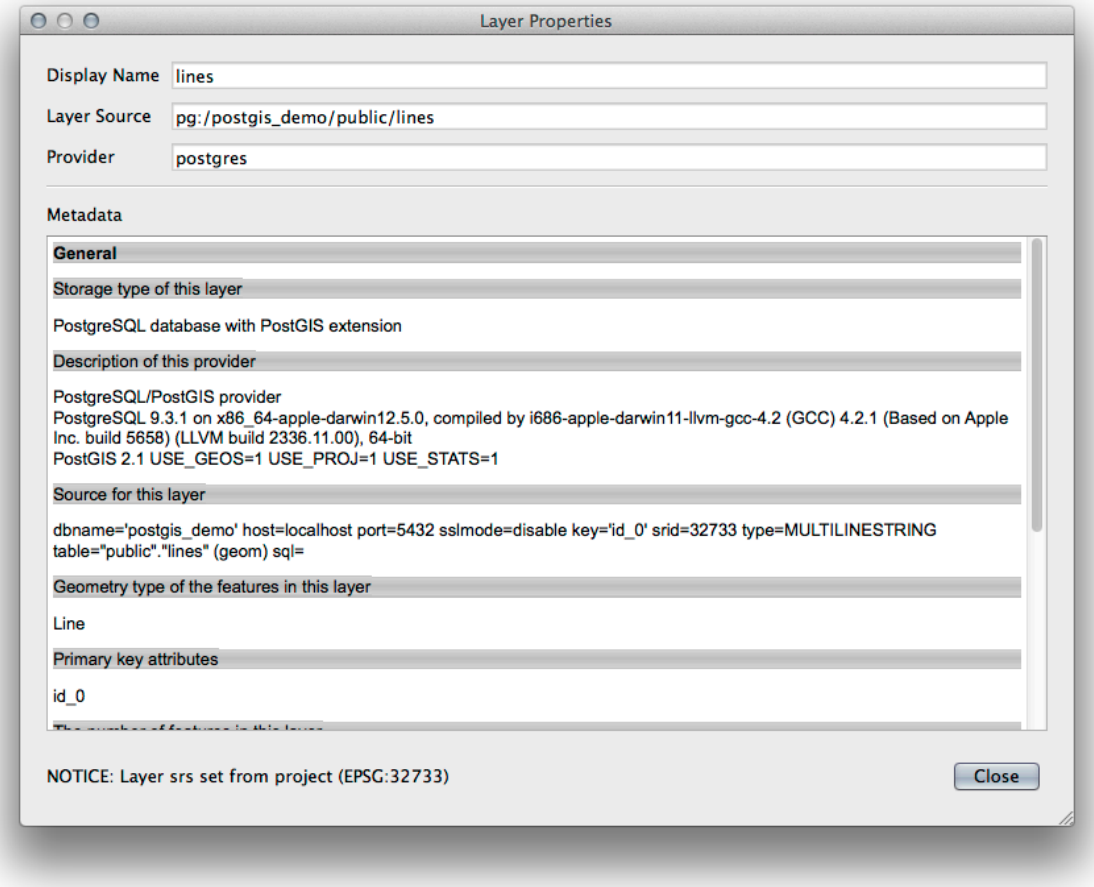
18.1.1  Follow Along: Adding Database Tables to QGIS using the Browser

We have already briefly looked at how to add tables from a database as QGIS layers, now lets look at this in a bit more detail and look at the different ways this can be done in QGIS. Lets start by looking at the new Browser interface.

- Start a new empty map in QGIS.
- Open the Browser by clicking the *Browser* tab at the bottom of the *Layer Panel*
- Open the PostGIS portion of the tree and you should find your previously configured connection available (you may need to click the Refresh button at the top of the browser window).



- Double clicking on any of the table/layers listed here will add it to the Map Canvas.
- Right Clicking on a table/layer in this view will give you a few options. Click on the *Properties* item to look at the properties of the layer.



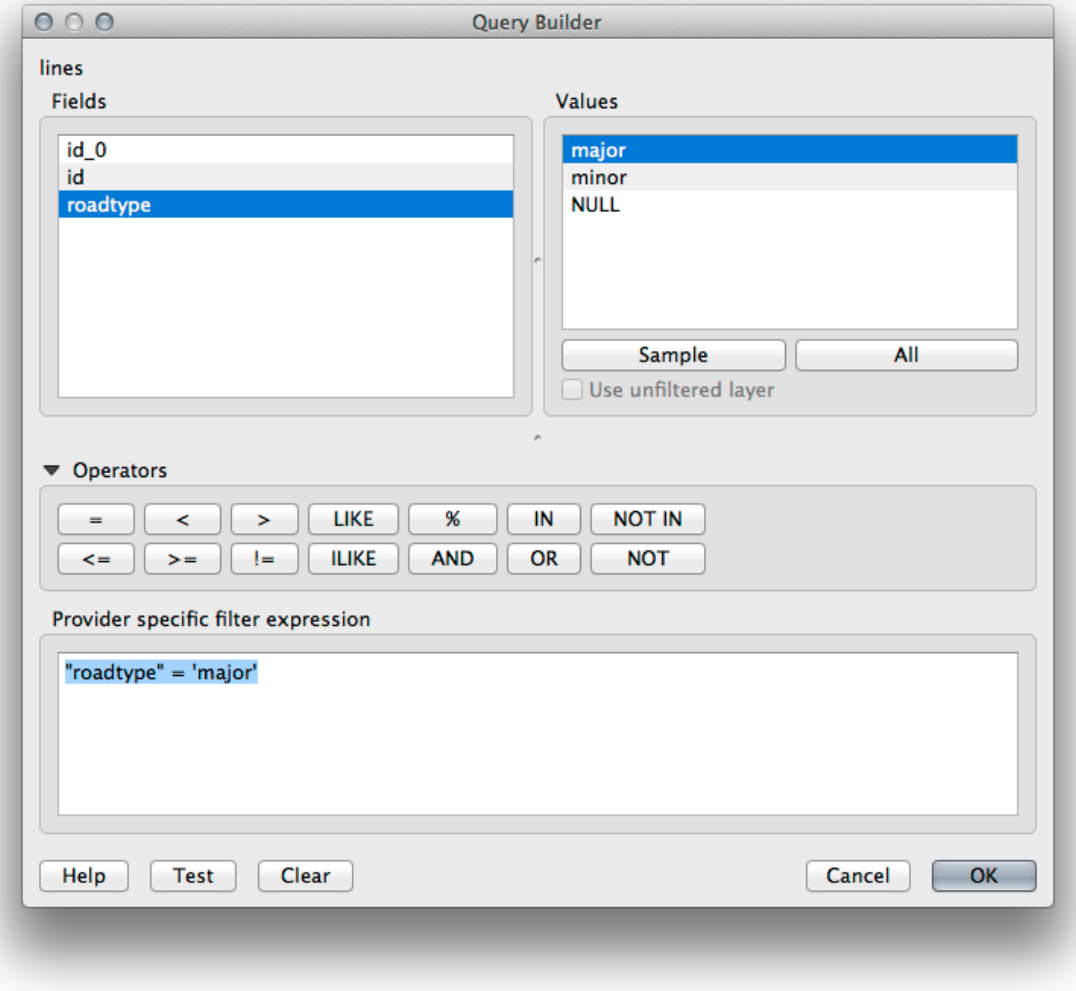
Примечание: Of course you can also use this interface to connect to PostGIS databases hosted on a server external to your workstation. Right clicking on the PostGIS entry in the tree will allow you to specify connection parameters for a new connection.

18.1.2  Follow Along: Adding a filtered set of records as a Layer

Now that we have seen how to add an entire table as a QGIS layer it might be nice to learn how to add a filtered set of records from a table as a layer by using queries that we learned about in previous sections.

- Start a new empty map with no layers
- Click the *Add PostGIS Layers* button or select *Layer* ► *Add PostGIS Layers* from the menu.
- In the *Add PostGIS Table(s)* dialog that comes up, connect to the `postgis_demo` connection.
- Expand the `public schema` and you should find the three tables we were working with previously.
- Click the `lines` layer to select it, but instead of adding it, click the *Set Filter* button to bring up the *Query Builder* dialog.
- Construct the following expression using the buttons or by entering it directly:

```
"roadtype" = 'major'
```



- Click *OK* to complete editing the filter and click *Add* to add the filtered layer to your map.
- Rename the `lines` layer in the tree to `roads_primary`.

You will notice that only the Primary Roads have been added to your map rather than the entire layer.

18.1.3 In Conclusion

You have seen how to interact with spatial databases using the QGIS Browser and how to add layers to your map based on a query filter.

18.1.4 What's Next?

Next you'll see how to work with the DB Manager interface in QGIS for a more complete set of database management tasks.

18.2 Lesson: Using DB Manager to work with Spatial Databases in QGIS

We have already seen how to perform many database operations with QGIS as well as with other tools, but now it's time to look at the DB Manager tool which provides much of this same functionality as well as more management oriented tools.

The goal for this lesson: To learn how to interact with spatial databases using the QGIS DB Manager.

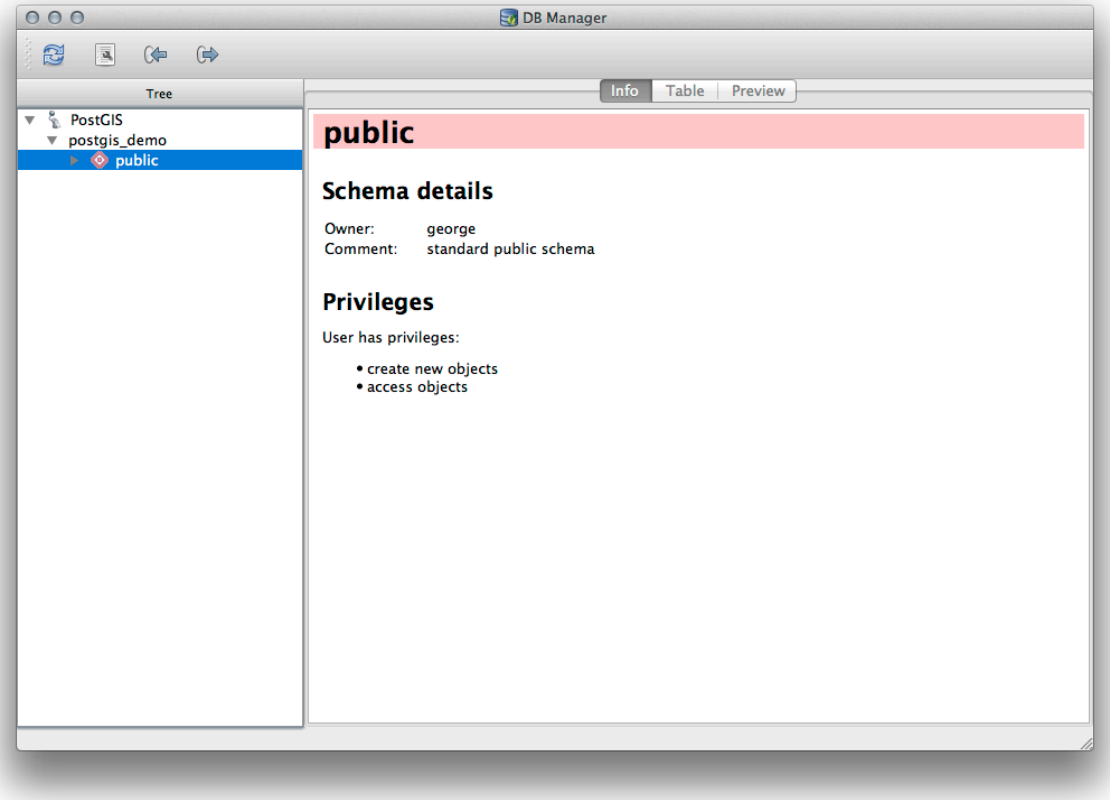
18.2.1 Follow Along: Managing PostGIS Databases with DB Manager

You should first open the DB Manager interface by selecting *Database –> DB Manager –> DB Manager* on the menu or by selecting the DB Manager icon on the toolbar.



You should already see the previous connections we have configured and be able to expand the `myPG` section and its `public` schema to see the tables we have worked with in previous sections.

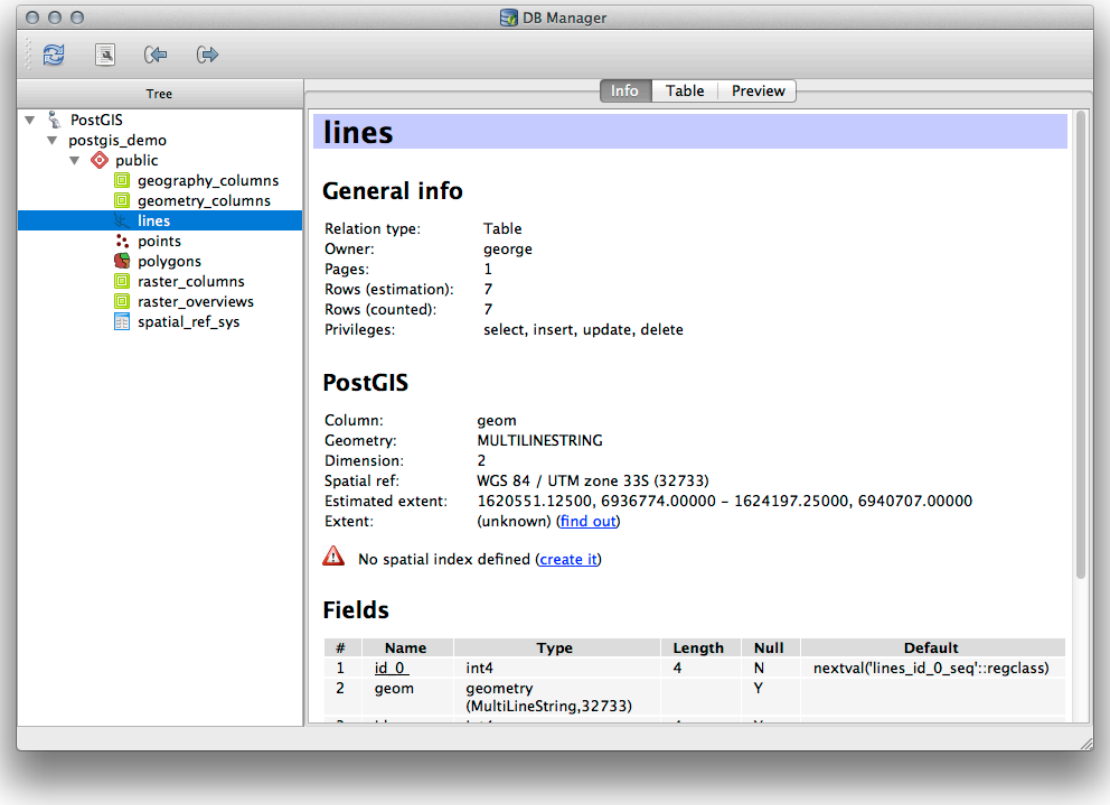
The first thing you may notice is that you can now see some metadata about the Schemas contained in your database.



Schemas are a way of grouping data tables and other objects in a PostgreSQL database and a container for permissions and other constraints. Managing PostgreSQL schemas is beyond the scope of this manual, but you can find more information about them in the PostgreSQL documentation on Schemas. You can use the DB Manager to create new Schemas, but will need to use a tool like pgAdmin III or the command line interface to manage them effectively.

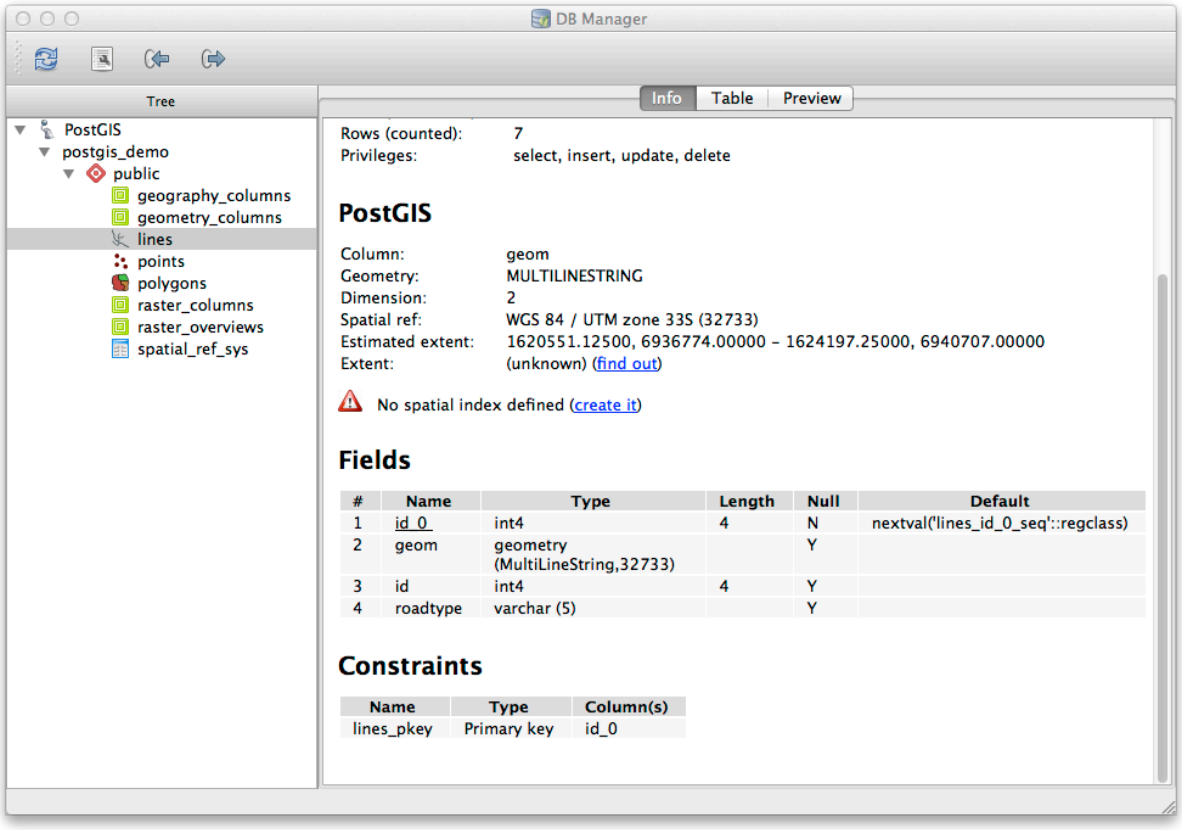
DB Manager can also be used to manage the tables within your database. We have already looked at various ways to create and manage tables on the command line, but now lets look at how to do this in DB Manager.

First, its useful to just look at a table's metadata by clicking on its name in tree and looking in the *Info* tab.

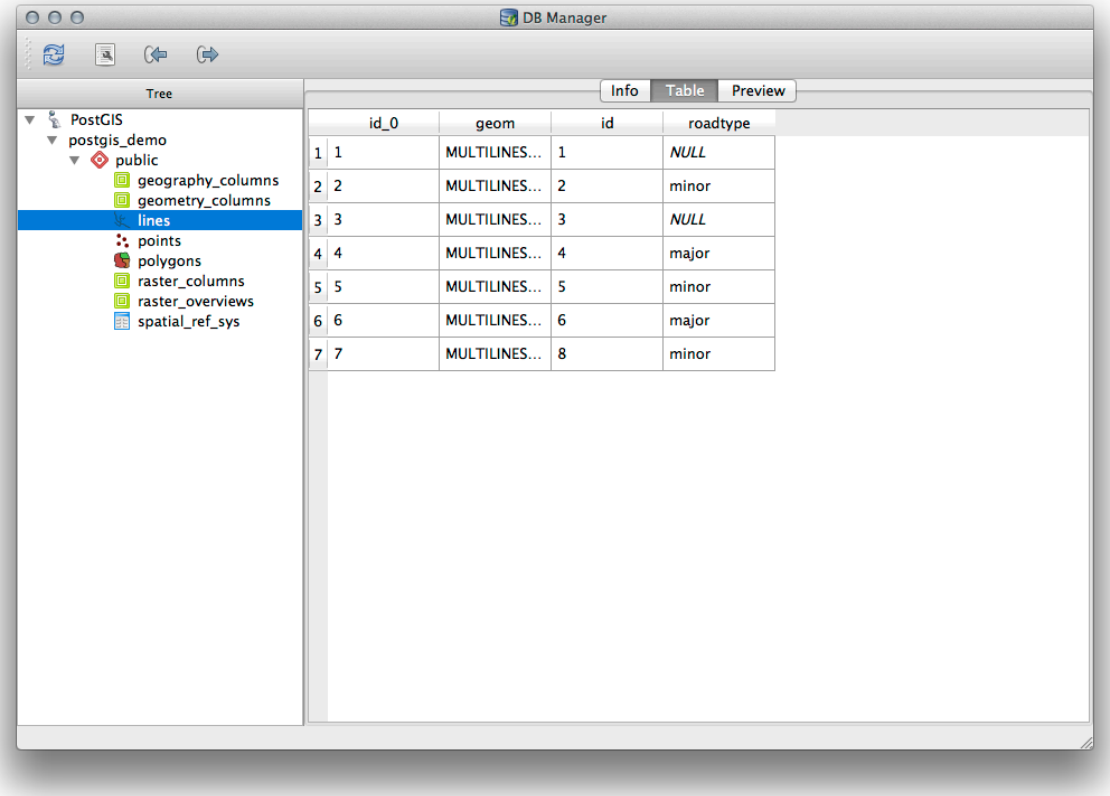


In this panel you can see the *General Info* about the table as well the information that the PostGIS extension maintains about the geometry and spatial reference system.

If you scroll down in the *Info* tab, you can see more information about the *Fields*, *Constraints* and *Indexes* for the table you are viewing.



Its also very useful to use DB Manager to simply look at the records in the database in much the same way you might do this by viewing the attribute table of a layer in the Layer Tree. You can browse the data by selecting the *Table* tab.

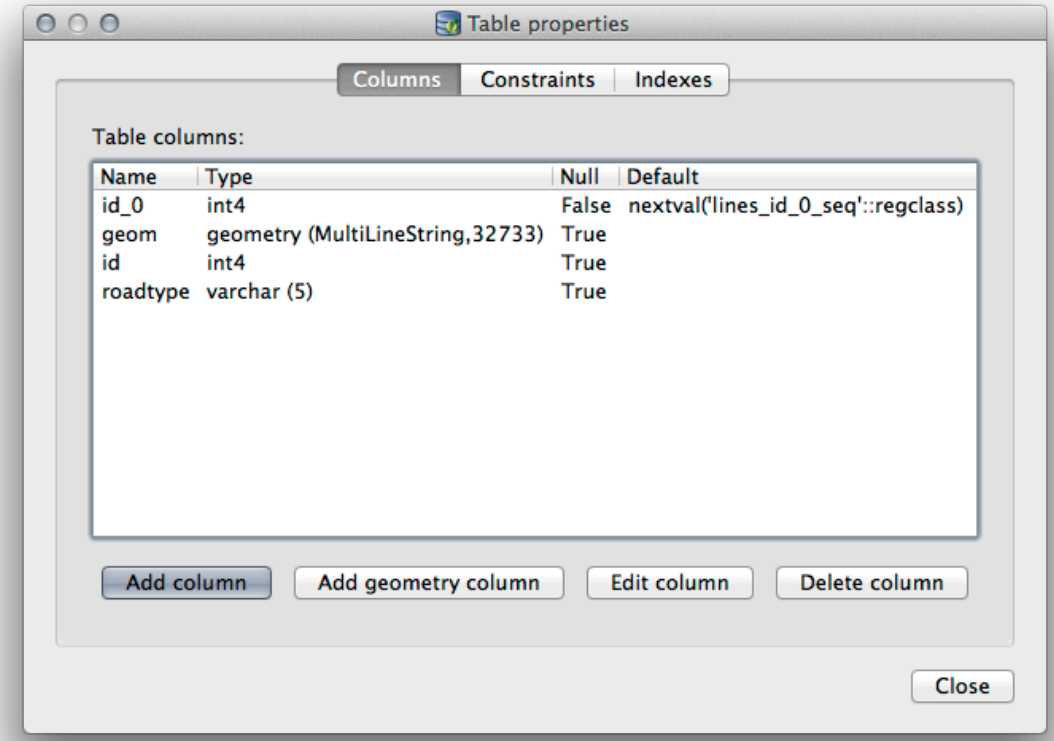


There is also a *Preview* tab which will show you the layer data in a map preview.

Right-clicking on a layer in the tree and clicking *Add to Canvas* will add this layer to your map.

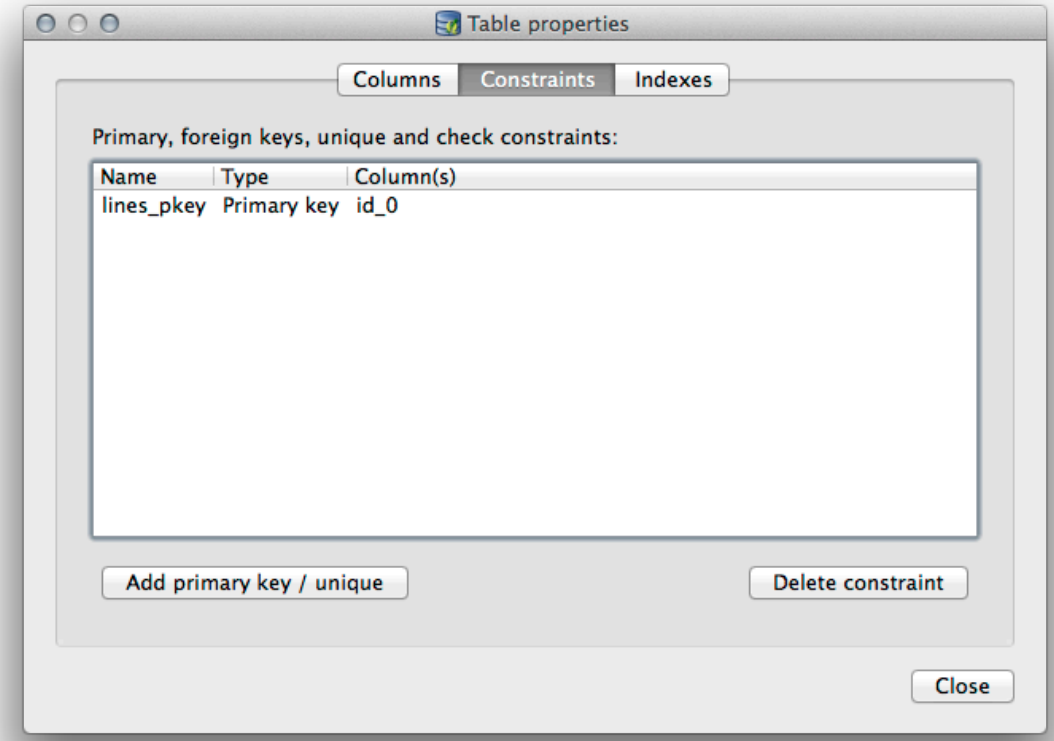
So far we have only been viewing the database its schemas and tables and their metadata, but what if we wanted to alter the table to add an additional column perhaps? DB Manager allows you to do this directly.

1. Select the table you want to edit in the tree
2. Select *Table* ► *Edit Table* from the menu, to open the *Table Properties* dialog.

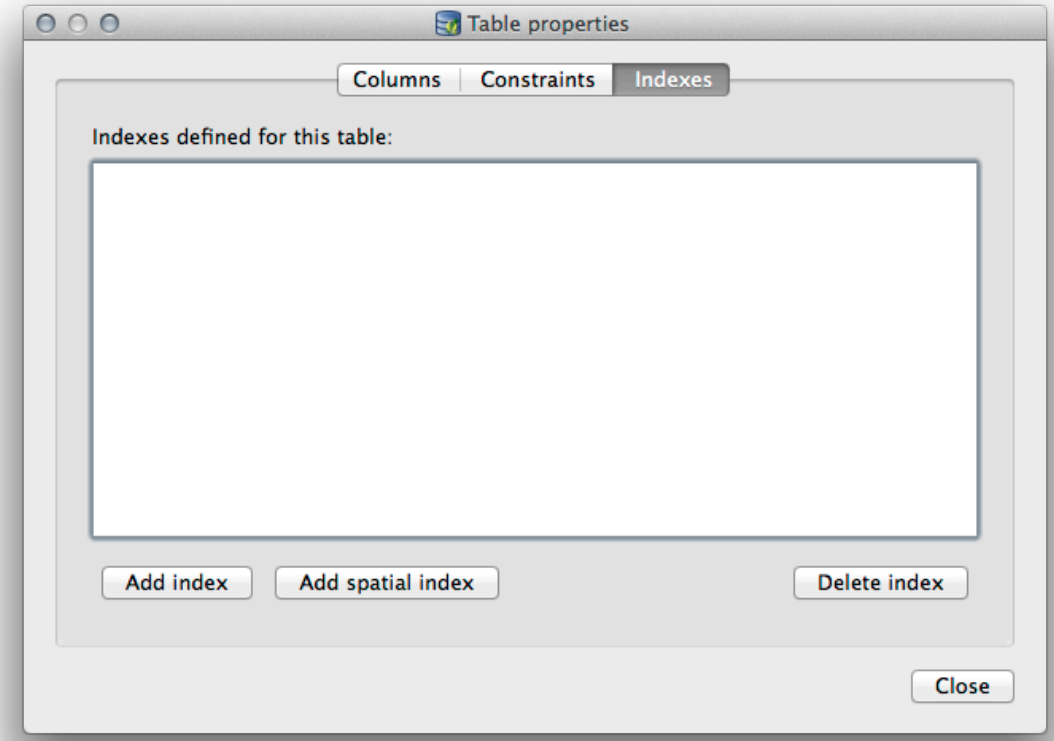


You can use this dialog to Add Columns, Add geometry columns, edit existing columns or to remove a column completely.

Using the *Constraints* tab, you can manage which fields are used as the primary key or to drop existing constraints.



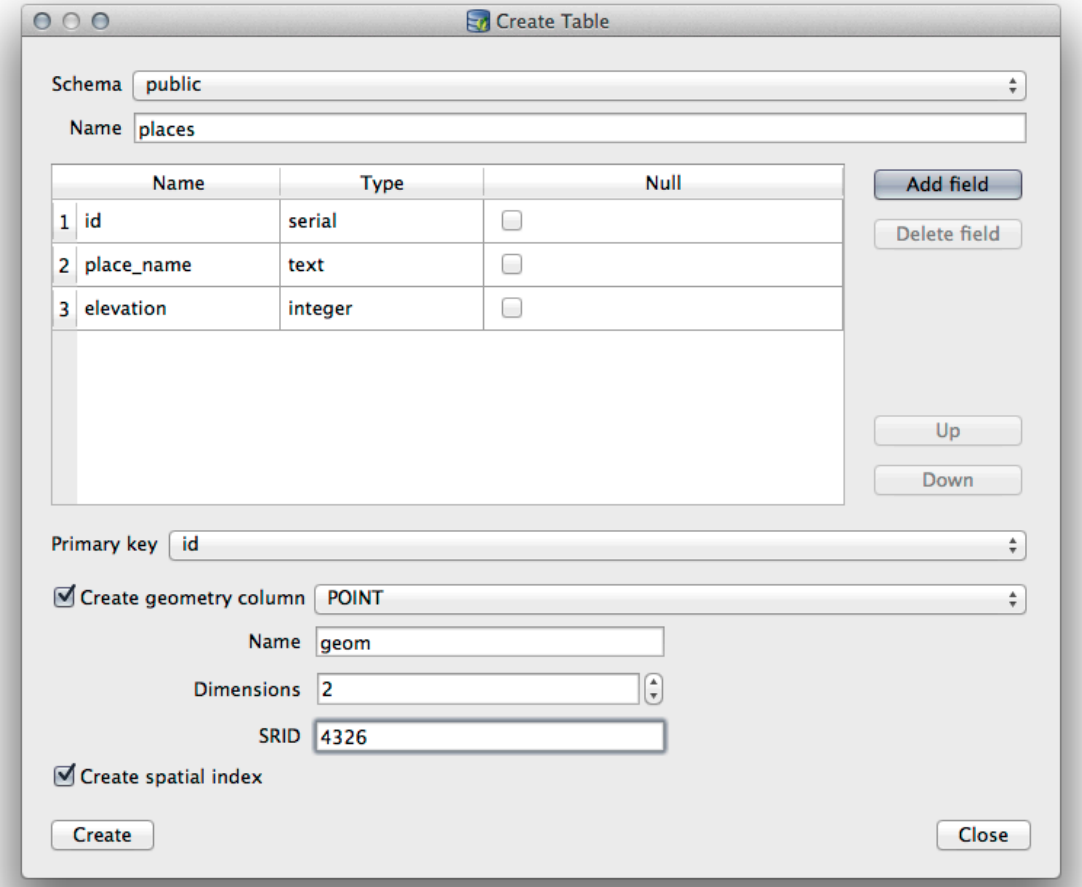
The *Indexes* tab can be used to add and delete both spatial and normal indexes.



18.2.2 Follow Along: Creating a New Table

Now that we have gone through the process of working with existing tables in our database, let's use DB Manager to create a new table.

1. If it is not already open, open the DB Manager window, and expand the tree until you see the list of tables already in your database.
2. From the menu select *Table –> Create Table* to bring up the Create Table dialog.
3. Use the default `Public` schema and name the table `places`.
4. Add the `id`, `place_name`, and `elevation` fields as shown below
5. Make sure the `id` field is set as the primary key.
6. Click the checkbox to *Create geometry column* and make sure it is set to a `POINT` type and leave it named `geom` and specify `4326` as the *SRID*.
7. Click the checkbox to *Create spatial index* and click *Create* to create the table.



8. Dismiss the dialog letting you know that the table was created and click *Close* to close the Create Table Dialog.

You can now inspect your table in the DB Manager and you will of course find that there is no data in it. From here you can *Toggle Editing* on the layer menu and begin to add places to your table.

18.2.3 Follow Along: Basic Database Administration

The DB Manager will also let you do some basic database administration tasks. It is certainly not a substitute for a more complete database administration tool, but it does provide some functionality that you can use to maintain your database.

Database tables can often become quite large and tables which are being modified frequently can end up leaving around remnants of records that are no longer needed by PostgreSQL. The *VACUUM* command takes care of doing a kind of garbage collection to compact and optional analyze your tables for better performance.

Let us take a look at how we can perform a *VACUUM ANALYZE* command from within DB Manager.

1. Select one of your tables in the DB Manager Tree
2. Select *Table* ► *Run Vacuum Analyze* from the menu

PostgreSQL will now perform the operation. Depending on how big your table is, this may take some time to complete.

You can find more information about the VACUUM ANALYZE process in the PostgreSQL Documentation on VACUUM ANALYZE.

18.2.4 Follow Along: Executing SQL Queries with DB Manager

DB Manager also provides a way for you to write queries against your database tables and to view the results. We have already seen this type of functionality in the *Browser* panel, but lets look at it again here with DB Manager.

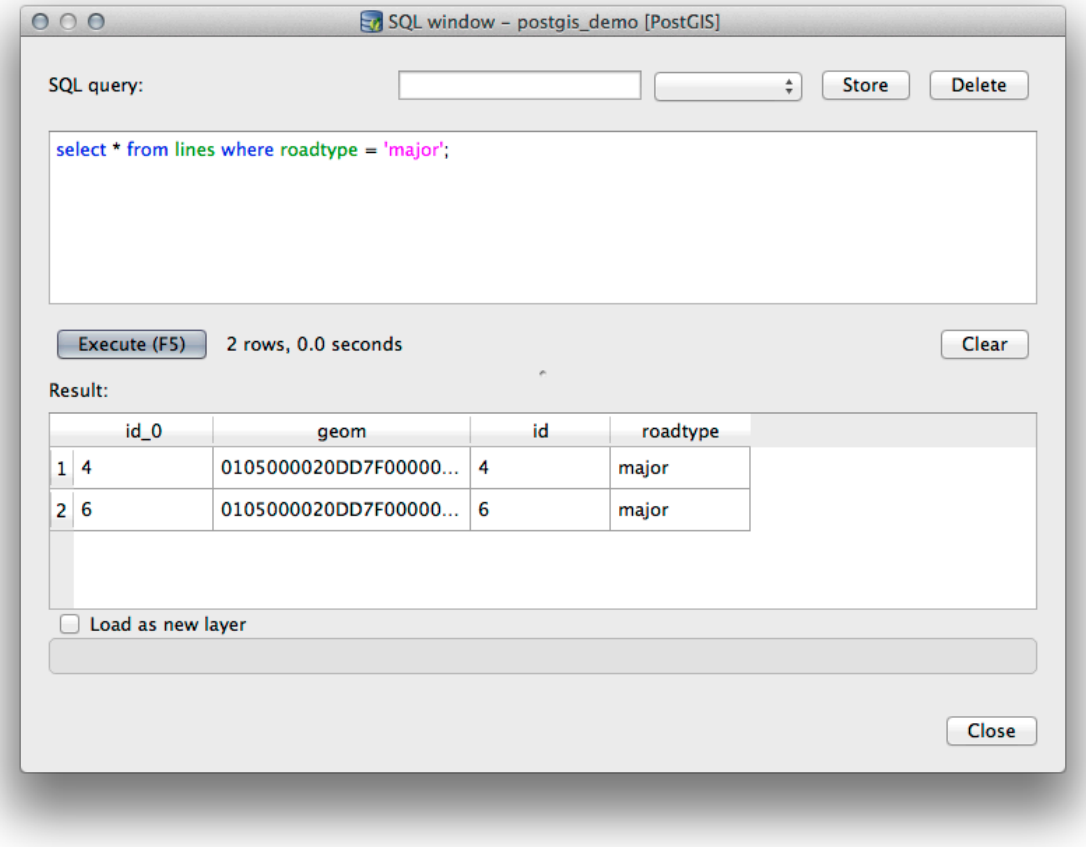
1. Select the `lines` table in the tree.
2. Select the *SQL window* button in the DB Manager toolbar.



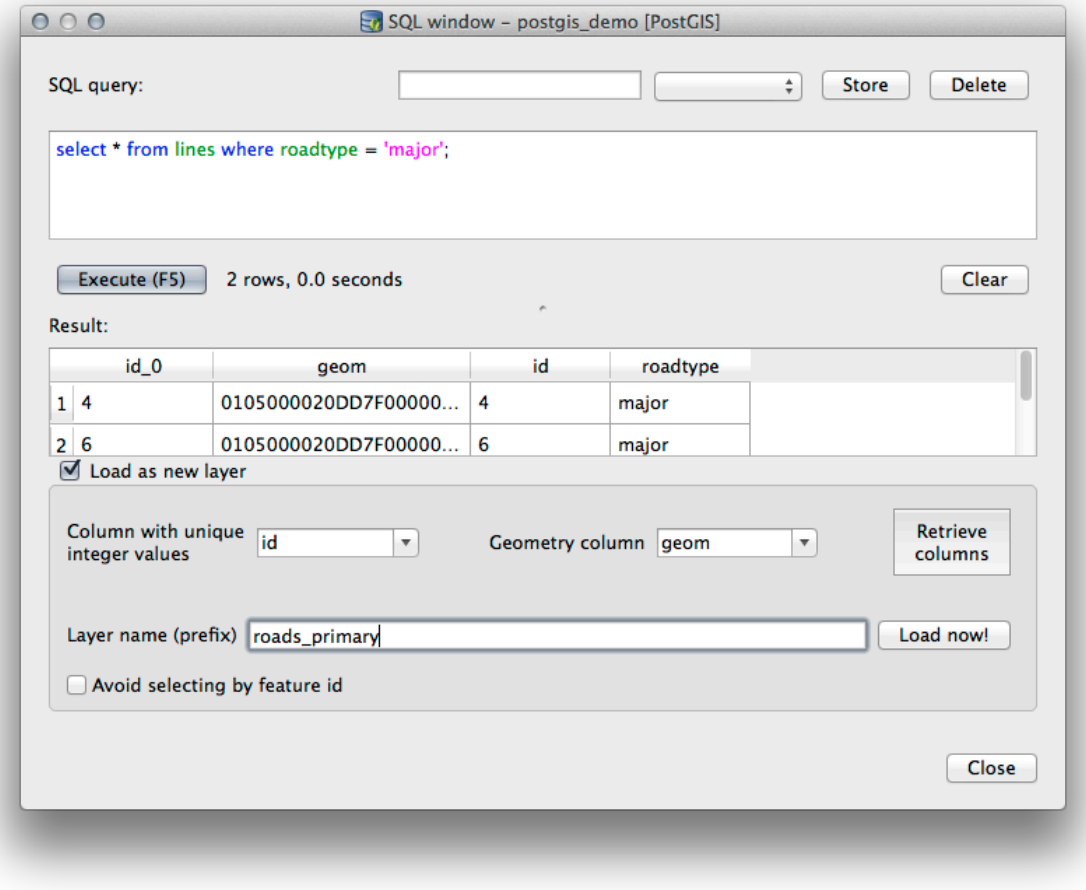
3. Compose the following *SQL query* in the space provided:

```
select * from lines where roadtype = 'major';
```

4. Click the *Execute (F5)* button to run the query.
5. You should now see the records that match in the *Result* panel.



6. Click the checkbox for *Load as new layer* to add the results to your map.
7. Select the `id` column as the *Column with unique integer values* and the `geom` column as the *Geometry column*.
8. Enter `roads_primary` as the *Layer name (prefix)*.
9. Click *Load now!* to load the results as a new layer into your map.



The layers that matched your query are now displayed on your map. You can of course use this query tool to execute any arbitrary SQL command including many of the ones we looked at in previous modules and sections.

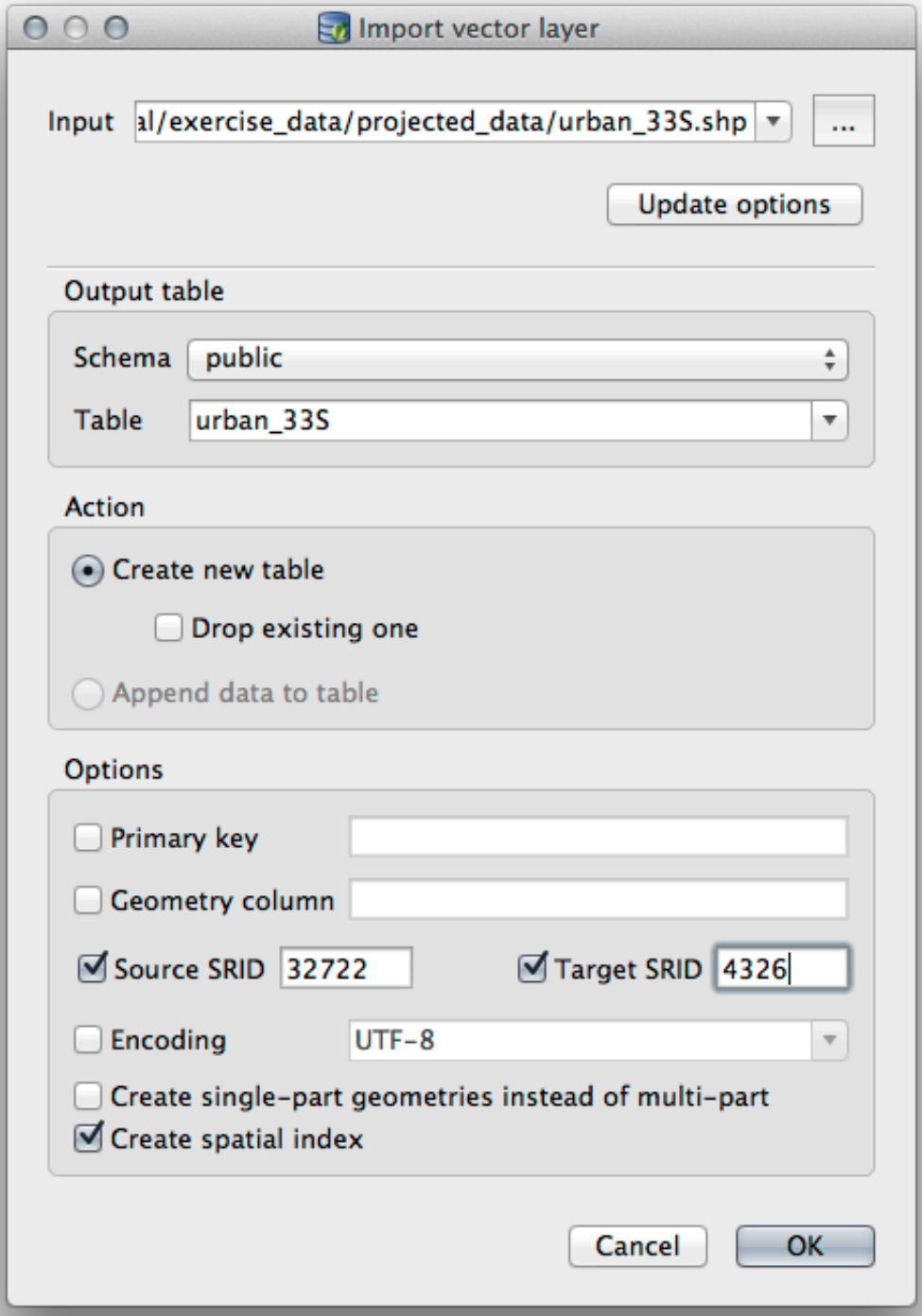
18.2.5 Importing Data into a Database with DB Manager

We have already looked at how to import data into a spatial database using command line tools, so now let's learn how to use DB Manager to do imports.

1. Click the *Import layer/file* button on the toolbar in the DB Manager dialog.

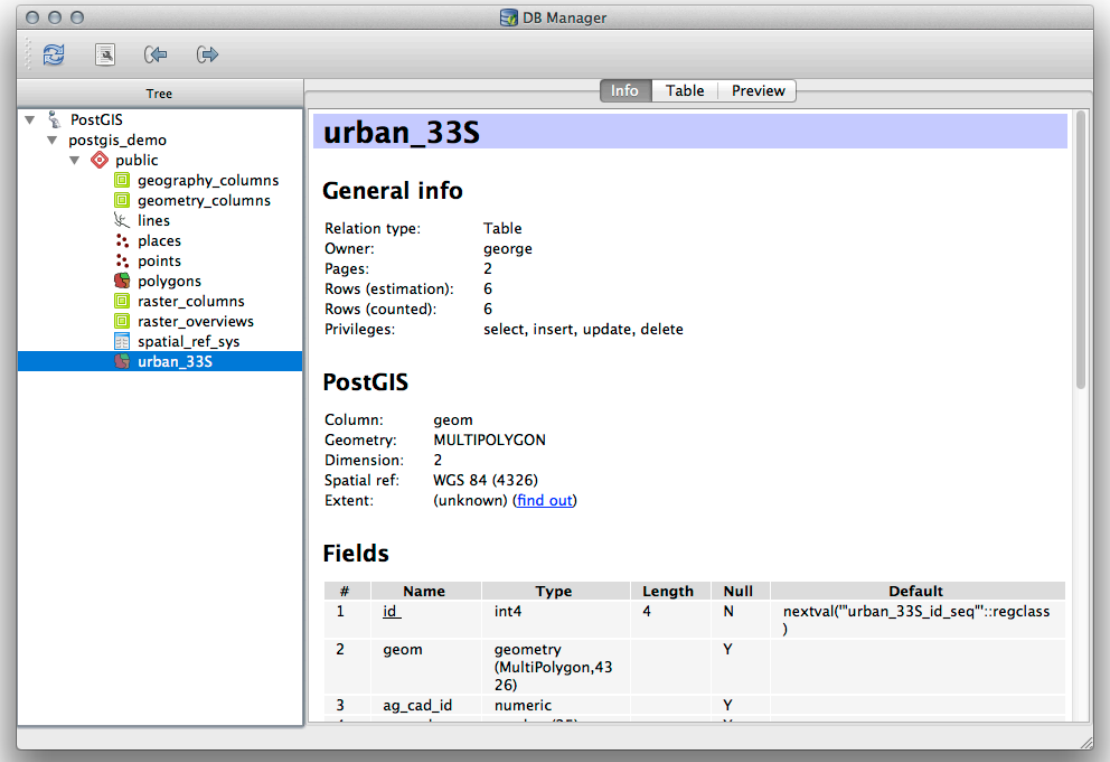


2. Select the `urban_33S.shp` file from `exercise_data/projected_data` as the input dataset
3. Click the *Update Options* button to pre-fill some of the form values.
4. Make sure that the *Create new table* option is selected
5. Specify the *Source SRID* as `32722` and the *Target SRID* as `4326`
6. Enable the checkbox to *Create Spatial Index*
7. Click *OK* to perform the import



8. Dismiss the dialog letting you know that the import was successful
9. Click the *Refresh* button on the DB Manager Toolbar

You can now inspect the table in your database by clicking on it in the Tree. Verify that the data has been reprojected by checking that the *Spatial ref:* is listed as `WGS 84 (4326)`.

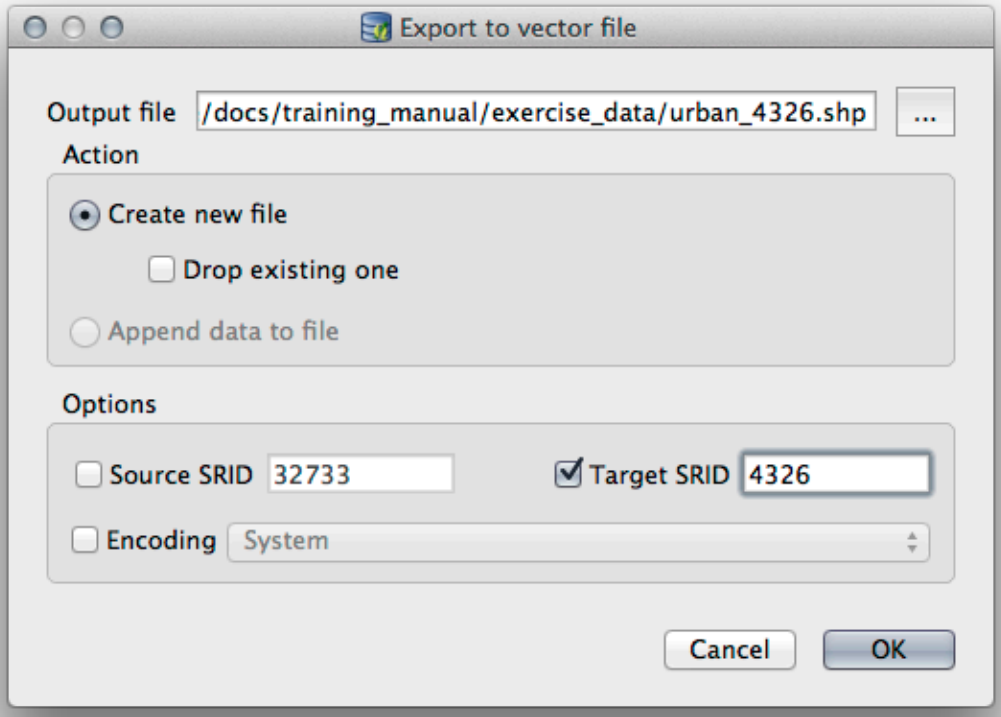


Right clicking on the table in the Tree and a selecting *Add to Canvas* will add the table as a layer in your map.

18.2.6 Exporting Data from a Database with DB Manager

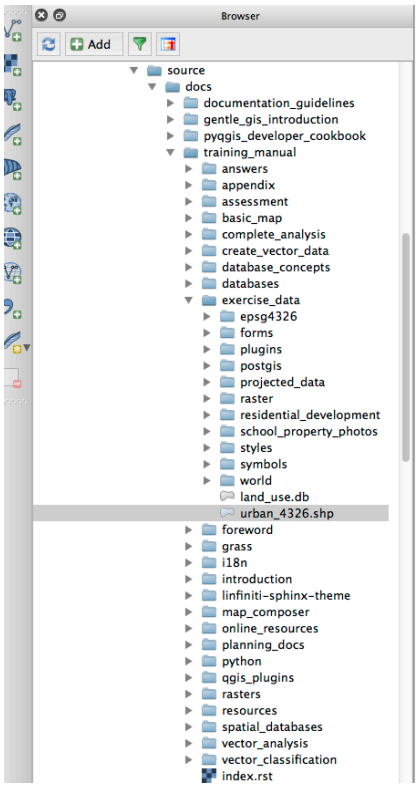
Of course DB Manager can also be used to export data from your spatial databases, so lets take a look at how that is done.

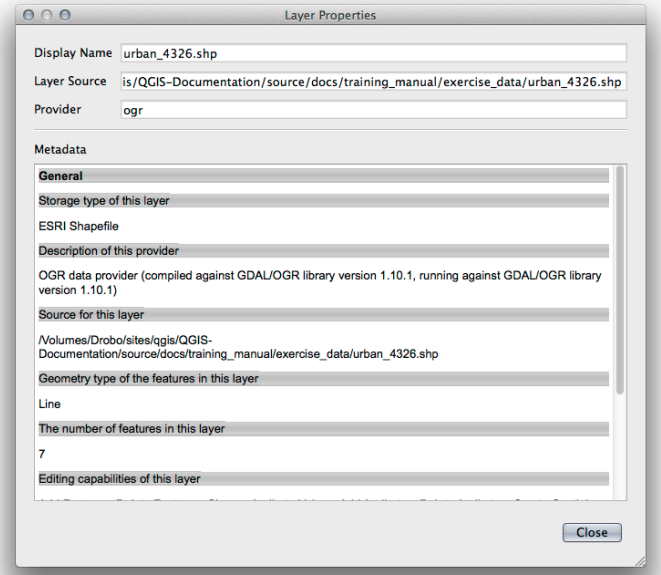
1. Select the `lines` layer in the Tree and click the *Export to File* button on the toolbar to open the *Export to vector file* dialog.
2. Click the … button to select the *Output file* and save the data to your `exercise_data` directory as `urban_4326`.
3. Set the *Target SRID* as `4326`.
4. Click *OK* to initialize the export.



5. Dismiss the dialog letting you know the export was successful and close the DB Manager.

You can now inspect the shapefile you created with the Browser panel.





18.2.7 In Conclusion

You have now seen how to use the DB Manager interface in QGIS to manage your spatial databases, to execute SQL queries against your data and how to import and export data.

18.2.8 What's Next?

Next, we will look at how to use many of these same techniques with *SpatiaLite* databases.

18.3 Lesson: Working with SpatiaLite databases in QGIS

While PostGIS is generally used on a server to provide spatial database capabilities to multiple users at the same time, QGIS also supports the use of a file format called *SpatiaLite* that is a lightweight, portable way to store an entire spatial database in a single file. Obviously, these 2 types of spatial databases should be used for different purposes, but the same basic principles and techniques apply to both. Let's create a new SpatiaLite database and explore the functionality provided to work with these databases in QGIS.

The goal for this lesson: To learn how to interact with SpatiaLite databases using the QGIS Browser interface.

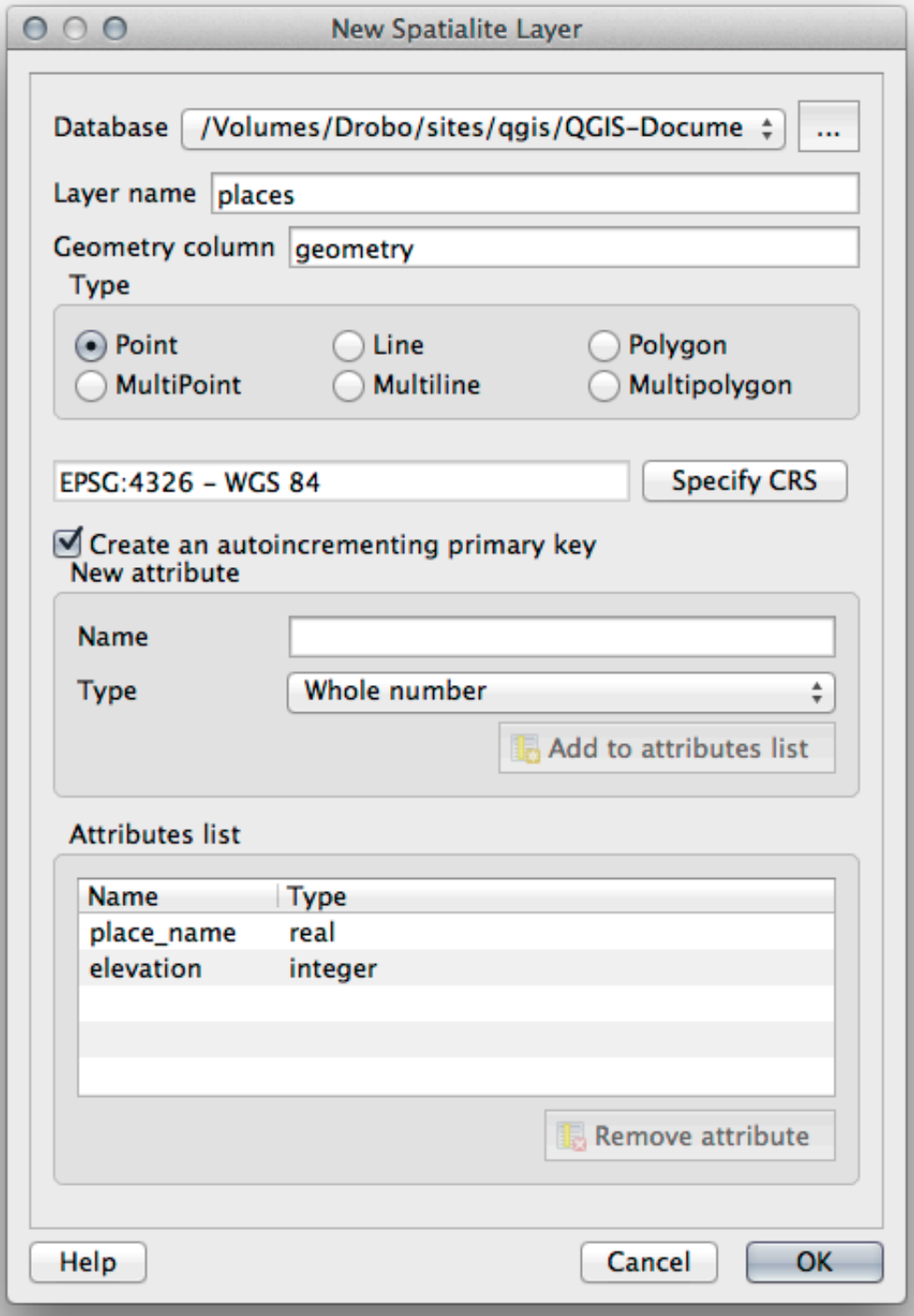
18.3.1  Follow Along: Creating a SpatiaLite database with the Browser

Using the Browser panel, we can create a new SpatiaLite database and get it setup for use in QGIS.

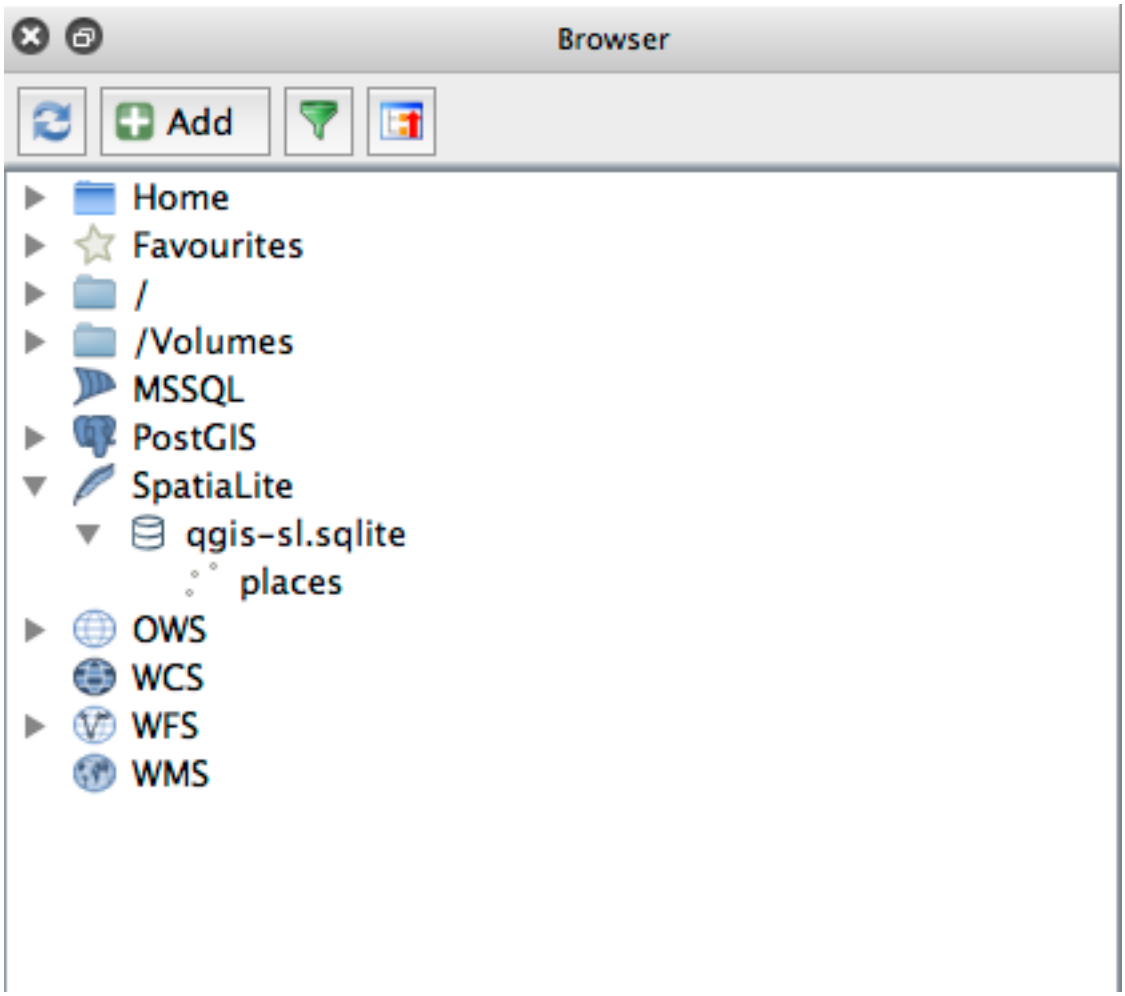
1. Right click on the *SpatiaLite* entry in the Browser tree and select *Create Database*.
2. Specify where on your filesystem you want to store the file and name it `qgis-sl.db`.
3. Again right click on the *SpatiaLite* entry in the Browser tree and now select the *New Connection* item. Find the file you created in the last step and open it.

Now that you have configured your new database you will find that the entry in Browser tree has nothing underneath it and the only thing you can do at this point is to delete the connection. This is of course because we haven't added any tables to this database. Let's go ahead and do that.

1. Find the button to create a new layer and use the dropdown to create a new SpatiaLite layer, or select *Layer* ► *New* ►  *New SpatiaLite Layer*.
2. Select the database we created in the previous steps in the drop down.
3. Give the layer the name `places`.
4. Tick the checkbox next to *Create an auto-incrementing primary key*.
5. Add two attributes as shown in below
6. Click *OK* to create the table.



7. Click the refresh button at the top of the Browser and you should now see your `places` table listed.



You can right click on the table and view its properties as we did in the previous exercise.

From here you can start an editing session and start adding data to your new database directly.

We also learned about how to import data into a database using the DB Manager and you can use this same technique to import data into your new SpatiaLite DB.

18.3.2 In Conclusion

You have seen how to create SpatiaLite databases, add tables to them and use these tables as layers in QGIS.

ГЛАВА 19

Appendix: Contributing To This Manual

To add materials to this course, you must follow the guidelines in this Appendix. You are not to alter the conditions in this Appendix except for clarification. This is to ensure that the quality and consistency of this manual can be maintained.

19.1 Downloading Resources

The source of this document is available at GitHub. Consult GitHub.com for instructions on how to use the git version control system.

19.2 Manual Format

This manual is written using Sphinx, a Python document generator using the reStructuredText markup language. Instructions on how to use these tools are available on their respective sites.

19.3 Adding a Module

- To add a new module, first create a new directory (directly under the top-level of the `qgis-training-manual` directory) with the name of the new module.
- Under this new directory, create a file called `index.rst`. Leave this file blank for now.
- Open the `index.rst` file under the top-level directory. Its first lines are:

```
.. toctree::
   :maxdepth: 2


   foreword/index
   introduction/index
```

You will note that this is a list of directory names, followed by the name `index`. This directs the top-level index file to the index files in each directory. The order in which they are listed determines the order they will have in the document.

- Add the name of your new module (i.e., the name you gave the new directory), followed by `/index`, to this list, wherever you want your module to appear.
- Remember to keep the order of the modules logical, such that later modules build on the knowledge presented in earlier modules.
- Open your new module's own index file (`[module name]/index.rst`).
- Along the top of the page, write a line of 80 asterisks (`*`). This represents a module heading.
- Follow this with a line containing the markup phrase `|MOD|` (which stands for «module»), followed by the name of your module.
- End this off with another line of 80 asterisks.
- Leave a line open beneath this.
- Write a short paragraph explaining the purpose and content of the module.
- Leave one line open, then add the following text:

```
.. toctree::
   :maxdepth: 2


   lesson1
   lesson2
```

… where `lesson1`, `lesson2`, etc., are the names of your planned lessons.

The module-level index file will look like this:

```
*******************************************************************************
|MOD| Module Name
*******************************************************************************

Short paragraph describing the module.

.. toctree::
   :maxdepth: 2


   lesson1
   lesson2
```

19.4 Adding a Lesson

To add a lesson to a new or existing module:

- Open the module directory.
- Open the `index.rst` file (created above in the case of new modules).
- Ensure that the name of the planned lesson is listed underneath the `toctree` directive, as shown above.
- Create a new file under the module directory.
- Name this file exactly the same as the name you provided in the module's `index.rst` file, and add the extension `.rst`.

Примечание: For editing purposes, a `.rst` file works exactly like a normal text file (`.txt`).

- To begin writing the lesson, write the markup phrase `|LS|`, followed by the lesson name.

- In the next line, write a line of 80 equal signs (=).
- Leave a line open after this.
- Write a short description of the lesson's intended purpose.
- Include a general introduction to the subject matter. See the existing lessons in this manual for examples.
- Beneath this, start a new paragraph, beginning with this phrase:

```
**The goal for this lesson:**
```

- Briefly explain the intended outcome of completing this lesson.
- If you can't describe the goal of the lesson in one or two sentences, consider breaking the subject matter up into multiple lessons.

Each lesson will be subdivided into multiple sections, which will be addressed next.

19.5 Adding a Section

There are two types of sections: «follow along» and «try yourself».

- A «follow along» section is a detailed set of directions intended to teach the reader how to use a given aspect of QGIS. This is typically done by giving click-by-click directions as clearly as possible, interspersed with screenshots.
- A «try yourself» section gives the reader a short assignment to try by themselves. It is usually associated with an entry in the answer sheet at the end of the documentation, which will show or explain how to complete the assignment, and will show the expected outcome if possible.

Every section comes with a difficulty level. An easy section is denoted by `|basic|`, moderate by `|moderate|`, and advanced by `|hard|`.

19.5.1 Adding a «follow along» section

- To start this section, write the markup phrase of the intended difficulty level (as shown above).
- Leave a space and then write `|FA|` (for «follow along»).
- Leave another space and write the name of the section (use only an initial capital letter, as well as capitals for proper nouns).
- In the next line, write a line of 80 minuses/dashes (−). Ensure that your text editor does not replace the default minus/dash character with a long dash or other character.
- Write a short introduction to the section, explaining its purpose. Then give detailed (click-by-click) instructions on the procedure to be demonstrated.
- In each section, include internal links, external links and screenshots as needed.
- Try to end each section with a short paragraph that concludes it and leads naturally to the next section, if possible.

19.5.2 Adding a «try yourself» section

- To start this section, write the markup phrase of the intended difficulty level (as shown above).
- Leave a space and then write `|TY|` (for «try yourself»).
- In the next line, write a line of 80 minuses/dashes (−). Ensure that your text editor does not replace the default minus/dash character with a long dash or other character.
- Explain the exercise that you want the reader to complete. Refer to previous sections, lessons or modules if necessary.
- Include screenshots to clarify the requirements if a plain textual description is not clear.

In most cases, you will want to provide an answer regarding how to complete the assignment given in this section. To do so, you will need to add an entry in the answer sheet.

- First, decide on a unique name for the answer. Ideally, this name will include the name of the lesson and an incrementing number.
- Create a link for this answer:

```
:ref:`Check your results <answer-name>`
```

- Open the answer sheet (`answers/answers.rst`).
- Create a link to the «try yourself» section by writing this line:

```
.. _answer-name:
```

- Write the instructions on how to complete the assignment, using links and images where needed.
- To end it off, include a link back to the «try yourself» section by writing this line:

```
:ref:`Back to text <backlink-answer-name>`
```

- To make this link work, add the following line above the heading to the «try yourself» section:

```
.. _backlink-answer-name:
```

Remember that each of these lines shown above must have a blank line above and below it, otherwise it could cause errors while creating the document.

19.6 Add a Conclusion

- To end a lesson, write the phrase `|IC|` for «in conclusion», followed by a new line of 80 minuses/dashes (−). Write a conclusion for the lesson, explaining which concepts have been covered in the lesson.

19.7 Add a Further Reading Section

- This section is optional.
- Write the phrase `FR` for «further reading», followed by a new line of 80 minuses/dashes (−).
- Include links to appropriate external websites.

19.8 Add a What's Next Section

- Write the phrase `|WN|` for «what's next», followed by a new line of 80 minuses/dashes (−).
- Explain how this lesson has prepared students for the next lesson or module.
- Remember to change the «what's next» section of the previous lesson if necessary, so that it refers to your new lesson. This will be necessary if you have inserted a new lesson among existing lessons, or after an existing lesson.

19.9 Using Markup

To adhere to the standards of this document, you will need to add standard markup to your text.

19.9.1 New concepts

- If you are explaining a new concept, you will need to write the new concept's name in italics by enclosing it in asterisks (`*`).

```
This sample text shows how to introduce a *new concept*.
```

19.9.2 Emphasis

- To emphasize a crucial term which is not a new concept, write the term in bold by enclosing it in double asterisks (`**`).
- Use this sparingly! If used too much, it can seem to the reader that you are shouting or being condescending.

```
This sample text shows how to use **emphasis** in a sentence. Include the
punctuation mark if it is followed by a **comma,** or at the **end of the
sentence.**
```

19.9.3 Images

- When adding an image, save it to the folder `_static/lesson_name/`.
- Include it in the document like this:

```
.. figure:: img/image_file.extension
   :align: center
```

- Remember to leave a line open above and below the image markup.

19.9.4 Internal links

- To create an anchor for a link, write the following line above the place where you want the link to point to:

```
.. _link-name:
```

- To create a link, add this line:

```
:ref:`Descriptive link text <link-name>`
```

- Remember to leave a line open above and below this line.

19.9.5 External links

- To create an external link, write it out like this:

```
`Descriptive link text <link-url>`_
```

- Remember to leave a line open above and below this line.

19.9.6 Using monospaced text

- When you are writing text that the user needs to enter, a path name, or the name of a database element such as a table or column name, you must write it in `monospaced text`. For example:

```
Enter the following path in the text box: :kbd:`path/to/file`.
```

19.9.7 Labeling GUI items

- If you are referring to a GUI item, such as a button, you must write its name in *the GUI label format*. For example:

```
To access this tool, click on the :guilabel:`Tool Name` button.
```

- This also applies if you are mentioning the name of a tool without requiring the user to click a button.

19.9.8 Menu selections

- If you are guiding a user through menus, you must use the *menu* ► *selection* ► *format*. For example:

```
To use the :guilabel:`Tool Name` tool, go to :menuselection:`Plugins -->
Tool Type --> Tool Name`.
```

19.9.9 Adding notes

- You might need to a note in the text, which explains extra details that can't easily be made part of the flow of the lesson. This is the markup:

```
[Normal paragraph.]

.. note:: Note text.
   New line within note.

   New paragraph within note.

[Unindented text resumes normal paragraph.]
```

19.9.10 Adding a sponsorship/authorship note

If you are writing a new module, lesson or section on behalf of a sponsor, you must include a short sponsor message of their choice. This must notify the reader of the name of the sponsor and must appear below the heading of the module, lesson or section that they sponsored. However, it may not be an advertisement for their company.

If you have volunteered to write a module, lesson or section in your own capacity, and not on behalf of a sponsor, you may include an authorship note below the heading of the module, lesson or section that you authored. This must take the form `This [module/lesson/section] contributed by [author name]`. Do not add further text, contact details, etc. Such details are to be added in the «Contributors» section of the Foreword, along with the name(s) of the part(s) you added. If you only made enhancements, corrections and/or additions, list yourself as an editor.

19.10 Thank You!

Thank you for contributing to this project! By so doing, you are making QGIS more accessible to users and adding value to the QGIS project as a whole.

ГЛАВА 20

Подготовка данных для занятий

Примечание: Этот процесс направлен на пользователей, уже имеющих опыт с QGIS, которые хотят создать локализованные наборы данных для их курса. Стандартные наборы данных предоставляются вместе с руководством, но Вы можете заменить их собственными, следуя этим инструкциям.

The *sample data provided* with the Training Manual refers to the town of Swellendam and its surroundings. Swellendam is located about 2 hours“ east of Cape Town in the Western Cape of South Africa. The dataset contains feature names in both English and Afrikaans.

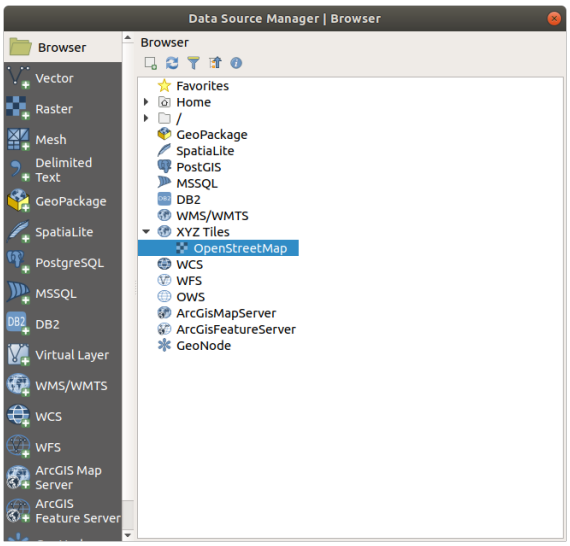
Каждый может без проблем использовать эти данные, но при желании Вы можете использовать данные для Вашей страны или города в модулях 3-7.2. Дальнейшие модули используют более сложные источники данных, которых для Вашего региона может просто не оказаться.

Примечание: Эти инструкции подразумевают, что у Вы уже хорошо знаете QGIS, и не предлагаются в качестве учебного материала.

20.1  Try Yourself Create OSM based vector Files

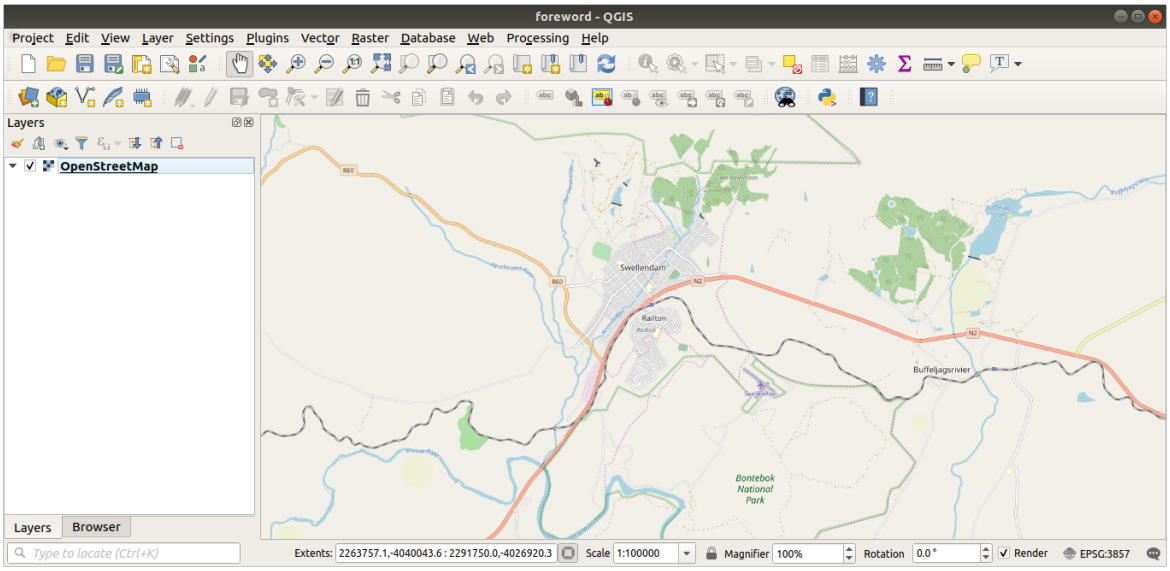
Если Вы желаете заменить стандартный набор данных локализованым набором для Вашего курса, это можно легко сделать инструментами QGIS. Выбранный Вами регион должен включать городские и сельские зоны с дорогами, границы участков (таких как заповедники и фермы) и водоёмы.

1. Откройте новый проект QGIS
2. Select *Layer* ► *Data Source Manager* to open the *Data Source Manager* dialog
3. In the *Browser* tab, expand the *XYZ Tiles* drop-down menu and double-click the *OpenStreetMap* item.



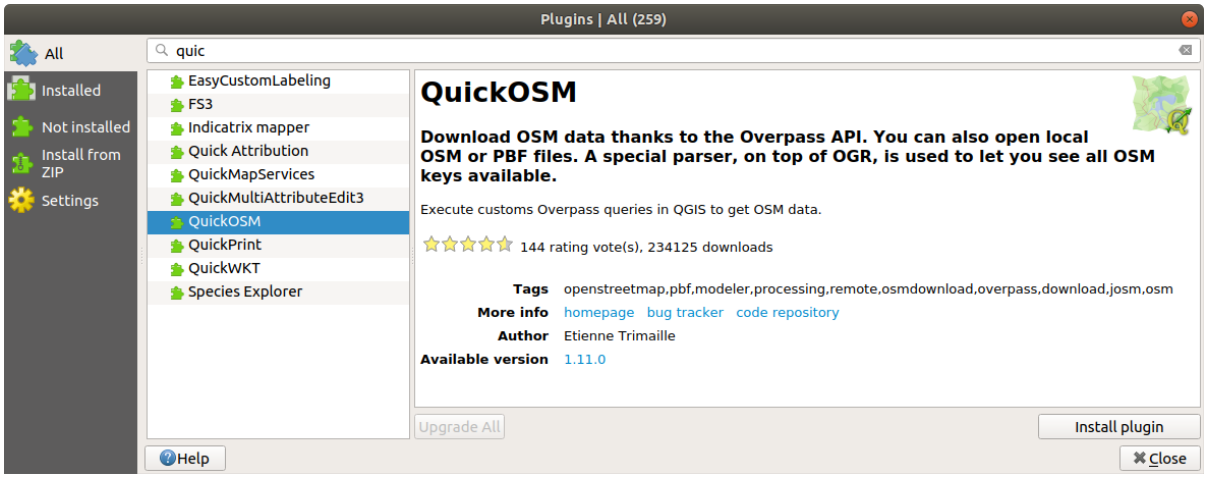
A map of the world is now visible on the map canvas.

4. Close the *Data Source Manager* dialog
5. Move to the area you'd like to use as study area

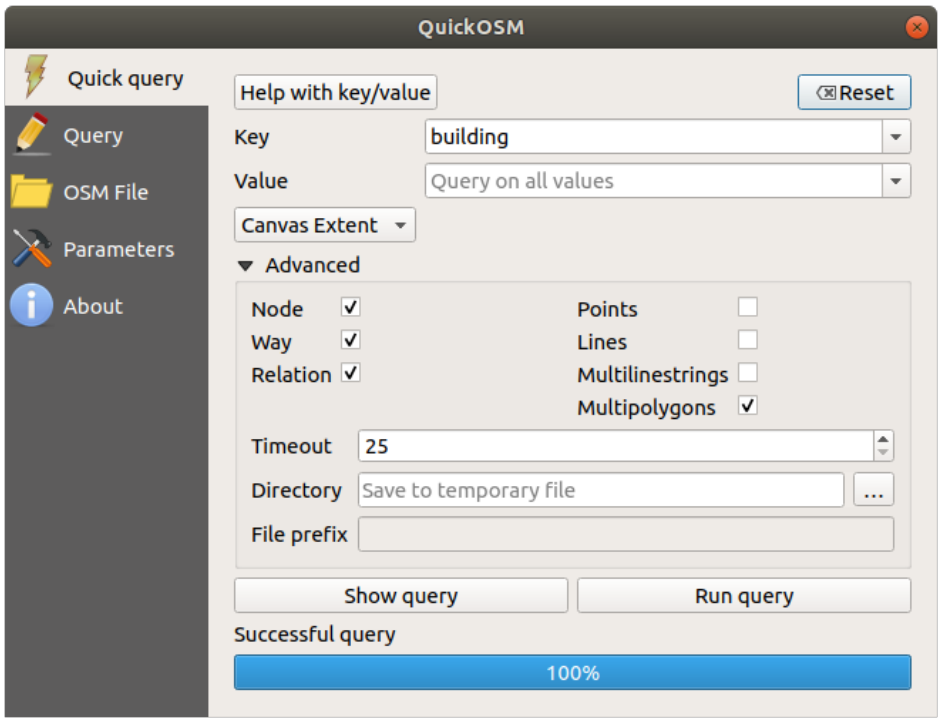


Now that we have the area we'll extract the data from, let's enable the extraction tools.

1. Go to *Plugins* ► *Manage/Install Plugins…*
2. In the *All* tab, type `QuickOSM` in the search box
3. Select the QuickOSM plugin, press *Install Plugin* and then *Close* the dialog.



4. Execute the new plugin from *Vector* ► *QuickOSM* ► *QuickOSM…* menu
5. In the *Quick query* tab, select `building` in the *Key* drop-down menu
6. Leave the *Value* field empty, meaning that you are querying all buildings.
7. Select *Canvas Extent* in the next drop-down menu
8. Expand the *Advanced* group below and uncheck all geometry types on the right except *Multipolygons*.
9. Press *Run query*

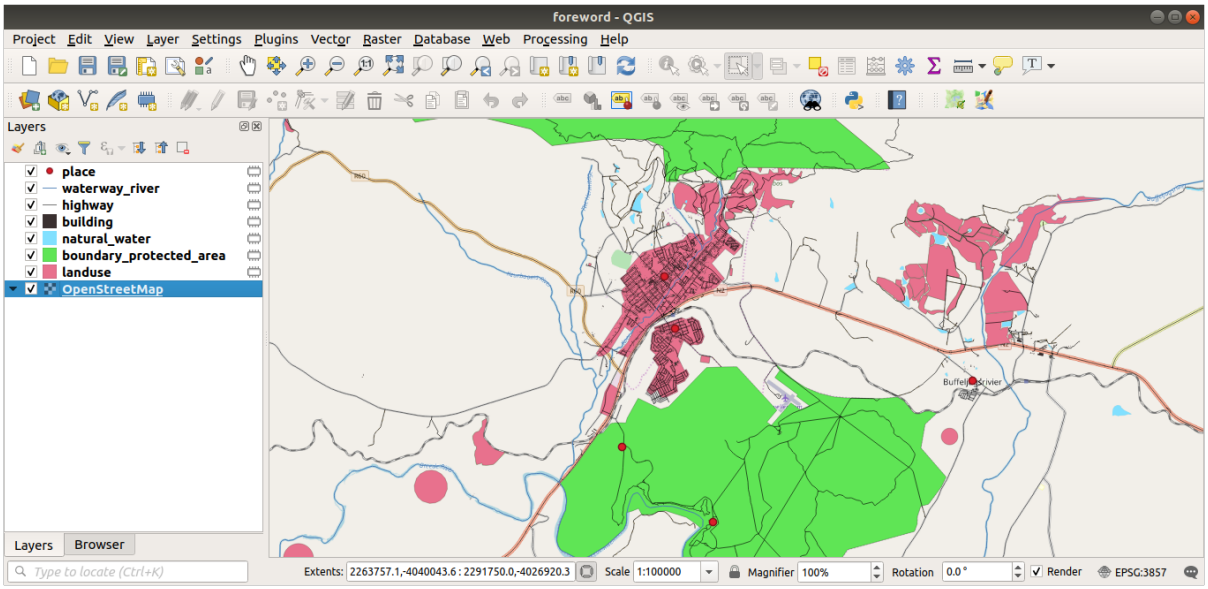


A new `building` layer is added to the *Layers* panel, showing buildings in the selected extent.

10. Proceed as above to extract other data:
    1. `Key = landuse` and `Multipolygons` geometry type.
    2. `Key = boundary`, `Value = protected_area` and `Multipolygons` geometry type.
    3. `Key = natural`, `Value = water` and `Multipolygons` geometry type.
    4. `Key = highway` and check `Lines` and `Multilines` geometry types.

5. `Key = waterway`, `Value = river` and check `Lines` and `Multilines` geometry types.
6. `Key = place` and `Points` geometry type.

This process adds the layers as temporary files (indicated by the icon next to their name).



You can sample the data your region contains in order to see what kind of results your region will yield.

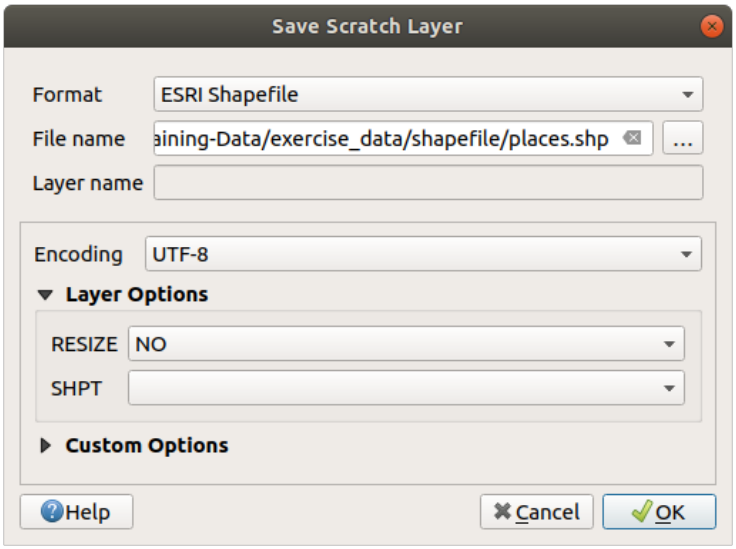
We now need to save the resulting data to use during your course. We'll be using ESRI Shapefile, GeoPackage and SpatiaLite formats depending on the data.

To convert the *place* temporary layer to another format:

1. Click the icon next to the *place* layer to open the *Save Scratch Layer* dialog.

   Примечание: If you need to change any of the temporary layer's properties (CRS, extent, fields…), use the *Export ► Save Features as…* contextual menu instead, and ensure the *Add saved file to map* option is checked. This adds a new layer.

2. Select the *ESRI Shapefile* format
3. Use the … button to browse to the `exercise_data/shapefile/` folder and save the file as `places.shp`.



4. Press *OK*

   In the *Layers* panel, the temporary *place* layer is replaced with the saved *places* shapefile layer and the temporary icon next to it removed.

5. Double-click the layer to open its *Layer Properties* ► *Source* tab and update the *Layer name* property to match the file name.

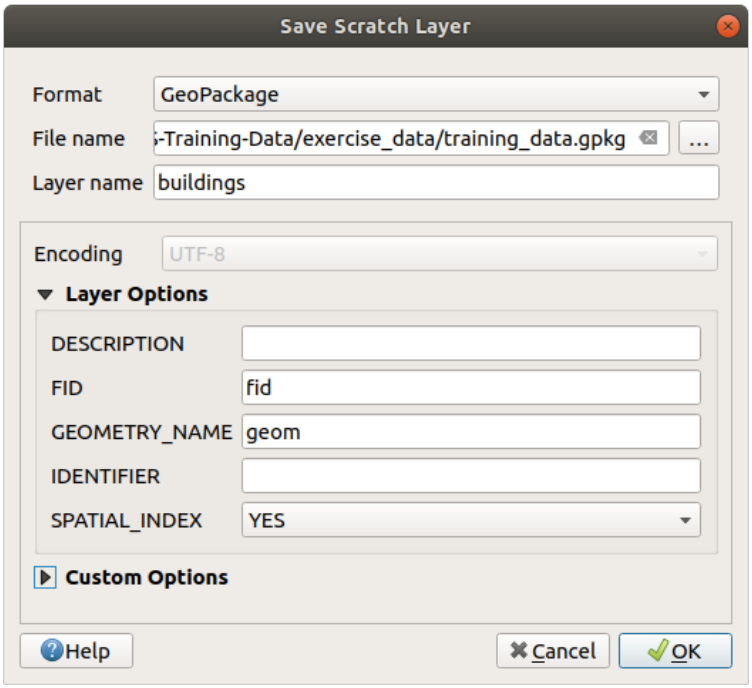
6. Repeat the process for other layers, renaming them as follows:

   - `natural_water` into `water`
   - `waterway_river` into `rivers`
   - `boundary_protected_area` into `protected_areas`

   Each resulting data set should be saved in the `exercise_data/shapefile/` directory.

The next step is to create a GeoPackage file from the *building* layer to use during the course:

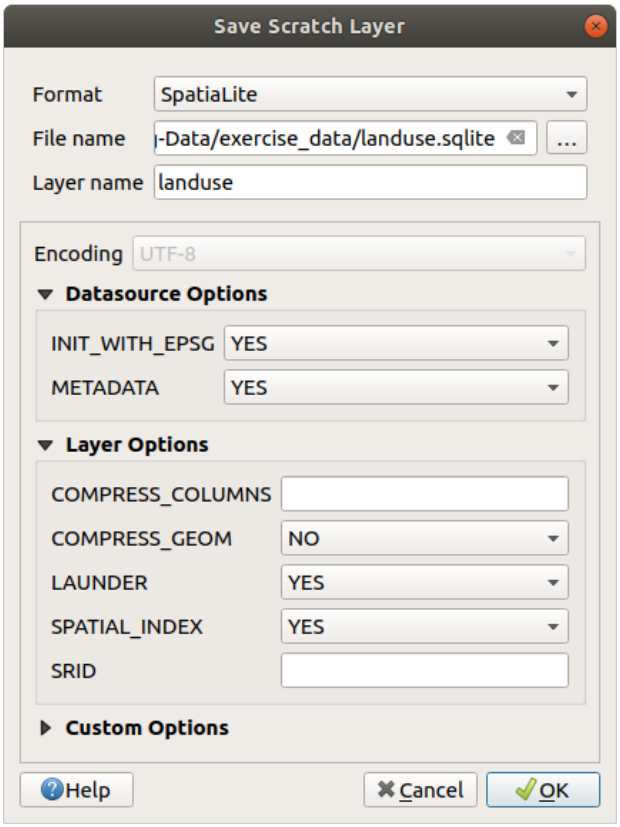
1. Click the icon next to the *building* layer
2. Select the *GeoPackage* format
3. Save the file as `training_data.gpkg` under the `exercise_data/` folder
4. By default, the *Layer name* is filled as the file name. Replace it with `buildings`.



5. Press *OK*
6. Rename the layer in its properties dialog
7. Repeat the process with the *highway* layer, saving it as `roads` in the same GeoPackage database.

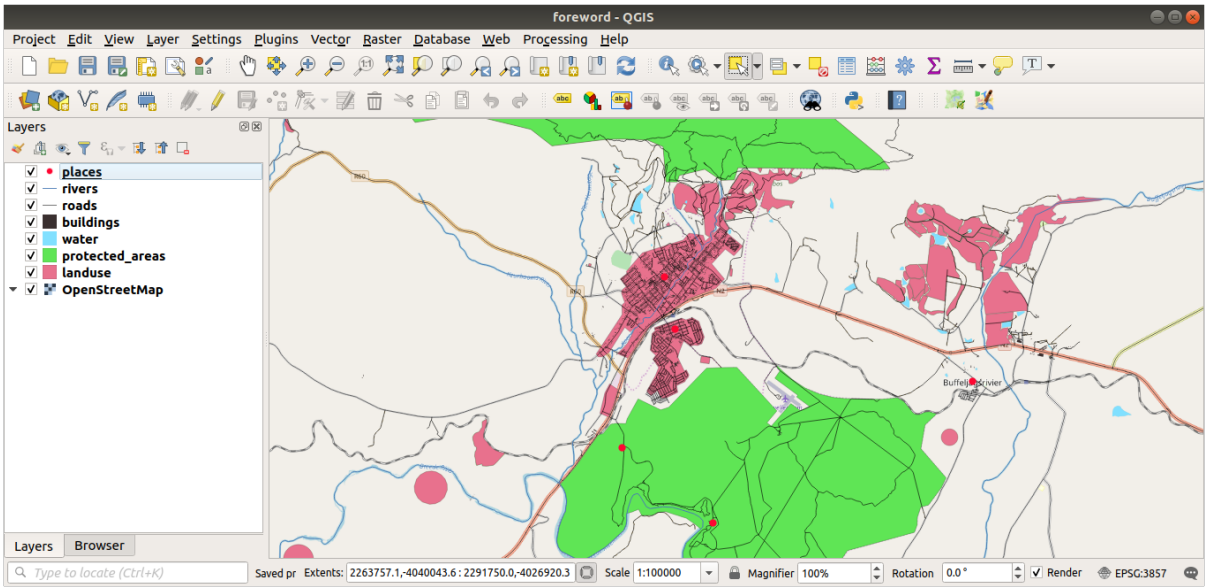
The last step is to save the remaining temporary file as a SpatiaLite file.

1. Click the icon next to the *landuse* layer
2. Select the *SpatiaLite* format
3. Save the file as `landuse.sqlite` under the `exercise_data/` folder. By default, the *Layer name* is filled as the file name. Do not change it.



4. Press *OK*

You should now have a map which looks something like this (the symbology will certainly be very different, because QGIS randomly assigns colors when layers are added to the map):



The important thing is that you have 7 vector layers matching those shown above and that all those layers have some data.

20.2  Try Yourself Создание файлов цифровой модели местности

For modules *Module: Модуль: Создание векторных данных* and *Модуль: Растры*, you'll also need raster images (SRTM DEM) which cover the region you have selected for your course.

The CGIAR-CGI provides some SRTM DEM you can download from https://srtm.csi.cgiar.org/srtmdata/.

You'll need images which cover the entire region you have chosen to use. To find the extent coordinates, in QGIS  zoom to the extent of the largest layer and pick the values in the  *Extents* box of the status bar. Keep the `GeoTiff` format. Once the form is filled, click on the *Click here to Begin Search* >> button and download the file(s).

Once you have downloaded the required file(s), they should be saved in the `exercise_data` directory, under `raster/SRTM` subfolders.

20.3  Try Yourself Create imagery tiff Files

In Module *Module: Модуль: Создание векторных данных*, *Идем дальше: Источник данных* lesson shows close-up images of three school sports fields which students are asked to digitize. You'll therefore need to reproduce these images using your new SRTM DEM tiff file(s). There is no obligation to use school sports fields: any three school land-use types can be used (e.g. different school buildings, playgrounds or car parks).

For reference, the image in the example data is:



20.4 Try Yourself Замените обозначения

Having created your localised dataset, the final step is to replace the tokens in the `substitutions.txt` file so that the appropriate names will appear in your localised version of the Training Manual.

Обозначения, которые Вам нужно заменить, таковы:

- `majorUrbanName`: this defaults to «Swellendam». Replace with the name of the major town in your region.
- `schoolAreaType1`: this defaults to «athletics field». Replace with the name of the largest school area type in your region.
- `largeLandUseArea`: this defaults to «Bontebok National Park». Replace with the name of a large landuse polygon in your region.
- `srtmFileName`: this defaults to `srtm_41_19.tif`. Replace this with the filename of your SRTM DEM file.
- `localCRS`: this defaults to `WGS 84 / UTM 34S`. You should replace this with the correct CRS for your region.