QGIS Training Manual

QGIS Project

03.03.2024

Inhaltsverzeichnis

1 Kurseinführung **1**

1.1 Vorwort . . . 1
1.1.1 Why QGIS? . . . 1
1.1.2 Hintergrund . . . 2
1.1.3 Lizenz . . . 2
1.1.4 Förderer . . . 3
1.1.5 Autoren . . . 3
1.1.6 Einzelne Mitwirkende . . . 3
1.1.7 Sponsoren . . . 3
1.1.8 Source files and Issue reports . . . 3
1.1.9 Aktuelle Version . . . 3
1.2 About the exercises . . . 4
1.2.1 How to use this tutorial . . . 4
1.2.2 Tiered course objectives . . . 4
1.2.3 Data . . . 5

2 Module: Erstellen und Erkunden einer einfachen Karte **7**

2.1 Lesson: An Overview of the Interface . . . 7
2.1.1 Try Yourself: The Basics . . . 8
2.1.2 Try Yourself 1 . . . 10
2.1.3 Try Yourself 2 . . . 10
2.1.4 What's Next? . . . 11
2.2 Lesson: Adding your first layers . . . 11
2.2.1 Follow Along: Prepare a map . . . 11
2.2.2 Try Yourself . . . 14
2.2.3 Follow Along: Loading vector data from a GeoPackage Database . . . 14
2.2.4 Follow Along: Loading vector data from a SpatiaLite Database with the Browser . . . 15
2.2.5 Try Yourself Load More Vector Data . . . 17
2.2.6 Follow Along: Reordering the Layers . . . 17
2.2.7 In Conclusion . . . 18
2.2.8 What's Next? . . . 18
2.3 Lesson: Navigating the Map Canvas . . . 18
2.3.1 Follow Along: Basic Navigation Tools . . . 18
2.3.2 In Conclusion . . . 22
2.4 Lesson: Symbologie . . . 22
2.4.1 Follow Along: Changing Colors . . . 23
2.4.2 Try Yourself . . . 24
2.4.3 Follow Along: Changing Symbol Structure . . . 24
2.4.4 Try Yourself . . . 25
2.4.5 Follow Along: Scale-Based Visibility . . . 27
2.4.6 Follow Along: Adding Symbol Layers . . . 28

2.4.7 Try Yourself . . . 31
2.4.8 Follow Along: Ordering Symbol Levels . . . 32
2.4.9 Try Yourself . . . 36
2.4.10 Try Yourself . . . 37
2.4.11 Follow Along: Symbol layer types . . . 39
2.4.12 Try Yourself . . . 46
2.4.13 Follow Along: Geometry generator symbology . . . 48
2.4.14 Follow Along: Creating a Custom SVG Fill . . . 52
2.4.15 In Conclusion . . . 58
2.4.16 Further Reading . . . 58
2.4.17 What's Next? . . . 58

3 Module: Vektordaten klassifizieren 59
3.1 Lesson: Vector Attribute Data . . . 59
3.1.1 Follow Along: Viewing Layer Attributes . . . 59
3.1.2 Try Yourself Exploring Vector Data Attributes . . . 63
3.1.3 In Conclusion . . . 64
3.1.4 What's Next? . . . 64
3.2 Lesson: Labels . . . 64
3.2.1 Follow Along: Using Labels . . . 64
3.2.2 Follow Along: Changing Label Options . . . 66
3.2.3 Follow Along: Using Labels Instead of Layer Symbology . . . 70
3.2.4 Try Yourself Customize the Labels . . . 73
3.2.5 Follow Along: Labeling Lines . . . 77
3.2.6 Follow Along: Data Defined Settings . . . 81
3.2.7 Try Yourself Using Data Defined Settings . . . 83
3.2.8 Further Possibilities With Labeling . . . 85
3.2.9 In Conclusion . . . 86
3.2.10 What's Next? . . . 86
3.3 Lesson: Klassifikation . . . 86
3.3.1 Follow Along: Classifying Nominal Data . . . 86
3.3.2 Try Yourself More Classification . . . 91
3.3.3 Follow Along: Ratio Classification . . . 91
3.3.4 Try Yourself Refine the Classification . . . 99
3.3.5 Follow Along: Rule-based Classification . . . 99
3.3.6 In Conclusion . . . 103
3.3.7 What's Next? . . . 103

4 Module: Layout der Karten 105
4.1 Lesson: Using Print Layout . . . 105
4.1.1 Follow Along: The Layout Manager . . . 105
4.1.2 Follow Along: Basic Map Composition . . . 107
4.1.3 Follow Along: Adding a Title . . . 109
4.1.4 Follow Along: Adding a Legend . . . 112
4.1.5 Follow Along: Customizing Legend Items . . . 113
4.1.6 Follow Along: Exporting Your Map . . . 115
4.1.7 In Conclusion . . . 117
4.2 Lesson: Creating a Dynamic Print Layout . . . 117
4.2.1 Follow Along: Creating the dynamic map canvas . . . 117
4.2.2 Follow Along: Creating the dynamic header . . . 118
4.2.3 Follow Along: Creating labels for the dynamic header . . . 119
4.2.4 Follow Along: Adding pictures to the dynamic header . . . 121
4.2.5 Follow Along: Creating the scalebar of the dynamic header . . . 122
4.2.6 What's Next? . . . 123
4.3 Zuordnung 1 . . . 123
4.3.1 In Conclusion . . . 123

5 Module: Vektordaten erzeugen 125
5.1 Lesson: Creating a New Vector Dataset . . . 125

5.1.1 Follow Along: The Layer Creation Dialog . . . 125
5.1.2 Follow Along: Datenquellen . . . 128
5.1.3 Try Yourself Digitizing Polygons . . . 141
5.1.4 Follow Along: Using Vertex Editor Table . . . 142
5.1.5 Try Yourself Digitizing Lines . . . 145
5.1.6 In Conclusion . . . 148
5.1.7 What's Next? . . . 148
5.2 Lesson: Funktionstopologie . . . 148
5.2.1 Follow Along: Snapping . . . 149
5.2.2 Follow Along: Correct Topological Features . . . 150
5.2.3 Follow Along: Tool: Simplify Feature . . . 153
5.2.4 Try Yourself Tool: Add Ring . . . 155
5.2.5 Try Yourself Tool: Add Part . . . 156
5.2.6 Follow Along: Tool: Reshape Features . . . 157
5.2.7 Try Yourself Tool: Split Features . . . 160
5.2.8 Try Yourself Tool: Merge Features . . . 161
5.2.9 In Conclusion . . . 162
5.2.10 What's Next? . . . 162
5.3 Lesson: Formulare . . . 162
5.3.1 Follow Along: Using QGIS' Form Design Functionality . . . 162
5.3.2 Try Yourself Using the Form to Edit Values . . . 163
5.3.3 Follow Along: Setting Form Field Types . . . 164
5.3.4 Try Yourself . . . 166
5.3.5 Try Yourself Creating Test Data . . . 168
5.3.6 Follow Along: Creating a New Form . . . 169
5.3.7 Follow Along: Associating the Form with Your Layer . . . 171
5.3.8 In Conclusion . . . 172
5.3.9 Further Reading . . . 172
5.3.10 What's Next? . . . 172
5.4 Lesson: Aktionen . . . 172
5.4.1 Follow Along: Add a Field for Images . . . 172
5.4.2 Follow Along: Creating an Action . . . 175
5.4.3 Follow Along: Searching the Internet . . . 179
5.4.4 Follow Along: Open a Webpage Directly in QGIS . . . 182
5.4.5 In Conclusion . . . 184
5.4.6 What's Next? . . . 184

6 Module: Vektoranalyse **185**
6.1 Lesson: Reprojecting and Transforming Data . . . 185
6.1.1 Follow Along: Projections . . . 186
6.1.2 Follow Along: „On the Fly" Reprojection . . . 186
6.1.3 Follow Along: Saving a Dataset to Another CRS . . . 187
6.1.4 Follow Along: Creating Your Own Projection . . . 190
6.1.5 In Conclusion . . . 192
6.1.6 Further Reading . . . 192
6.1.7 What's Next? . . . 192
6.2 Lesson: Vector Analysis . . . 192
6.2.1 The GIS Process . . . 193
6.2.2 The Problem . . . 193
6.2.3 The Data . . . 193
6.2.4 Follow Along: Start a Project and get the Data . . . 194
6.2.5 Try Yourself Convert Layers' CRS . . . 195
6.2.6 Follow Along: Analyzing the Problem: Distances From Schools and Roads . . . 197
6.2.7 Try Yourself Distance from schools . . . 204
6.2.8 Follow Along: Overlapping Areas . . . 207
6.2.9 Follow Along: Extract the Buildings . . . 210
6.2.10 Try Yourself Further Filter our Buildings . . . 211
6.2.11 Follow Along: Select Buildings of the Right Size . . . 214

6.2.12 Try Yourself . . . 215
6.2.13 In Conclusion . . . 215
6.2.14 What's Next? . . . 216
6.3 Lesson: Netzwerkanalyse . . . 216
6.3.1 Follow Along: The Tools and the Data . . . 216
6.3.2 Calculate the shortest path (point to point) . . . 217
6.3.3 Try Yourself Fastest path . . . 219
6.3.4 Follow Along: Advanced options . . . 220
6.3.5 Shortest path with speed limit . . . 223
6.3.6 Service area (from layer) . . . 226
6.3.7 In Conclusion . . . 228
6.3.8 What's Next? . . . 228
6.4 Lesson: Spatial Statistics . . . 228
6.4.1 Follow Along: Create a Test Dataset . . . 229
6.4.2 Follow Along: Basic Statistics . . . 233
6.4.3 Follow Along: Compute statistics on distances between points . . . 235
6.4.4 Follow Along: Nearest Neighbor Analysis (within layer) . . . 237
6.4.5 Follow Along: Mean Coordinates . . . 238
6.4.6 Follow Along: Image Histograms . . . 239
6.4.7 Follow Along: Spatial Interpolation . . . 242
6.4.8 Try Yourself Different interpolation methods . . . 242
6.4.9 In Conclusion . . . 243
6.4.10 What's Next? . . . 243

7 Module: Raster **245**
7.1 Lesson: Arbeiten mit Rasterdaten . . . 245
7.1.1 Follow Along: Rasterdaten laden . . . 245
7.1.2 Follow Along: Create a Virtual Raster . . . 246
7.1.3 Transforming Raster Data . . . 248
7.1.4 In Conclusion . . . 250
7.1.5 What's Next? . . . 251
7.2 Lesson: Changing Raster Symbology . . . 251
7.2.1 Try Yourself . . . 251
7.2.2 Follow Along: Changing Raster Layer Symbology . . . 252
7.2.3 Follow Along: Singleband gray . . . 252
7.2.4 Follow Along: Singleband pseudocolor . . . 254
7.2.5 Follow Along: Changing the transparency . . . 256
7.2.6 In Conclusion . . . 259
7.2.7 Referenz . . . 259
7.2.8 What's Next? . . . 259
7.3 Lesson: Terrain Analysis . . . 259
7.3.1 Follow Along: Calculating a Hillshade . . . 259
7.3.2 Follow Along: Using a Hillshade as an Overlay . . . 261
7.3.3 Follow Along: Finding the best areas . . . 262
7.3.4 Follow Along: Calculating the Slope . . . 262
7.3.5 Try Yourself Calculating the aspect . . . 263
7.3.6 Follow Along: Finding the north-facing aspect . . . 265
7.3.7 Try Yourself More criteria . . . 267
7.3.8 Follow Along: Combining Raster Analysis Results . . . 270
7.3.9 Follow Along: Simplifying the Raster . . . 272
7.3.10 Follow Along: Reclassifying the Raster . . . 274
7.3.11 Follow Along: Querying the raster . . . 278
7.3.12 In Conclusion . . . 281
7.3.13 What's Next? . . . 281

8 Module: Fertigstellung der Analyse **283**
8.1 Lesson: Raster to Vector Conversion . . . 283
8.1.1 Follow Along: The Raster to Vector Tool . . . 283

8.1.2 Try Yourself . . . 284
8.1.3 Follow Along: The Vector to Raster Tool . . . 285
8.1.4 In Conclusion . . . 287
8.1.5 What's Next? . . . 287
8.2 Lesson: Combining the Analyses . . . 287
8.2.1 Try Yourself . . . 287
8.2.2 Try Yourself Inspecting the Results . . . 288
8.2.3 Try Yourself Refining the Analysis . . . 288
8.2.4 In Conclusion . . . 293
8.2.5 What's Next? . . . 293
8.3 Zuordnung . . . 293
8.4 Lesson: Supplementary Exercise . . . 294
8.4.1 Problem Statement . . . 294
8.4.2 Solution Outline . . . 294
8.4.3 Follow Along: Setting up the Map . . . 294
8.4.4 Loading Data into the Map . . . 295
8.4.5 Changing the layer order . . . 296
8.4.6 Find the Correct Districts . . . 296
8.4.7 Clip the Rasters . . . 297
8.4.8 Changing the symbology of vector layers . . . 298
8.4.9 Changing the symbology of raster layers . . . 298
8.4.10 Clean up the map . . . 298
8.4.11 Create the hillshade . . . 299
8.4.12 Slope . . . 299
8.4.13 Try Yourself Aspect . . . 300
8.4.14 Reclassifying rasters . . . 300
8.4.15 Combining rasters . . . 300
8.4.16 Finding rural areas . . . 301
8.4.17 Creating a negative buffer . . . 301
8.4.18 Vectorizing the raster . . . 302
8.4.19 Fixing geometry . . . 302
8.4.20 Determining the Intersection of vectors . . . 303
8.4.21 Calculating the area for each polygon . . . 303
8.4.22 Selecting areas of a given size . . . 304
8.4.23 Digitize the University of Cape Town . . . 304
8.4.24 Find the locations that are closest to the University of Cape Town . . . 305

9 Module: Erweiterungen **307**
9.1 Lesson: Installing and Managing Plugins . . . 307
9.1.1 Follow Along: Managing Plugins . . . 307
9.1.2 Follow Along: Installing New Plugins . . . 308
9.1.3 Follow Along: Configuring Additional Plugin Repositories . . . 310
9.1.4 In Conclusion . . . 312
9.1.5 What's Next? . . . 312
9.2 Lesson: Nützliche QGIS Erweiterungen . . . 312
9.2.1 Follow Along: The QuickMapServices Plugin . . . 312
9.2.2 Follow Along: The QuickOSM Plugin . . . 314
9.2.3 Follow Along: The QuickOSM Query engine . . . 315
9.2.4 Follow Along: The DataPlotly Plugin . . . 316
9.2.5 In Conclusion . . . 320
9.2.6 What's Next? . . . 320

10 Module: Datenquellen aus dem Internet **321**
10.1 Lesson: Web Mapping Services . . . 321
10.1.1 Follow Along: Loading a WMS Layer . . . 321
10.1.2 Try Yourself . . . 331
10.1.3 Try Yourself . . . 332
10.1.4 Try Yourself . . . 333

10.1.5 In Conclusion . . . 334
10.1.6 Further Reading . . . 334
10.1.7 What's Next? . . . 335
10.2 Lesson: Web Feature Services . . . 335
10.2.1 Follow Along: Loading a WFS Layer . . . 335
10.2.2 Follow Along: Querying a WFS Layer . . . 339
10.2.3 In Conclusion . . . 341
10.2.4 What's Next? . . . 341

11 Module: QGIS Server **343**
11.1 Lesson: Install QGIS Server . . . 343
11.1.1 Follow Along: Install from packages . . . 343
11.1.2 Follow Along: QGIS Server Executable . . . 344
11.1.3 HTTP Server Configuration . . . 344
11.1.4 Follow Along: Create another virtual host . . . 344
11.1.5 In Conclusion . . . 345
11.1.6 What's Next? . . . 345
11.2 Lesson: Serving WMS . . . 345
11.2.1 Logging . . . 348
11.2.2 GetMap requests . . . 349
11.2.3 Try Yourself Change the Image and Layers parameters . . . 350
11.2.4 Follow Along: Use Filter, Opacities and Styles parameters . . . 351
11.2.5 Follow Along: Use Redlining . . . 352
11.2.6 GetPrint requests . . . 353
11.2.7 In Conclusion . . . 355
11.2.8 What's Next? . . . 355

12 Module: GRASS **357**
12.1 Lesson: GRASS-Einrichtung . . . 357
12.1.1 Follow Along: Start einer neuen GRASS Sitzung . . . 357
12.1.2 Follow Along: Starten eines neuen GRASS Projektes . . . 360
12.1.3 Follow Along: Vektordaten in GRASS laden . . . 367
12.1.4 Follow Along: Rasterdaten in GRASS laden . . . 371
12.1.5 Try Yourself Layer zum Mapset hinzufügen . . . 373
12.1.6 Öffnen eines vorhandenen Mapsets . . . 373
12.1.7 In Conclusion . . . 376
12.1.8 What's Next? . . . 376
12.2 Lesson: GRASS-Werkzeuge . . . 376
12.2.1 Follow Along: Erstellung einer Karte der Expostition . . . 376
12.2.2 Follow Along: Eine einfache Statistik über den Rasterlayer erstellen . . . 378
12.2.3 Follow Along: Das Reklassifizierungswerkzeug . . . 380
12.2.4 Try Yourself Reklassifizieren mit eigenen Regeln . . . 382
12.2.5 Follow Along: Das Mapcalc Werkzeug . . . 383
12.2.6 In Conclusion . . . 385

13 Module: Bewertung **387**
13.1 Erstellen Sie eine Basiskarte . . . 387
13.1.1 Fügen Sie den Punktlayer hinzu . . . 387
13.1.2 Fügen Sie einen Linien Layer hinzu . . . 388
13.1.3 Fügen Sie den Polygonlayer hinzu . . . 389
13.1.4 Erstellen Sie den Rasterhintergrund . . . 389
13.1.5 Stellen Sie die Basiskarte fertig . . . 389
13.2 Analysieren Sie die Daten . . . 390
13.2.1 / . . . 390
13.3 Endgültige Karte . . . 390

14 Module: Anwendung in der Forstwirtschaft **391**
14.1 Lesson: Forestry Module Presentation . . . 391
14.1.1 Forestry Sample Data . . . 392

14.2 Lesson: Georeferencing a Map . . . 392
14.2.1 Scan the map . . . 392
14.2.2 Follow Along: Georeferencing the scanned map . . . 392
14.2.3 In Conclusion . . . 398
14.2.4 What's Next? . . . 398
14.3 Lesson: Digitizing Forest Stands . . . 398
14.3.1 Follow Along: Extracting the Forest Stands Borders . . . 398
14.3.2 Try Yourself Georeference the Green Pixels Image . . . 401
14.3.3 Follow Along: Creating Supporting Points for Digitizing . . . 402
14.3.4 Follow Along: Digitize the Forest Stands . . . 404
14.3.5 Try Yourself Finish Digitizing the Forest Stands . . . 408
14.3.6 Follow Along: Joining the Forest Stand Data . . . 409
14.3.7 Try Yourself Adding Area and Perimeter . . . 412
14.3.8 In Conclusion . . . 413
14.3.9 What's Next? . . . 413
14.4 Lesson: Updating Forest Stands . . . 413
14.4.1 Comparing the Old Forest Stands to Current Aerial Photographs . . . 413
14.4.2 Interpreting the CIR Image . . . 414
14.4.3 Try Yourself Digitizing Forest Stands from CIR Imagery . . . 416
14.4.4 Follow Along: Updating Forest Stands with Conservation Information . . . 418
14.4.5 Try Yourself Updating Forest Stands with Distance to the Stream . . . 422
14.4.6 In Conclusion . . . 422
14.4.7 What's Next? . . . 423
14.5 Lesson: Systematic Sampling Design . . . 423
14.5.1 Inventorying the Forest . . . 423
14.5.2 Follow Along: Implementing a Systematic Sampling Plot Design . . . 423
14.5.3 Follow Along: Exporting Sample Plots as GPX format . . . 427
14.5.4 In Conclusion . . . 428
14.5.5 What's Next? . . . 428
14.6 Lesson: Creating Detailed Maps with the Atlas Tool . . . 428
14.6.1 Follow Along: Preparing the Print Layout . . . 428
14.6.2 Follow Along: Adding Background Map . . . 430
14.6.3 Try Yourself Changing the Symbology of the Layers . . . 430
14.6.4 Try Yourself Create a Basic Map Template . . . 432
14.6.5 Follow Along: Adding More Elements to the Print Layout . . . 433
14.6.6 Follow Along: Creating an Atlas Coverage . . . 435
14.6.7 Follow Along: Setting Up the Atlas Tool . . . 437
14.6.8 Follow Along: Editing the Coverage Layer . . . 439
14.6.9 Follow Along: Printing the Maps . . . 442
14.6.10 In Conclusion . . . 442
14.6.11 What's Next? . . . 443
14.7 Lesson: Calculating the Forest Parameters . . . 443
14.7.1 Follow Along: Adding the Inventory Results . . . 443
14.7.2 Follow Along: Whole Forest Parameters Estimation . . . 444
14.7.3 Follow Along: Estimating Stand Parameters . . . 444
14.7.4 In Conclusion . . . 448
14.7.5 What's Next? . . . 449
14.8 Lesson: DEM zu LiDAR-Daten . . . 449
14.8.1 Follow Along: Installing Lastools . . . 449
14.8.2 Follow Along: Calculating a DEM with LAStools . . . 451
14.8.3 Follow Along: Creating a Terrain Hillshade . . . 456
14.8.4 In Conclusion . . . 458
14.8.5 What's Next? . . . 459
14.9 Lesson: Kartendarstellung . . . 459
14.9.1 Follow Along: Preparing the Map Data . . . 459
14.9.2 Try Yourself Try Different Blending Modes . . . 461
14.9.3 Try Yourself Using a Layout Template to Create the Map result . . . 462
14.9.4 In Conclusion . . . 465

15 Module: Datenbank-Konzepte mit PostgreSQL **467**
15.1 Lesson: Einführung in Datenbanken . . . 467
15.1.1 Was ist eine Datenbank? . . . 467
15.1.2 Tabellen . . . 467
15.1.3 Spalten / Felder . . . 468
15.1.4 Datensätze . . . 468
15.1.5 Datentypen . . . 468
15.1.6 Modellierung einer Adressendatenbank . . . 469
15.1.7 Datenbanktheorie . . . 469
15.1.8 Normalisierung . . . 469
15.1.9 Try Yourself . . . 470
15.1.10 Indexe . . . 471
15.1.11 Sequenzen . . . 471
15.1.12 Entity Relationship Diagramme . . . 471
15.1.13 Einschränkungen, Primärschlüssel und Fremdschlüssel . . . 473
15.1.14 Transaktionen . . . 474
15.1.15 In Conclusion . . . 474
15.1.16 What's Next? . . . 474
15.2 Lesson: Implementierung des Datenmodells . . . 474
15.2.1 Installation von PostgreSQL . . . 474
15.2.2 Hilfe . . . 475
15.2.3 Erstellen eines Datenbankbenutzers . . . 475
15.2.4 Überprüfen Sie das neue Konto: . . . 475
15.2.5 Eine Datenbank erstellen . . . 476
15.2.6 Start einer Datenbank shell Sitzung . . . 476
15.2.7 Erstellen von Tabellen in SQL . . . 477
15.2.8 Erstellen von Schlüsseln in SQL . . . 478
15.2.9 Indexe mit SQL erstellen . . . 479
15.2.10 Löschen von Tabellen mit SQL . . . 480
15.2.11 Ein Wort zu pgAdmin III . . . 480
15.2.12 In Conclusion . . . 480
15.2.13 What's Next? . . . 480
15.3 Lesson: Daten in das Modell einfügen . . . 481
15.3.1 Die Einfügeanweisung . . . 481
15.3.2 Aneinanderreihen von Einfügeaktionen aufgrund von Beschränkungen . . . 482
15.3.3 Try Yourself . . . 482
15.3.4 Auswahl von Daten . . . 482
15.3.5 Aktualisierung von Daten . . . 483
15.3.6 Löschen von Daten . . . 483
15.3.7 Try Yourself . . . 483
15.3.8 In Conclusion . . . 483
15.3.9 What's Next? . . . 484
15.4 Lesson: Abfragen . . . 484
15.4.1 Ordnen der Ergebnisse . . . 484
15.4.2 Filtern . . . 485
15.4.3 Verbindungen . . . 486
15.4.4 Unterauswahl . . . 486
15.4.5 Abfragen zur Aggregierung . . . 487
15.4.6 In Conclusion . . . 488
15.4.7 What's Next? . . . 488
15.5 Lesson: Ansichten . . . 488
15.5.1 Erstellen einer Sicht . . . 488
15.5.2 Veränderung einer Sicht . . . 489
15.5.3 Verwerfen einer Sicht . . . 489
15.5.4 In Conclusion . . . 489
15.5.5 What's Next? . . . 490
15.6 Lesson: Regeln . . . 490
15.6.1 Erstellen einer Regel zum protokollieren . . . 490

15.6.2 In Conclusion . . . 491
15.6.3 What's Next? . . . 491

16 Module: Spatial Database Concepts with PostGIS 493
16.1 Lesson: PostGIS-Einrichtung . . . 493
16.1.1 Installation unter Ubuntu . . . 494
16.1.2 Installation unter Windows . . . 494
16.1.3 Installation auf anderen Plattformen . . . 494
16.1.4 Konfiguration von Datenbanken zur Nutzung von PostGIS . . . 494
16.1.5 Die installierten PostGIS Funktionen . . . 495
16.1.6 Räumliche Bezugssysteme . . . 496
16.1.7 In Conclusion . . . 496
16.1.8 What's Next? . . . 497
16.2 Simple-Feature-Modell . . . 497
16.2.1 Was ist OGC . . . 497
16.2.2 Was ist das SFS Modell . . . 497
16.2.3 Hinzufügen eines Geometriefeldes zu einer Tabelle . . . 498
16.2.4 Hinzufügen einer Einschränkung basierend auf dem Geometrietyp . . . 498
16.2.5 Try Yourself . . . 498
16.2.6 Füllen der geometry_columns Tabelle . . . 498
16.2.7 Hinzufügen eines Geomtriedatensatzes zur Tabelle mit Hilfe von SQL . . . 499
16.2.8 In Conclusion . . . 502
16.2.9 What's Next? . . . 502
16.3 Lesson: Importieren und Exportieren . . . 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: Räumliche Abfragen . . . 504
16.4.1 Räumliche Operationen . . . 504
16.4.2 Räumliche Indexe . . . 504
16.4.3 Try Yourself . . . 505
16.4.4 Demonstration der räumlichen Funktionen von PostGIS . . . 505
16.4.5 In Conclusion . . . 512
16.4.6 What's Next? . . . 512
16.5 Lesson: Geometrieaufbau . . . 512
16.5.1 Erstellen von Linien . . . 513
16.5.2 Try Yourself . . . 513
16.5.3 Erstellung von Polygonen . . . 513
16.5.4 Übernung: Anbindung Städte an Personen . . . 514
16.5.5 Unser Schema . . . 515
16.5.6 Try Yourself . . . 516
16.5.7 Zugriff auf Unter-Objekte . . . 516
16.5.8 Datenverarbeitung . . . 516
16.5.9 Ausschneiden . . . 516
16.5.10 Geometrien aus anderen Geometrien erstellen . . . 518
16.5.11 Geometriebereinigung . . . 519
16.5.12 Unterschiede zwischen Tabellen . . . 519
16.5.13 Tablespaces . . . 520
16.5.14 In Conclusion . . . 520

17 Der QGIS-Verarbeitungsleitfaden 521
17.1 Einleitung . . . 521
17.2 Eine wichtige Warnung zu Beginn . . . 522
17.3 Einrichtung der Verarbeitungsumgebung . . . 523
17.4 Ausführung unseres ersten Algorithmus. Der Werkzeugkasten . . . 526

17.5 Mehr Algorithmen und Datentypen . . . 529
17.6 KBS - reprojizieren . . . 537
17.7 Auswahl . . . 540
17.8 Ausführung eines externen Algorithmus . . . 542
17.9 Das Prozessierungsprotokoll . . . 548
17.9.1 Fortgeschritten . . . 549
17.10 Der Rasterrechner. Nullwerte . . . 550
17.11 Vektorberechner . . . 554
17.12 Definierung von Ausschnitten . . . 558
17.13 HTML-Ausgaben . . . 562
17.14 Erstes Analysebeispiel . . . 564
17.15 Zuschneiden und Verschmelzen von Rasterlayern . . . 573
17.16 Hydrologische Analyse . . . 582
17.17 Starting with the model designer . . . 594
17.18 Mehr komplexe Modelle . . . 605
17.19 Nummerische Berechnungen im Modellierer . . . 610
17.20 Ein Model in einem Modell . . . 615
17.21 Erstellung eines Modells mit modellinternen Werkzeugen . . . 616
17.22 Interpolation . . . 622
17.23 Noch mehr Interpolation . . . 631
17.24 Die iterative Ausführung von Algorithmen . . . 638
17.25 Noch mehr iterative Ausführung von Algortihmen . . . 642
17.26 Die Schnittstelle zur Stapelverarbeitung . . . 644
17.27 Modelle innerhalb der Stapelverarbeitung . . . 647
17.28 Aktionsskripte vor oder nach Ausführung . . . 648
17.29 Andere Programme . . . 649
17.29.1 GRASS . . . 649
17.29.2 R . . . 649
17.29.3 Andere . . . 650
17.29.4 Comparison among backends . . . 650
17.30 Interpolation und Kontourlinienerzeugung . . . 651
17.30.1 Interpolation . . . 651
17.30.2 Contour . . . 651
17.31 Vektorvereinfachung und Glättung . . . 651
17.32 Planung eines Solarparks . . . 652
17.33 Nutzung von R-Skripten in der Prozessierung . . . 652
17.33.1 Hinzufügen von Skripten . . . 653
17.33.2 Erstellung von Plots . . . 654
17.33.3 Erstellen eines Vektorlayers . . . 657
17.33.4 Textausgabe und grafische Ausgabe mit der R - Syntax . . . 660
17.34 Predicting landslides . . . 660

18 Module: Räumliche Datenbanken in QGIS nutzen **661**
18.1 Lesson: Arbeit mit Datenbanken im QGIS Browser . . . 661
18.1.1 Follow Along: Datenbanktabellen mit Hilfe des Browsers zu QGIS hinzufügen . . . 661
18.1.2 Follow Along: Hinzufügen einer gefilterten Liste von Datensätzen als Layer . . . 663
18.1.3 In Conclusion . . . 664
18.1.4 What's Next? . . . 665
18.2 Lesson: Verwendung der DB-Verwaltung zur Arbeit mit räumlichen Datenbanken in QGIS . . . 665
18.2.1 Follow Along: Managen von PostGIS Datenbanken mit der DB-Verwaltung . . . 665
18.2.2 Follow Along: Erstellen einer neuen Tabelle . . . 670
18.2.3 Follow Along: Einfache Datenbankadministration . . . 671
18.2.4 Follow Along: Executing SQL Queries with DB Manager . . . 672
18.2.5 Importing Data into a Database with DB Manager . . . 673
18.2.6 Exporting Data from a Database with DB Manager . . . 675
18.2.7 In Conclusion . . . 677
18.2.8 What's Next? . . . 677
18.3 Lesson: Working with SpatiaLite databases in QGIS . . . 677

18.3.1 Follow Along: Creating a SpatiaLite database with the Browser . . . 677
18.3.2 In Conclusion . . . 679

19 Anhang: Mitwirkende an diesem Handbuch 681
19.1 Ressourcen herunterladen . . . 681
19.2 Manual Format . . . 681
19.3 Hinzufügen eines Moduls . . . 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 Vorbereitung der Übungsdaten 689
20.1 Try Yourself Create OSM based vector files . . . 689
20.2 Try Yourself Create SRTM DEM tiff files . . . 696
20.3 Try Yourself Create imagery tiff files . . . 696
20.4 Try Yourself Replace tokens . . . 697

KAPITEL 1

Kurseinführung

1.1 Vorwort

Welcome to our course! We will be showing you how to use QGIS easily and efficiently. If you are new to GIS, we will tell you what you need to get started. If you are an experienced user, you will see how QGIS fulfills all the functions you expect from a GIS program, and more!

1.1.1 Why QGIS?

As information becomes increasingly spatially aware, there is no shortage of tools able to fulfill some or all commonly used GIS functions. Why should anyone be using QGIS over some other GIS software package?

Here are only some of the reasons:

- *It's free, as in lunch.* Installing and using the QGIS program costs you a grand total of zero money. No initial fee, no recurring fee, nothing.
- *It's free, as in liberty.* If you need extra functionality in QGIS, you can do more than just hope it will be included in the next release. You can sponsor the development of a feature, or add it yourself if you are familiar with programming.
- *It's constantly developing.* Because anyone can add new features and improve on existing ones, QGIS never stagnates. The development of a new tool can happen as quickly as you need it to.
- *Extensive help and documentation is available.* If you're stuck with anything, you can turn to the extensive documentation, your fellow QGIS users, or even the developers.
- *Cross-platform.* QGIS can be installed on macOS, Windows and Linux.

Now that you know why you want to use QGIS, these exercises will make you know how.

1.1.2 Hintergrund

In 2008 we launched the Gentle Introduction to GIS, a completely free, open content resource for people who want to learn about GIS without being overloaded with jargon and new terminology. It was sponsored by the South African government and has been a phenomenal success, with people all over the world writing to us to tell us how they are using the materials to run University Training Courses, teach themselves GIS and so on. The Gentle Introduction is not a software tutorial, but rather aims to be a generic text (although we used QGIS in all examples) for someone learning about GIS. There is also the QGIS manual which provides a detailed functional overview of the QGIS application. However, it is not structured as a tutorial, but rather as a reference guide. At Linfiniti Consulting CC. we frequently run training courses and have realised that a third resource is needed - one that leads the reader sequentially through learning the key aspects of QGIS in a trainer-trainee format - which prompted us to produce this work.

Dieses Trainingshandbuch ist dafür vorgesehen alle Materialien bereitzustellen, die für einen Kurs über 5 Tage in QGIS, PostgreSQL und PostQIS nötig sind. Dieser Kurs beinhaltet Inhalte für Anfänger, bereits vertraute und fortgeschrittene Benutzer und besitzt ebenso viele Übungen mit Antworten im Text.

1.1.3 Lizenz

The Free Quantum GIS Training Manual by Linfiniti Consulting CC. is based on an earlier version from Linfiniti and is licensed under a Creative Commons Attribution 4.0 International. Permissions beyond the scope of this license may be available at below.

Wir haben das QGIS Trainingshandbuch unter einer freien Lizenz veröffentlicht, was Ihnen die Freiheit bietet die Inhalte zu kopieren, modifizieren und zu verteilen. In einfachen Worten gesprochen gelten folgende Nutzungsbedingungen:

- Es ist nicht erlaubt die Inhalte als eigene Arbeit darzustellen, urheberrechtlich Texte oder Namensnennungen zu entfernen.
- Sie sind nicht berechtigt diese Arbeit und mehr eingeschränkten Berechtigungen zu verteilen, als wie sie Ihnen zur Verfügung gestellt wurde.
- Falls Sie weitere Inhalte zu dieser Arbeit hinzufügen (zumindest ein komplettes Modul), dann können Sie Ihren Namen am Ende der Autorenliste dieses Dokumentes einfügen (Dieser wird auf der Startseite angezeigt)
- Falls Sie kleinere Änderungen beifügen, dann tragen Sie Ihren Namen bei *Einzelne Mitwirkende* ein.
- Falls Sie dieses Dokument komplett Übersetzt haben, dann können Sie ihren Namen in der Autorenliste in Form von „Übersetzt von Joe Bloggs“ eintragen.
- Falls Sie ein Modul oder eine Lehreinheit gesponsert haben, dann können Sie den Autor bitten eine Anmerkung an den Beginn jeder beigetragenen Lehreinheit beizufügen, z.B.:

Bemerkung: Diese Lektion wurde von Megacorp gesponsert.

- Falls Sie unsicher sind, wie Sie mit dieser Lizenz arbeiten können, dann kontaktieren Sie uns bitte unter: office@linfiniti.com und wir beraten Sie, ob Ihr vorhaben in Ordnung ist.
- If you publish this work under a self publishing site such as https://www.lulu.com we request that you donate the profits to the QGIS project.
- Sie dürfen diese Arbeit nicht vermarkten, außer mit der Erlaubnis der Autoren. Hierbei ist gemeint, dass die Arbeit nicht zum Profit verkauft werden darf, wie das erstellen eines kommerziell abgeleiteten Werkes (z.B. Inhalt an Magazine verkaufen). Die Ausnahme besteht darin, den gesamten Gewinn an das QGIS-Projekt weiter zu leiten. Sie dürfen (und wir ermutigen Sie dazu) diese Arbeit als Lehrbuch zu verwenden, wen Sie Kurse anbieten, selbst wenn diese kommerziell sind. Anders gesagt, selbstverständlich dürfen Sie einen

kostenpflichtigen Kurs anbieten und dieses Lehrbuch verwenden, aber Sie dürfen das Lehrbuch selber nicht verkaufen - solche Gelder sollten an das QGIS-Projekt gehen.

1.1.4 Förderer

Diese Arbeit umfasst nicht die komplette Abhandlung der Dinge die mit QGIS getan werden können und wir ermutigen euch Inhalte beizutragen, welche diese Lücken schließen. Linfiniti Consulting CC. kann auf kommerzieller Basis für Sie weitere Inhalte erstellten, diese werden jedoch unter der gleichen Lizenz veröffentlich.

1.1.5 Autoren

- 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.
- Horst Düster (horst.duester@kappasys.ch) - Horst war Co-Autor in den Bereichen PostSQL und PostGIS
- Marcelle Sutton - Marcelle provided proof-reading and editorial advice during the creation of this work.

1.1.6 Einzelne Mitwirkende

Hier könnte Ihr Name stehen!

1.1.7 Sponsoren

- Cape Peninsula University of Technology

1.1.8 Source files and Issue reports

The source of this document is available at GitHub QGIS Documentation repository. Consult GitHub.com for instructions on how to use the git version control system.

Despite our efforts, you could find some errors or miss some information while following this training. Please report them at https://github.com/qgis/QGIS-Documentation/issues.

1.1.9 Aktuelle Version

You can always obtain the latest version of this document by visiting the online version which is part of the QGIS documentation website (https://docs.qgis.org).

Bemerkung: The documentation website contains links to both online and PDF versions of the Training manual and other parts of the QGIS documentation.

1.2 About the exercises

Now that you know why you want to use QGIS, we can show you how.

Warnung: This course 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!

1.2.1 How to use this tutorial

Any text that *looks like this* refers to something that you can see in the QGIS user interface.

Text that *looks* ► *like* ► *this* directs you through menus.

`This kind of text` refers to something you can type, such as a command.

`This/kind/of.text` refers to a path or filename.

`This+That` refers to a keyboard shortcut comprised of two buttons.

1.2.2 Tiered course objectives

This course caters to different user experience levels. Depending on which category you consider yourself to be in, you can expect a different set of course outcomes. Each category contains information that is essential for the next one, so it's important to do all exercises that are at or below your level of experience.

Basic

In this category, the course assumes that you have little or no prior experience with theoretical GIS knowledge or the operation of GIS software.

Limited theoretical background will be provided to explain the purpose of an action you will be performing in the program, but the emphasis is on learning by doing.

When you complete the course, you will have a better concept of the possibilities of GIS, and how to harness their power via QGIS.

Intermediate

In this category, it is assumed that you have working knowledge and experience of the everyday uses of GIS software.

Building on the instructions for the beginner level will provide you with familiar ground, as well as to make you aware of the cases where QGIS does things slightly differently from other software you may be used to. You will also learn how to use analysis functions in QGIS.

When you complete the course, you should be comfortable with using QGIS for all of the functions you usually need for everyday use.

In this category, the assumption is that you are experienced with GIS software, have knowledge of and experience with spatial databases, using data on a remote server, perhaps writing scripts for analysis purposes, etc.

Building on the instructions for the other two levels will familiarize you with the approach that the QGIS interface follows, and will ensure that you know how to access the basic functions that you need. You will also be shown how to make use of the QGIS plugin system, database access, and so on.

When you complete the course, you should be well-acquainted with the everyday operation of QGIS, as well as its more advanced functions.

1.2.3 Data

The sample data that accompanies this resource is freely available and comes from the following sources:

- Streets and Places datasets from OpenStreetMap (https://www.openstreetmap.org/)
- Property boundaries (urban and rural), water bodies from NGI (https://ngi.dalrrd.gov.za/)
- SRTM DEM from the CGIAR-CGI (http://srtm.csi.cgiar.org/)

Download the prepared dataset from the Training data repository and unzip the file. All the necessary data are provided in the `exercise_data` folder.

If you are an instructor, and would like to use more relevant data, you will find instructions for creating local data in the *Vorbereitung der Übungsdaten* appendix.

KAPITEL 2

Module: Erstellen und Erkunden einer einfachen Karte

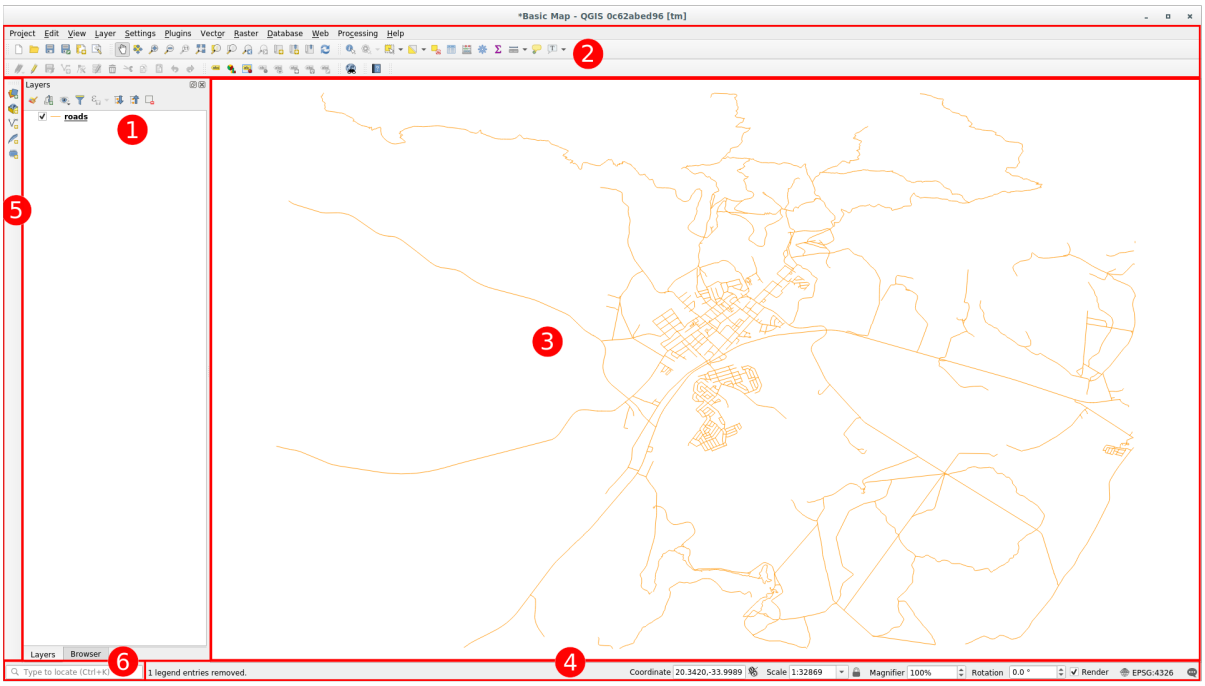
In diesem Modul erstellen Sie eine grundlegende Karte, die später als Grundlage für weitere Demonstrationen der QGIS-Funktionalität verwendet wird.

2.1 Lesson: An Overview of the Interface

We will explore the QGIS user interface so that you are familiar with the menus, toolbars, map canvas and layers list that form the basic structure of the interface.

The goal for this lesson: To understand the basics of the QGIS user interface.

2.1.1 Try Yourself: The Basics



The elements identified in the figure above are:

1. Layers List / Browser Panel
2. Werkzeugleiste
3. Map canvas
4. Status bar
5. Side Toolbar
6. Suchleiste

The Layers List

In the Layers list, you can see a list, at any time, of all the layers available to you.

Expanding collapsed items (by clicking the arrow or plus symbol beside them) will provide you with more information on the layer's current appearance.

Hovering over the layer will give you some basic information: layer name, type of geometry, coordinate reference system and the complete path of the location on your device.

Right-clicking on a layer will give you a menu with lots of extra options. You will be using some of them before long, so take a look around!

Bemerkung: A vector layer is a dataset, usually of a specific kind of object, such as roads, trees, etc. A vector layer can consist of either points, lines or polygons.

The Browser Panel

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

If you have saved a project, the Browser Panel will also give you quick access to all the layers stored in the same path of the project file under in the *Project Home* item.

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

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

Toolbars

Your most often used sets of tools can be turned into toolbars for basic access. For example, the *Project* toolbar allows you to save, load, print, and start a new project. You can easily customize the interface to see only the tools you use most often, adding or removing toolbars as necessary via the *View* ► *Toolbars* menu.

Even if they are not visible in a toolbar, all of your tools will remain accessible via the menus. For example, if you remove the *Project* toolbar (which contains the *Save* button), you can still save your map by clicking on the *Project* menu and then clicking on *Save*.

The Map Canvas

This is where the map itself is displayed and where layers are loaded. In the map canvas you can interact with the visible layers: zoom in/out, move the map, select features and many other operations that we will deeply see in the next sections.

The Status Bar

Shows you information about the current map. Also allows you to adjust the map scale, the map rotation and see the mouse cursor's coordinates on the map.

The Side Toolbar

By default the Side toolbar contains the buttons to load the layer and all the buttons to create a new layer. But remember that you can move all the toolbars wherever it is more comfortable for you.

The Locator Bar

Within this bar you can access to almost all the objects of QGIS: layers, layer features, algorithms, spatial bookmarks, etc. Check all the different options in the locator_options section of the QGIS User Manual.

Tipp: With the shortcut `Ctrl+K` you can easily access the bar.

2.1.2 Try Yourself 1

Try to identify the four elements listed above on your own screen, without referring to the diagram above. See if you can identify their names and functions. You will become more familiar with these elements as you use them in the coming days.

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 Try Yourself 2

Try to find each of these tools on your screen. What is their purpose?

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

Bemerkung: If any of these tools is not visible on the screen, try enabling some toolbars that are currently hidden. Also keep in mind that if there isn't enough space on the screen, a toolbar may be shortened by hiding some of its tools. You can see the hidden tools by clicking on the double right arrow button in any such collapsed toolbar. You can see a tooltip with the name of any tool by holding your mouse over the tool for a while.

Answer

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

2.1.4 What's Next?

Now that you are familiar with the basics of the QGIS interface, in the next lesson we will see how to load some common data types.

2.2 Lesson: Adding your first layers

We will start the application, and create a basic map to use for examples and exercises.

The goal for this lesson: To get started with an example map.

Bemerkung: Before starting this exercise, QGIS must be installed on your computer. Also, you should have downloaded the *sample data* to use.

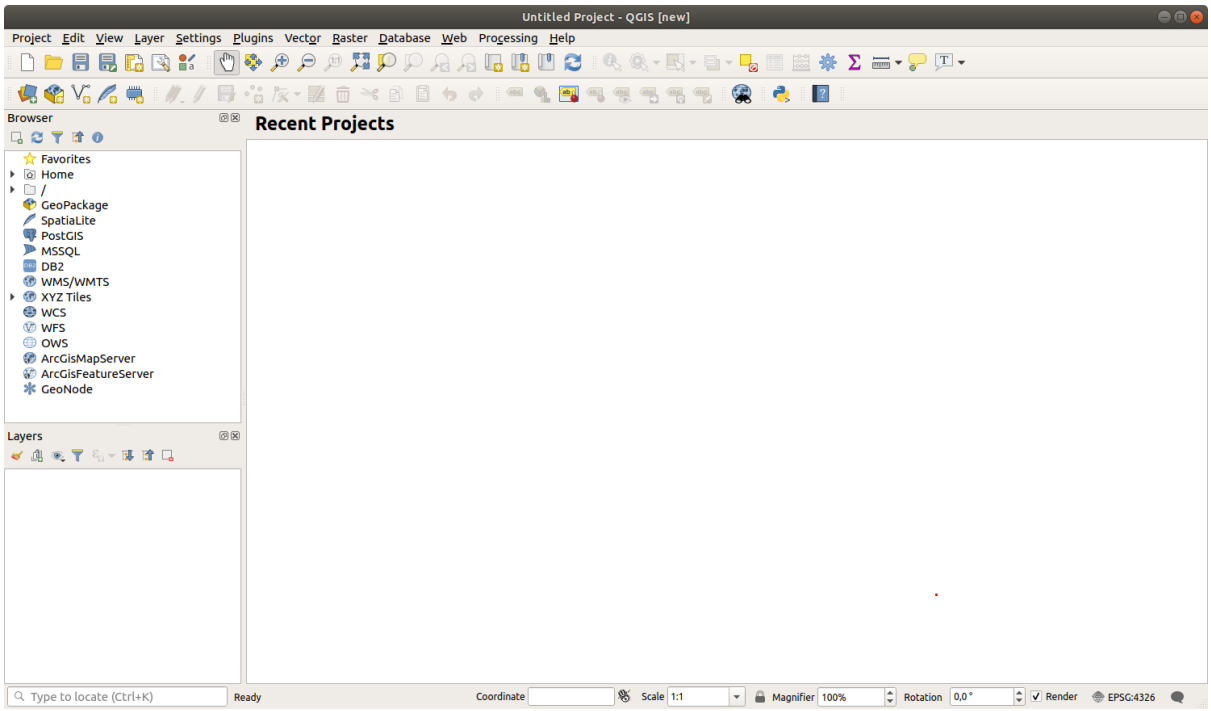
Launch QGIS from its desktop shortcut, menu item, etc., depending on how you configured its installation.

Bemerkung: The screenshots for this course were taken in QGIS 3.4 running on Linux. Depending on your setup, the screens you encounter may well appear somewhat different. However, all the same buttons will still be available, and the instructions will work on any OS. You will need QGIS 3.4 (the latest version at time of writing) to use this course.

Let's get started right away!

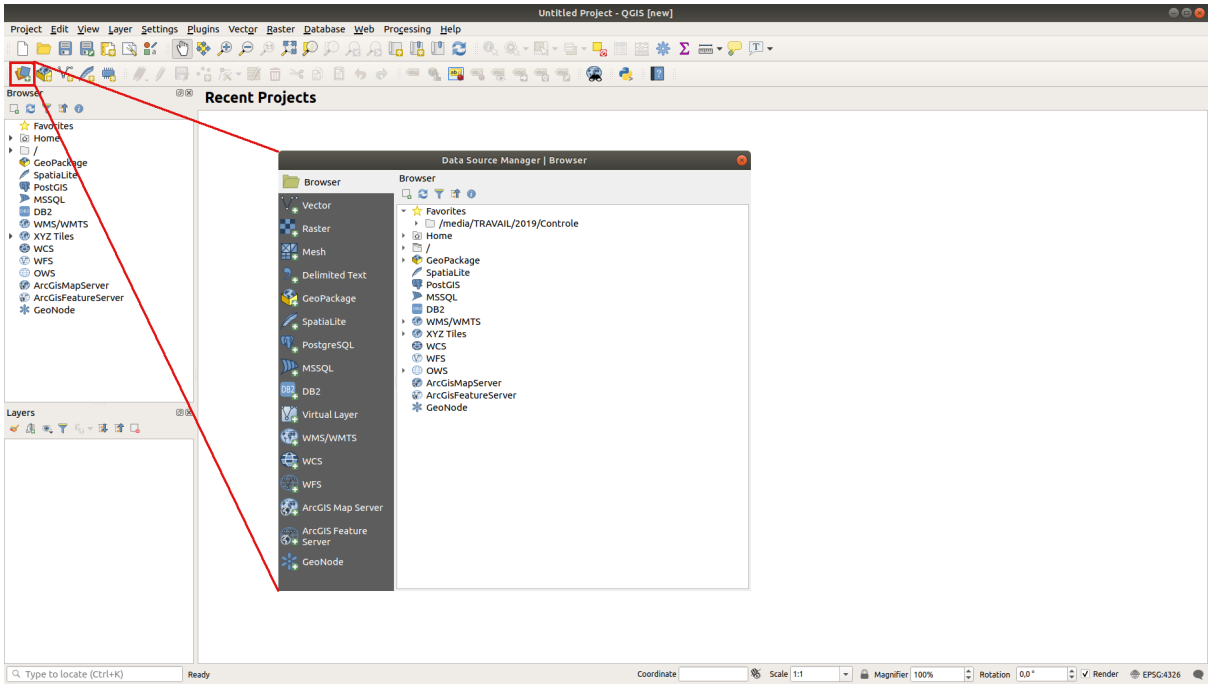
2.2.1 Follow Along: Prepare a map

1. Open QGIS. You will have a new, blank map.

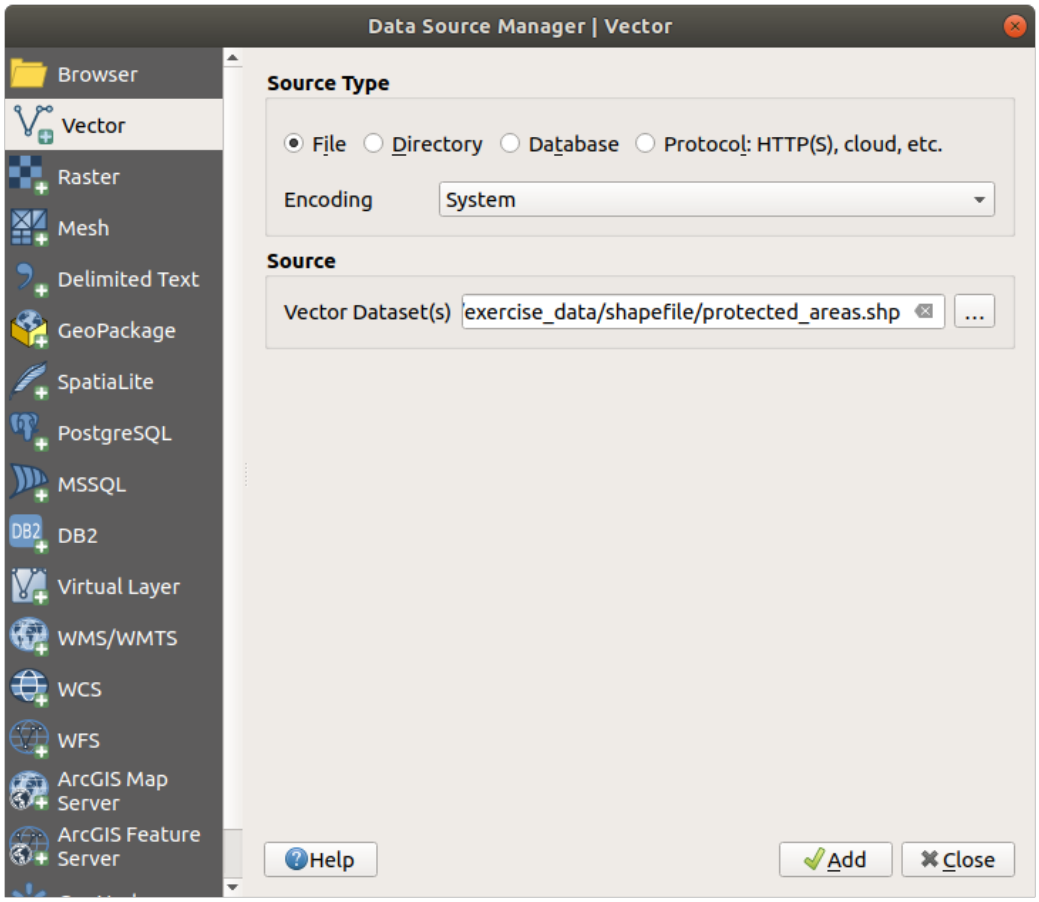


2. The *Data Source Manager* dialog allows you to choose the data to load depending on the data type. We'll use it to load our dataset: click the Open Data Source Manager button.

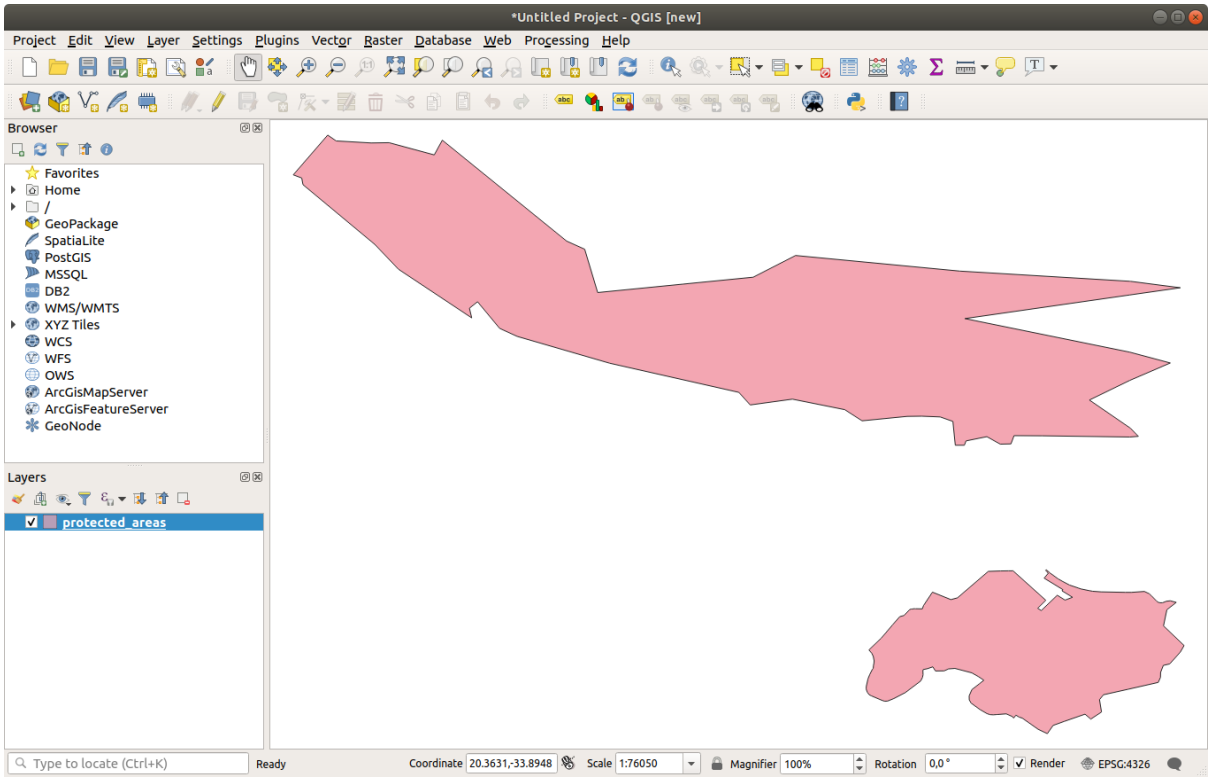
If you can't find the icon, check that the *Data Source Manager* toolbar is enabled in the *View* ► *Toolbars* menu.



3. Load the `protected_areas.shp` vector dataset:
    1. Click on the *Vector* tab.
    2. Enable the *File* source type.
    3. Press the … button next to *Vector Dataset(s)*.
    4. Select the `exercise_data/shapefile/protected_areas.shp` file in your training directory.
    5. Click *Open*. You will see the original dialog, with the file path filled in.



6. Click *Add* here as well. The data you specified will now load: you can see a `protected_areas` item in the *Layers* panel (bottom left) with its features shown in the main map canvas.



Congratulations! You now have a basic map. Now would be a good time to save your work.

1. Click on the *Save As* button:
2. Save the map under a `solution` folder next to `exercise_data` and call it `basic_map.qgz`.

2.2.2 Try Yourself

Repeat the steps above to add the `places.shp` and `rivers.shp` layers from the same folder (`exercise_data/shapefile`) to the map.

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 Abb. 2.1 below):

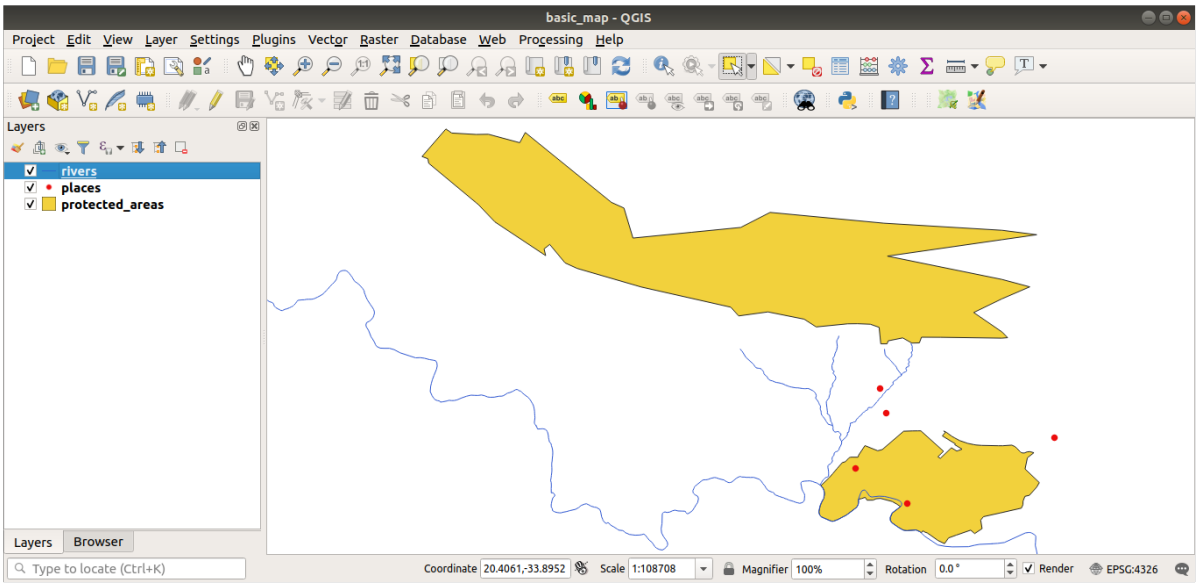


Abb. 2.1: A basic map

2.2.3 Follow Along: Loading vector data from a GeoPackage Database

Databases allow you to store a large volume of associated data in one file. You may already be familiar with a database management system (DBMS) such as Libreoffice Base or MS Access. GIS applications can also make use of databases. GIS-specific DBMSes (such as PostGIS) have extra functions, because they need to handle spatial data.

The GeoPackage open format is a container that allows you to store GIS data (layers) in a single file. Unlike the ESRI Shapefile format (e.g. the `protected_areas.shp` dataset you loaded earlier), a single GeoPackage file can contain various data (both vector and raster data) in different coordinate reference systems, as well as tables without spatial information; all these features allow you to share data easily and avoid file duplication.

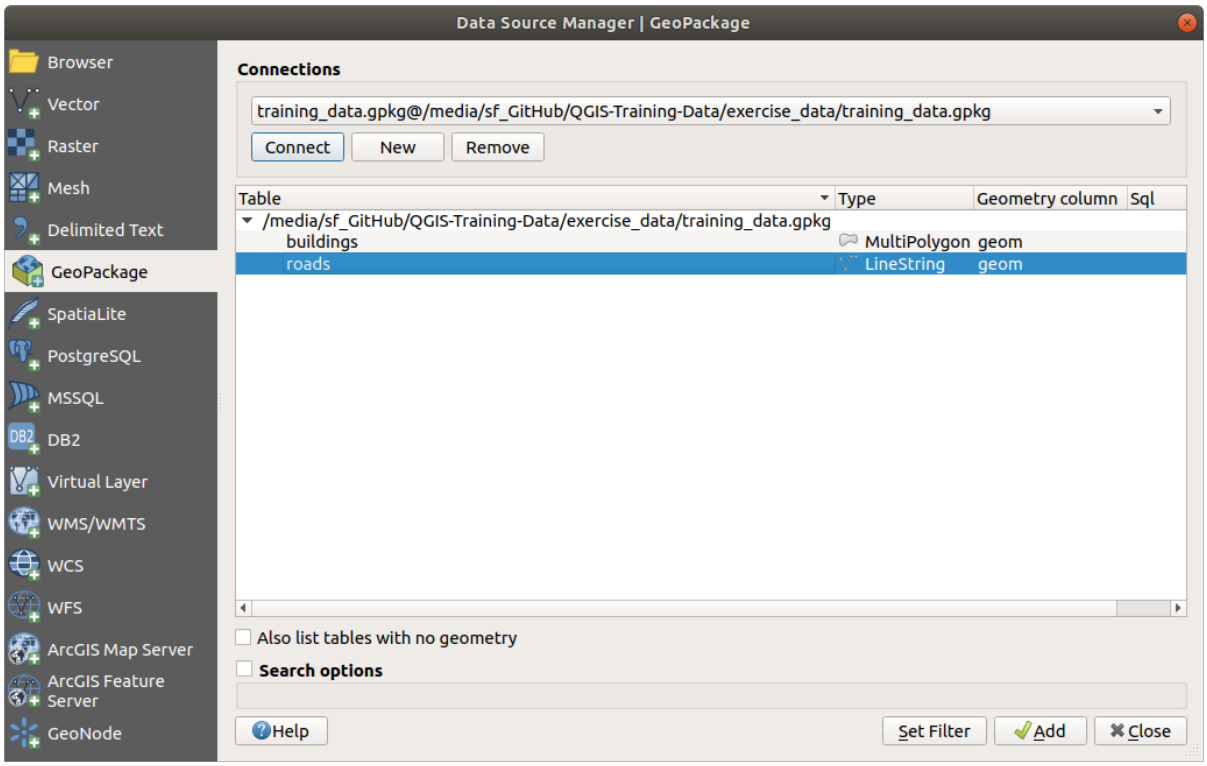
In order to load a layer from a GeoPackage, you will first need to create the connection to it:

1. Click on the Open Data Source Manager button.
2. On the left click on the *GeoPackage* tab.

3. Click on the *New* button and browse to the `training_data.gpkg` file in the `exercise_data` folder you downloaded before.
4. Select the file and press *Open*. The file path is now added to the Geopackage connections list, and appears in the drop-down menu.

You are now ready to add any layer from this GeoPackage to QGIS.

1. Click on the *Connect* button. In the central part of the window you should now see the list of all the layers contained in the GeoPackage file.
2. Select the *roads* layer and click on the *Add* button.



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

3. Click on *Close*.

Congratulations! You have loaded the first layer from a GeoPackage.

2.2.4 Follow Along: Loading vector data from a SpatiaLite Database with the Browser

QGIS provides access to many other database formats. Like GeoPackage, the SpatiaLite database format is an extension of the SQLite library. And adding a layer from a SpatiaLite provider follows the same rules as described above: Create the connection –> Enable it –> Add the layer(s).

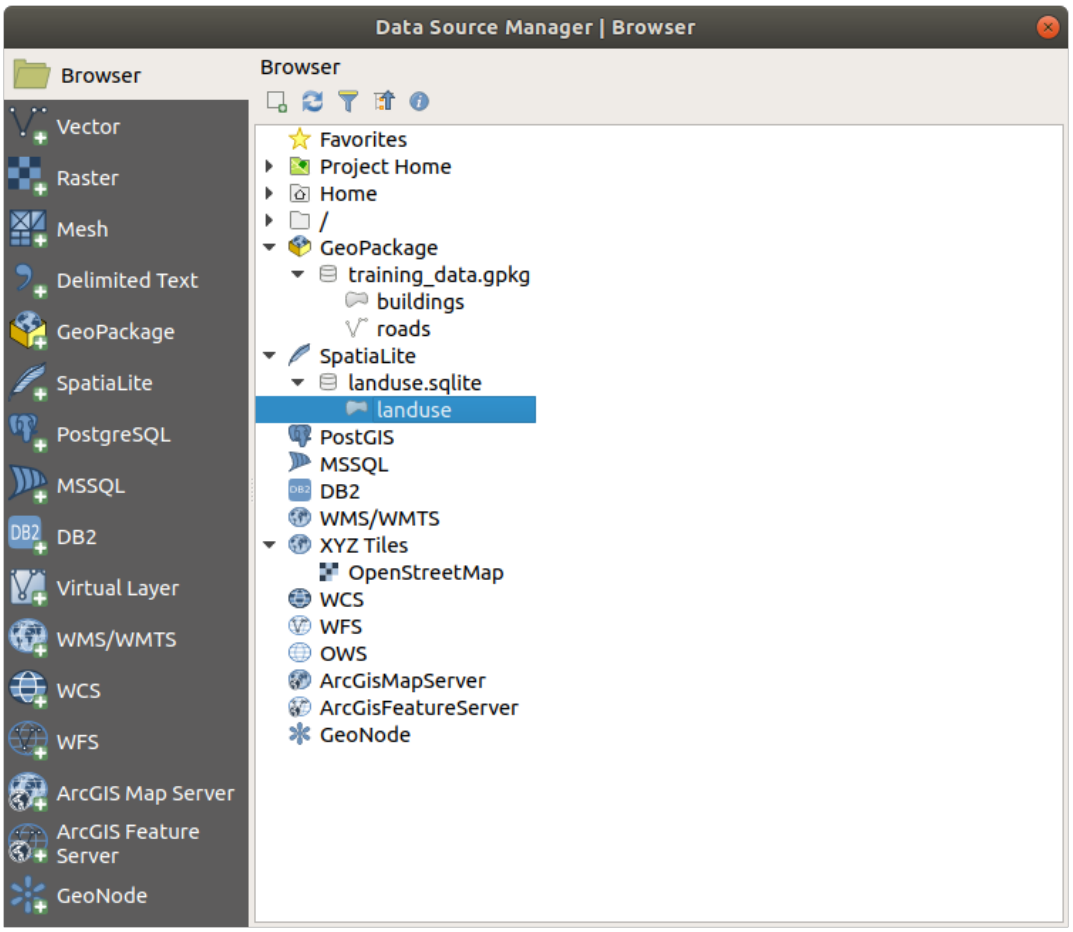
While this is one way to add SpatiaLite data to your map, let's explore another powerful way to add data: the *Browser*.

1. Click the icon to open the *Data Source Manager* window.
2. Click on the *Browser* tab.
3. In this tab you can see all the storage disks connected to your computer as well as entries for most of the tabs in the left. These allow quick access to connected databases or folders.

For example, click on the drop-down icon next to the *GeoPackage* entry. You'll see the `training-data.gpkg` file we previously connected to (and its layers, if expanded).

4. Right-click the *SpatiaLite* entry and select *New Connection…*.
5. Navigate to the `exercise_data` folder, select the `landuse.sqlite` file and click *Open*.

   Notice that a *landuse.sqlite* entry has been added under the *SpatiaLite* one.
6. Expand the *landuse.sqlite* entry.
7. Double-click the *landuse* layer or select and drag-and-drop it onto the map canvas. A new layer is added to the *Layers* panel and its features are displayed on the map canvas.



Tipp: Enable the *Browser* panel in *View* ► *Panels* ► and use it to add your data. It's a handy shortcut for the *Data Source Manager* ► *Browser* tab, with the same functionality.

Bemerkung: Remember to save your project frequently! The project file doesn't contain any of the data itself, but it remembers which layers you loaded into your map.

2.2.5 Try Yourself Load More Vector Data

Load the following datasets from the `exercise_data` folder into your map using any of the methods explained above:

- *buildings*
- *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`).

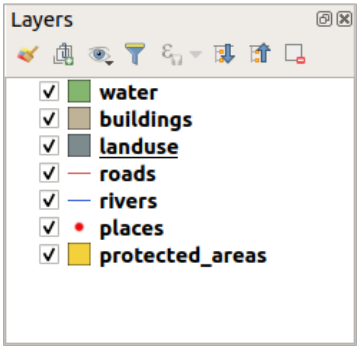
2.2.6 Follow Along: Reordering the Layers

The layers in your Layers list are drawn on the map in a certain order. The layer at the bottom of the list is drawn first, and the layer at the top is drawn last. By changing the order that they are shown on the list, you can change the order they are drawn in.

Bemerkung: You can alter this behavior using the *Control rendering order* checkbox beneath the *Layer Order* panel. We will however not discuss this feature yet.

The order in which the layers have been loaded into the map is probably not logical at this stage. It's possible that the road layer is completely hidden because other layers are on top of it.

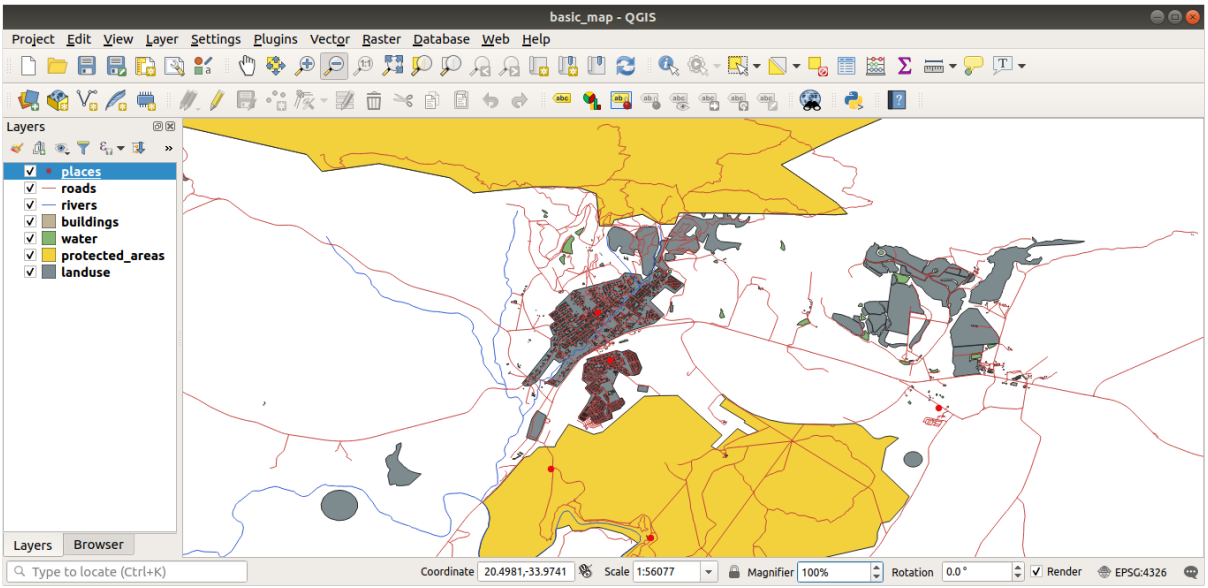
For example, this layer order…



… would result in roads and places being hidden as they run *underneath* the polygons of the landuse layer.

To resolve this problem:

1. Click and drag on a layer in the Layers list.
2. Reorder them to look like this:



You'll see that the map now makes more sense visually, with roads and buildings appearing above the land use regions.

2.2.7 In Conclusion

Now you've added all the layers you need from several different sources and created a basic map!

2.2.8 What's Next?

Now you're familiar with the basic function of the *Open Data Source Manager* button, but what about all the others? How does this interface work? Before we go on, let's take a look at some basic interaction with the QGIS interface. This is the topic of the next lesson.

2.3 Lesson: Navigating the Map Canvas

Dieser Abschnitt behandelt grundlegende Navigationswerkzeuge in QGIS, die zur Navigation in der Kartenansicht genutzt werden. Diese Werkzeuge erlauben es die Kartenlayer in verschiedenen Maßstäben visuell zu erkunden.

Ziel dieser Lerneinheit: Lerne die Werkzeuge Pan und Zoom in QGIS zu nutzen und lerne etwas über Kartenmaßstäbe.

2.3.1 Follow Along: Basic Navigation Tools

Before learning how to navigate within the Map Canvas, let's add some layers that we can explore during this tutorial.

1. Open a new blank project and using the steps learnt in *Create a Map*, load the previously seen `protected_areas`, `roads` and `buildings` layers to the project. The result view should look similar to the snippet in Abb. 2.2 below (colors do not matter):

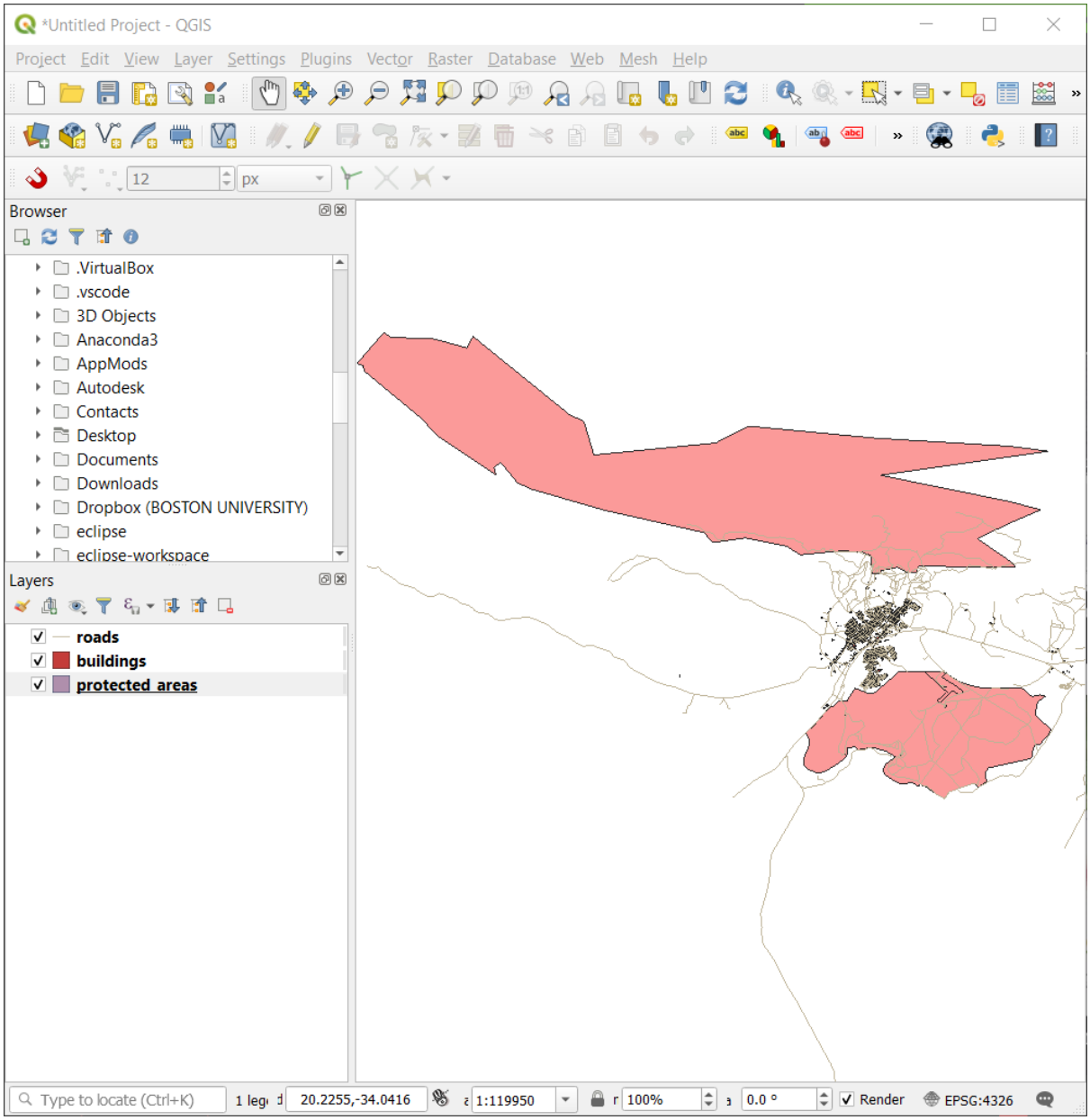


Abb. 2.2: Protected areas, roads and buildings added

Let's first learn how to use the Pan Tool.

1. In the *Map Navigation Toolbar*, make sure the Pan button is activated.
2. Move the mouse to the center of the Map Canvas area.
3. Left-click and hold, and drag the mouse in any direction to pan the map.

Next, let's zoom in and take a closer look at the layers we imported.

1. In the *Map Navigation Toolbar*, click on the Zoom In button.
2. Move your mouse to approximately the top left area of where there is the highest density of buildings and roads.
3. Left click and hold.
4. Then drag the mouse, which will create a rectangle, and cover the dense area of buildings and roads (Abb. 2.3).

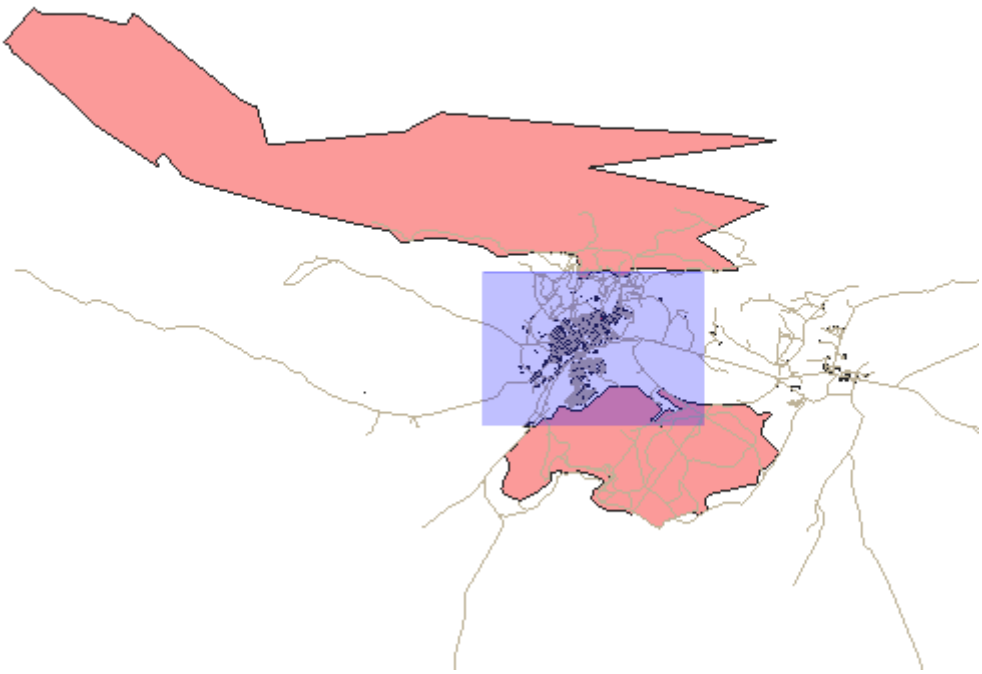
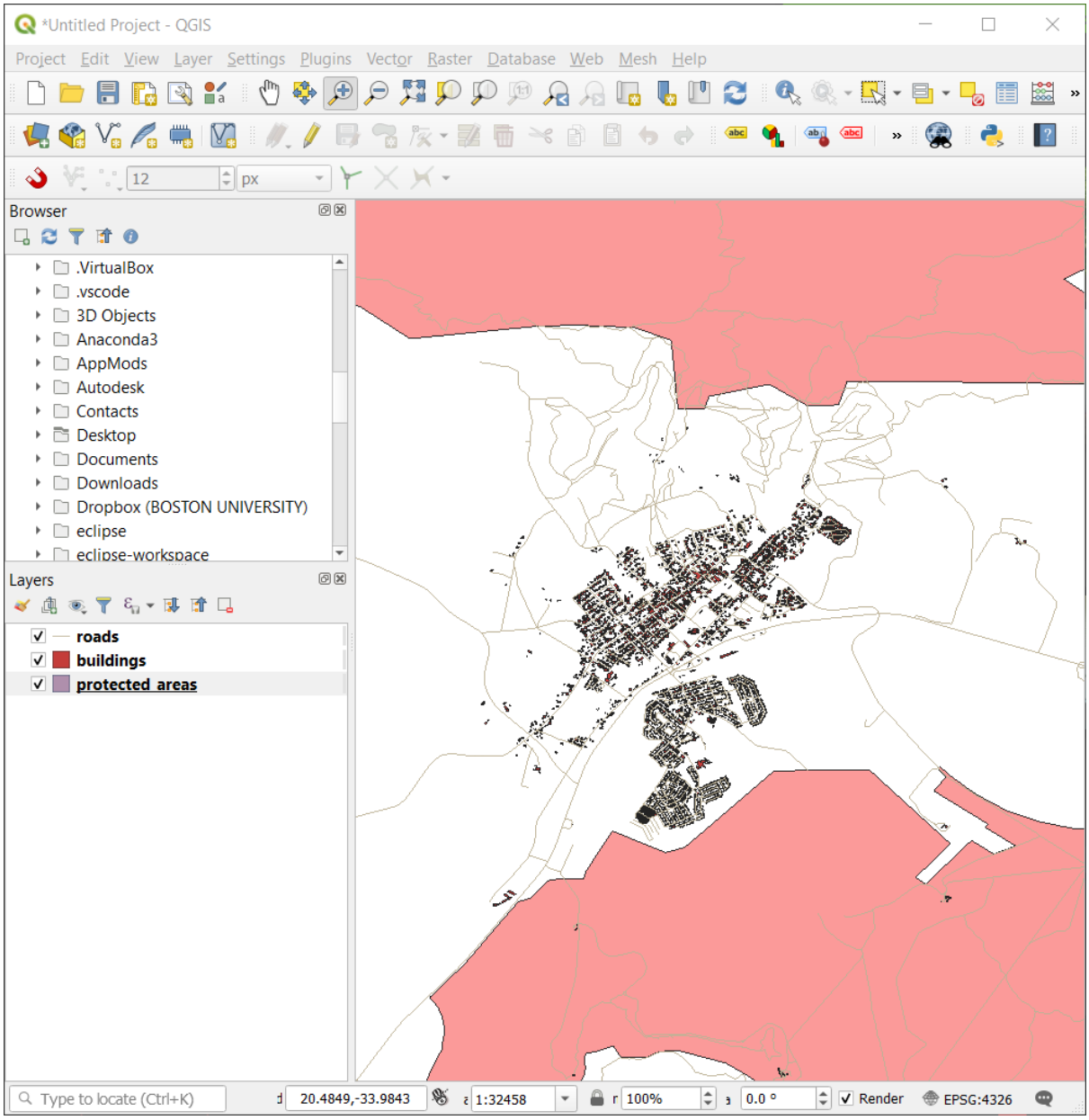


Abb. 2.3: Zoom in

5. Release the left click. This will zoom in to include the area that you selected with your rectangle.



6. To zoom out, select the Zoom Out button and perform the same action as you did for zooming in.

As you pan, zoom in, or zoom out, QGIS saves these views in a history. This allows you to backtrack to a previous view.

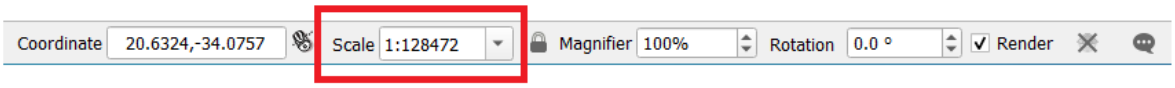
1. In the *Map Navigation Toolbar*, click on Zoom Last button to go to your previous view.

2. Click on Zoom Next button to proceed to move forward in your history.

Sometimes after exploring the data, we need to reset our view to the extent of all the layers. Instead of trying to use the Zoom Out tool multiple times, QGIS provides us with a button to do that action for us.

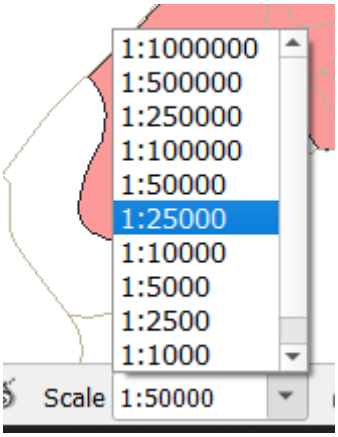
1. Click on the Zoom Full Extent button.

As you zoomed in and out, notice that the *Scale* value in the Status Bar changes. The *Scale* value represents the Map Scale. In general, the number to the right of *:* represents how many times smaller the object you are seeing in the Map Canvas is to the actual object in the real world.



You can also use this field to set the Map Scale manually.

1. In the Status Bar, click on the *Scale* textfield.
2. Type in `50000` and press `Enter`. This will redraw the features in the Map Canvas to reflect the scale you typed in.
3. Alternatively, click on the options arrow of the *Scale* field to see the preset map scales.



4. Select *1:5000*. This will also update the map scale in the Map Canvas.

Now you know the basics of navigating the Map Canvas. Check out the User Manual on Zooming and Panning to learn about alternative ways of navigating the Map Canvas.

2.3.2 In Conclusion

Knowing how to navigate the Map Canvas is important, as it allows one to explore and visually inspect the layers. This could be done for initial data exploration, or to validate output of a spatial analysis.

2.4 Lesson: Symbologie

The symbology of a layer is its visual appearance on the map. The basic strength of GIS over other ways of representing data with spatial aspects is that with GIS, you have a dynamic visual representation of the data you're working with.

Therefore, the visual appearance of the map (which depends on the symbology of the individual layers) is very important. The end user of the maps you produce will need to be able to easily see what the map represents. Equally as important, you need to be able to explore the data as you're working with it, and good symbology helps a lot.

In other words, having proper symbology is not a luxury or just nice to have. In fact, it's essential for you to use a GIS properly and produce maps and information that people will be able to use.

The goal for this lesson: To be able to create any symbology you want for any vector layer.

2.4.1 Follow Along: Changing Colors

To change a layer's symbology, open its *Layer Properties*. Let's begin by changing the color of the *landuse* layer.

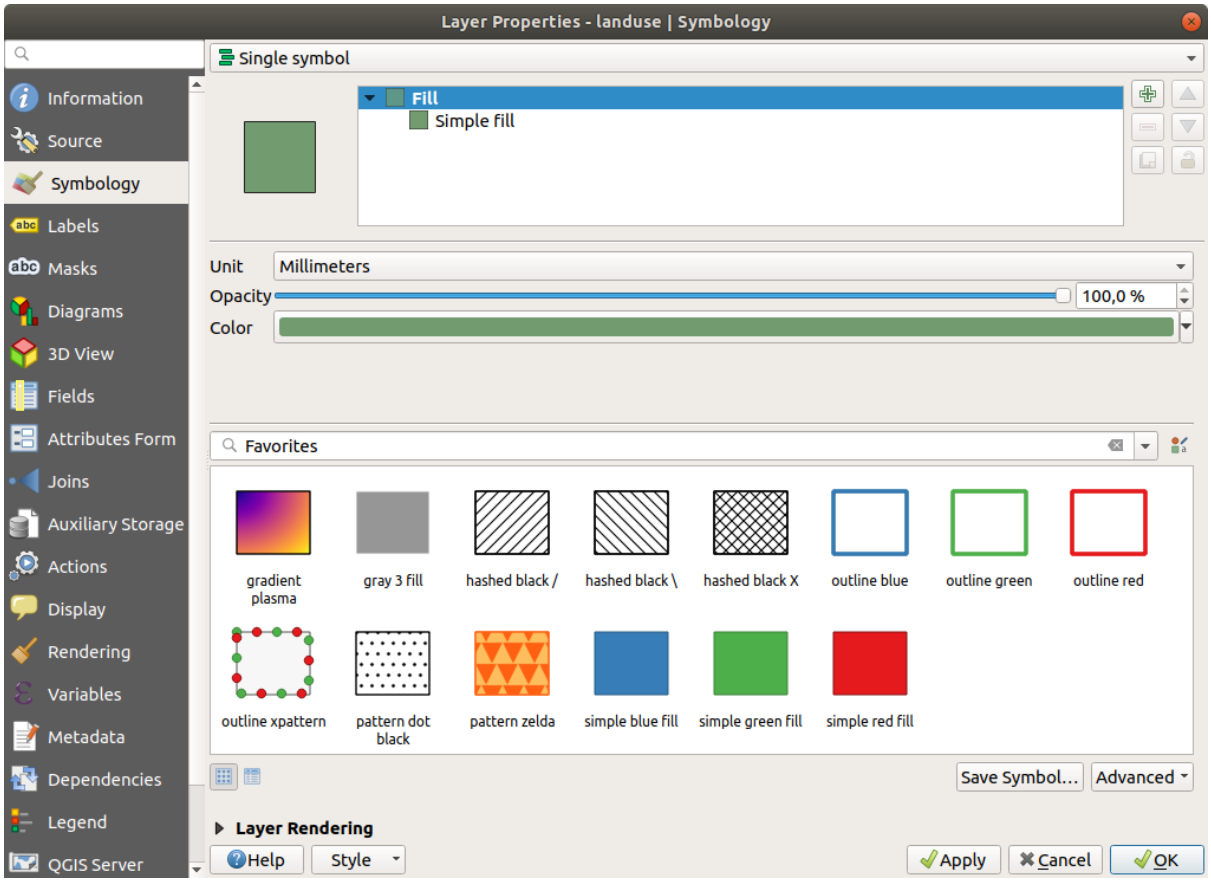
1. Right-click on the *landuse* layer in the layers list.
2. Select the menu item *Properties...* in the menu that appears.

Bemerkung: By default, you can also access a layer's properties by double-clicking on the layer in the Layers list.

Tipp: The button at the top of the *Layers* panel will open the *Layer Styling* panel. You can use this panel to change some properties of the layer: by default, changes will be applied immediately!

---

3. In the *Layer Properties* window, select the *Symbology* tab:



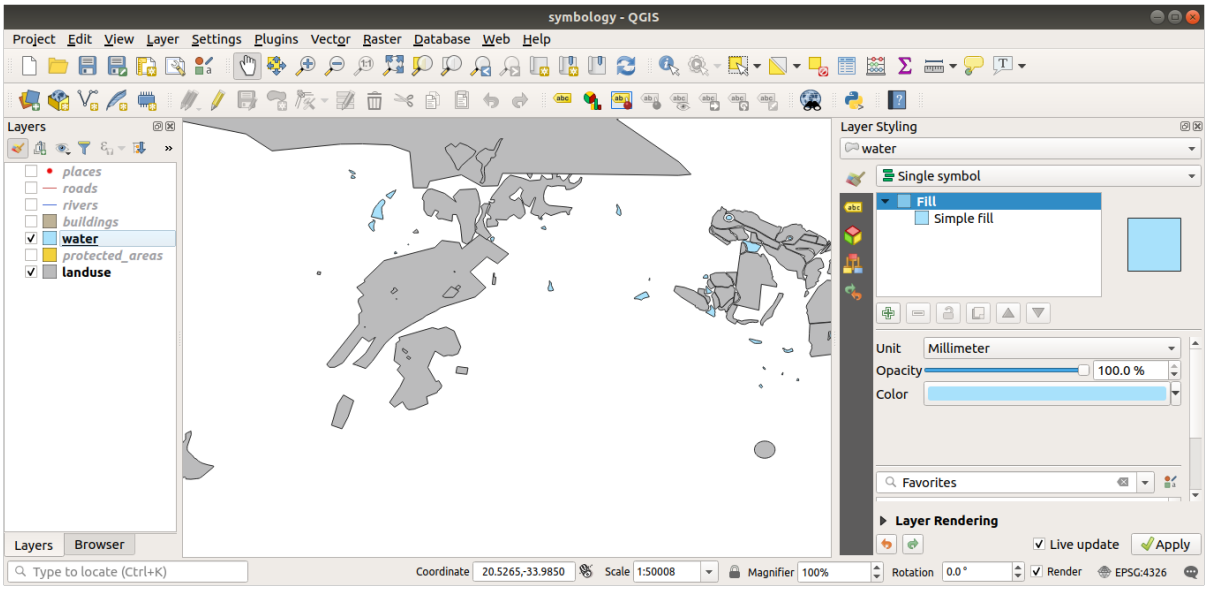
4. Click the color select button next to the *Color* label. A standard color dialog will appear.
5. Choose a gray color and click *OK*.
6. Click *OK* again in the *Layer Properties* window, and you will see the color change being applied to the layer.

2.4.2 Try Yourself

Change the color of the *water* layer to light blue. Try to use the *Layer Styling* panel instead of the *Layer Properties* menu.

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.

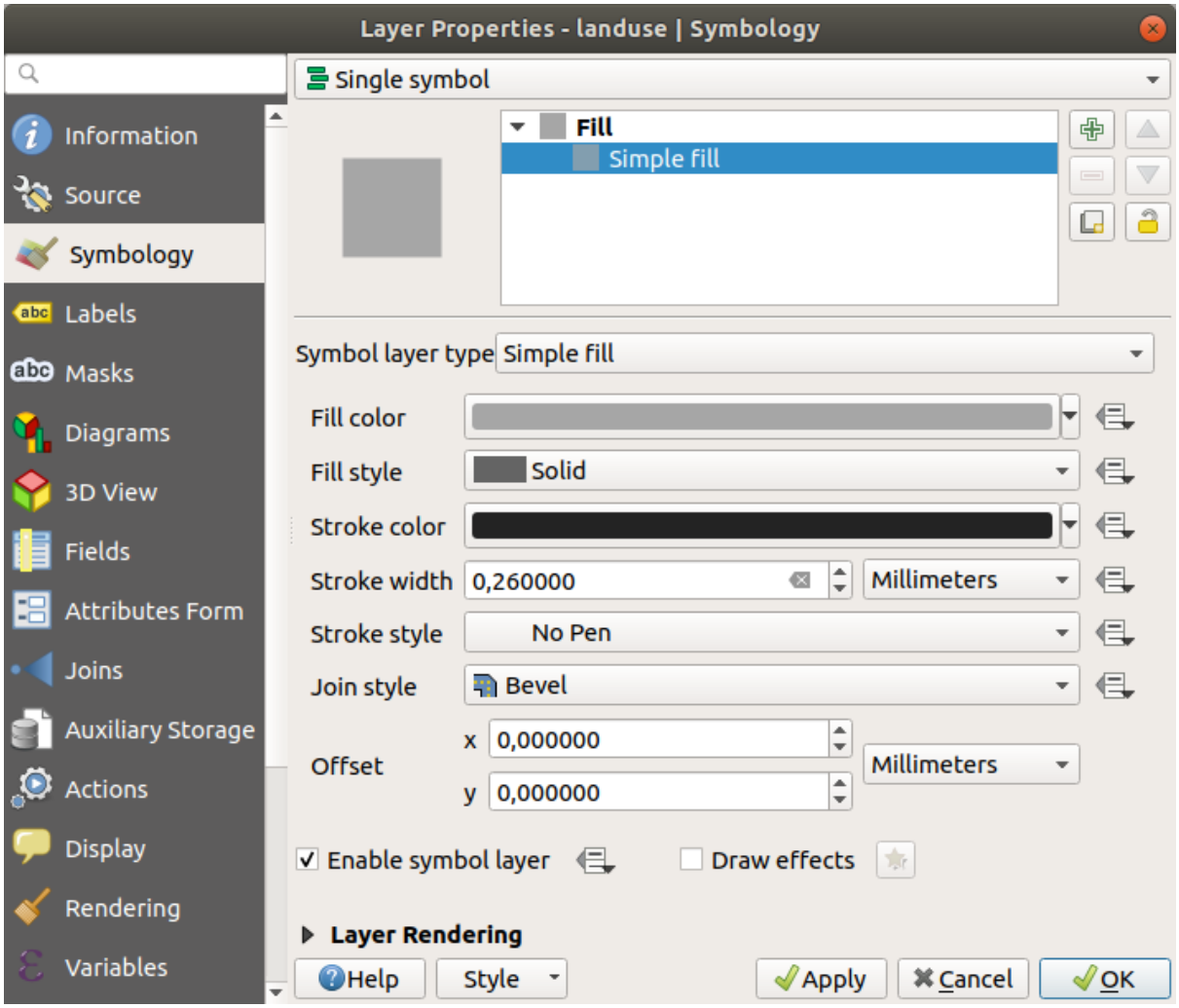
2.4.3 Follow Along: Changing Symbol Structure

This is good stuff so far, but there's more to a layer's symbology than just its color. Next we want to eliminate the lines between the different land use areas so as to make the map less visually cluttered.

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

   Under the *Symbology* tab, you will see the same kind of dialog as before. This time, however, you're doing more than just quickly changing the color.

2. In the symbol layers tree, expand the *Fill* dropdown and select the *Simple fill* option.
3. Click on the *Stroke style* dropdown. At the moment, it should be showing a short line and the words *Solid Line*.
4. Change this to *No Pen*.



5. Click *OK*.

Now the *landuse* layer won't have any lines between areas.

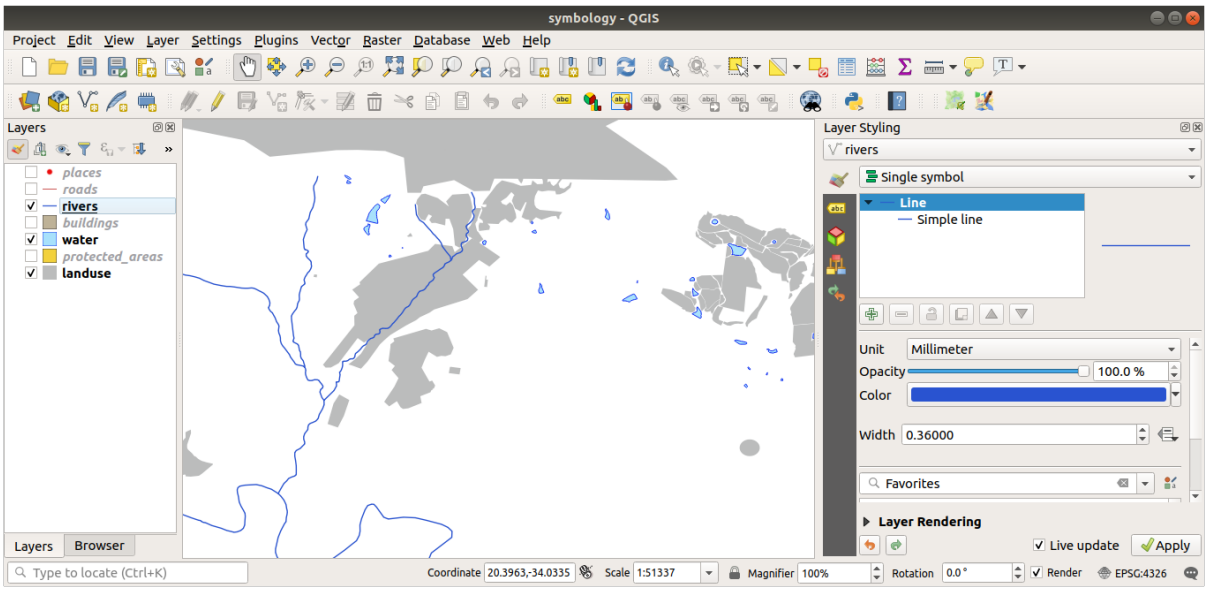
2.4.4 Try Yourself

- Change the *water* layer's symbology again so that it has a darker blue outline.
- Change the *rivers* layer's symbology to a sensible representation of waterways.

Remember: you can use the Open the Layer Styling panel button and see all the changes instantly. That panel also allows you to undo individual changes while symbolizing a layer.

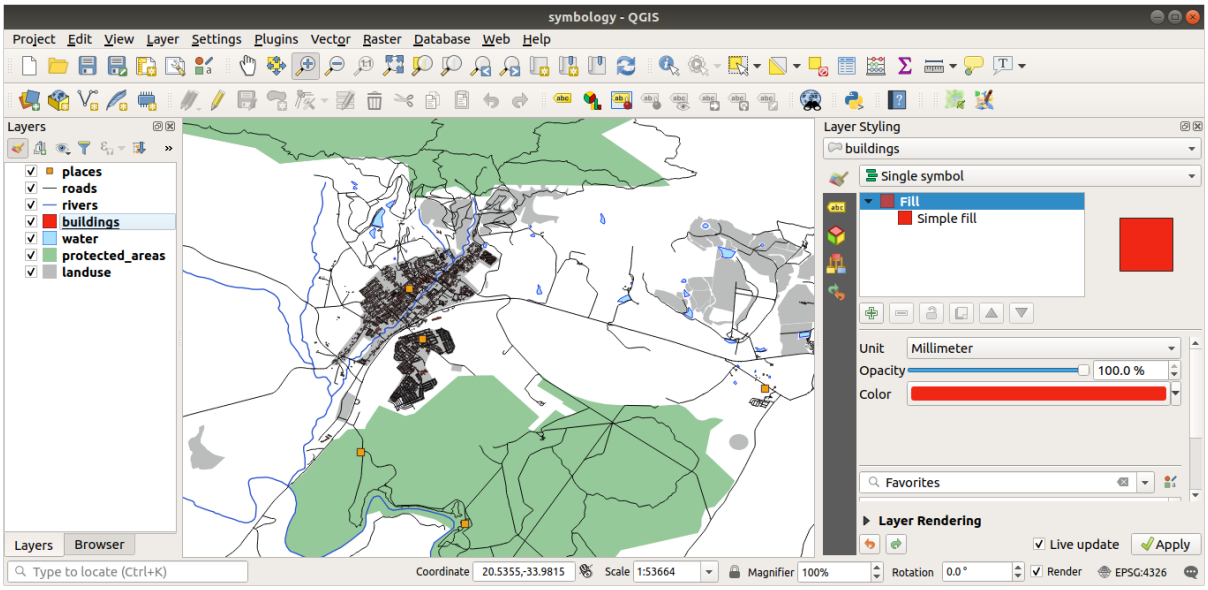
Answer

Your map should now look like this:



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

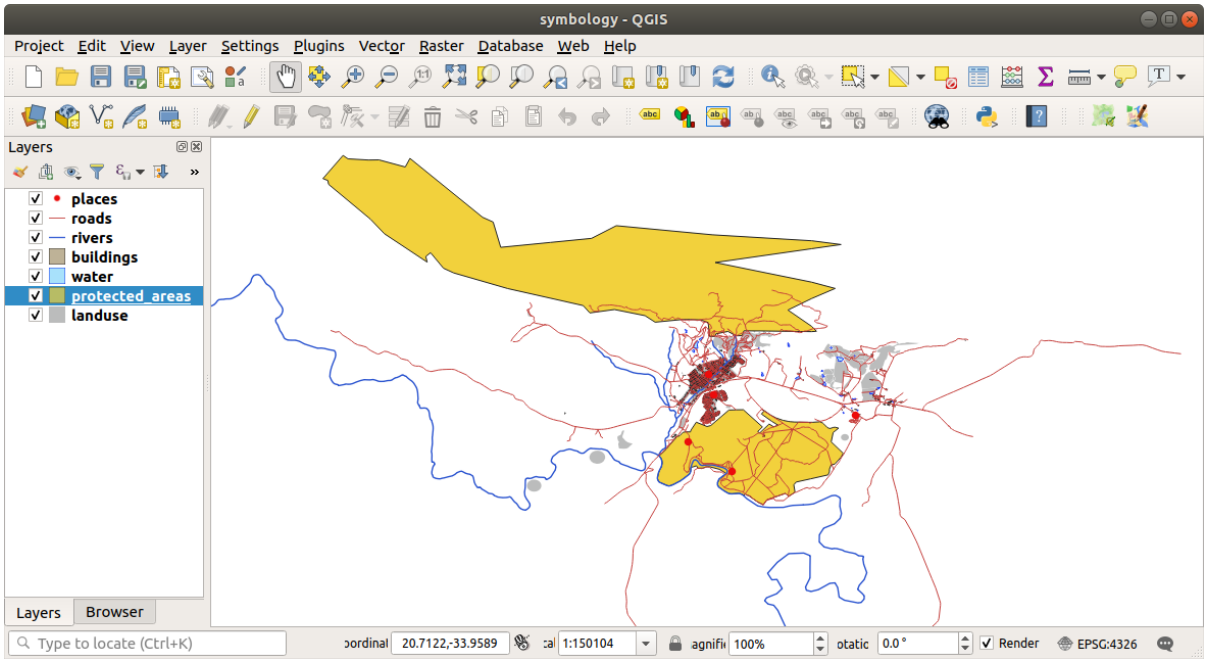
- Use the method above to change the colors and styles for all the remaining layers.
- Try using natural colors for the objects. For example, a road should not be red or blue, but can be gray or black.
- Also feel free to experiment with different *Fill style* and *Stroke style* settings for the polygons.



2.4.5 Follow Along: Scale-Based Visibility

Sometimes you will find that a layer is not suitable for a given scale. For example, a dataset of all the continents may have low detail, and not be very accurate at street level. When that happens, you want to be able to hide the dataset at inappropriate scales.

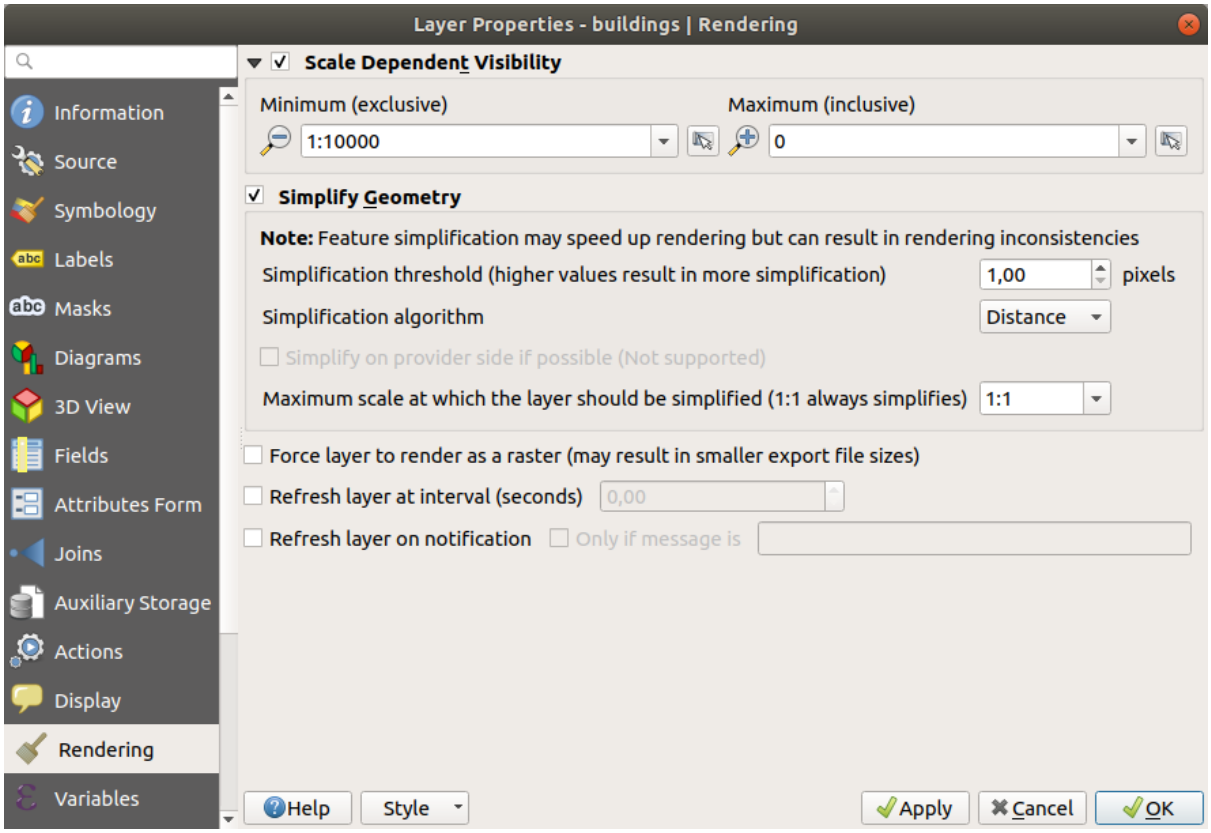
In our case, we may decide to hide the buildings from view at small scales. This map, for example…



… is not very useful. The buildings are hard to distinguish at that scale.

To enable scale-based rendering:

1. Open the *Layer Properties* dialog for the *buildings* layer.
2. Activate the *Rendering* tab.
3. Enable scale-based rendering by clicking on the checkbox labeled *Scale dependent visibility*:
4. Change the *Minimum* value to `1:10000`.



5. Click *OK*.

Test the effects of this by zooming in and out in your map, noting when the *buildings* layer disappears and reappears.

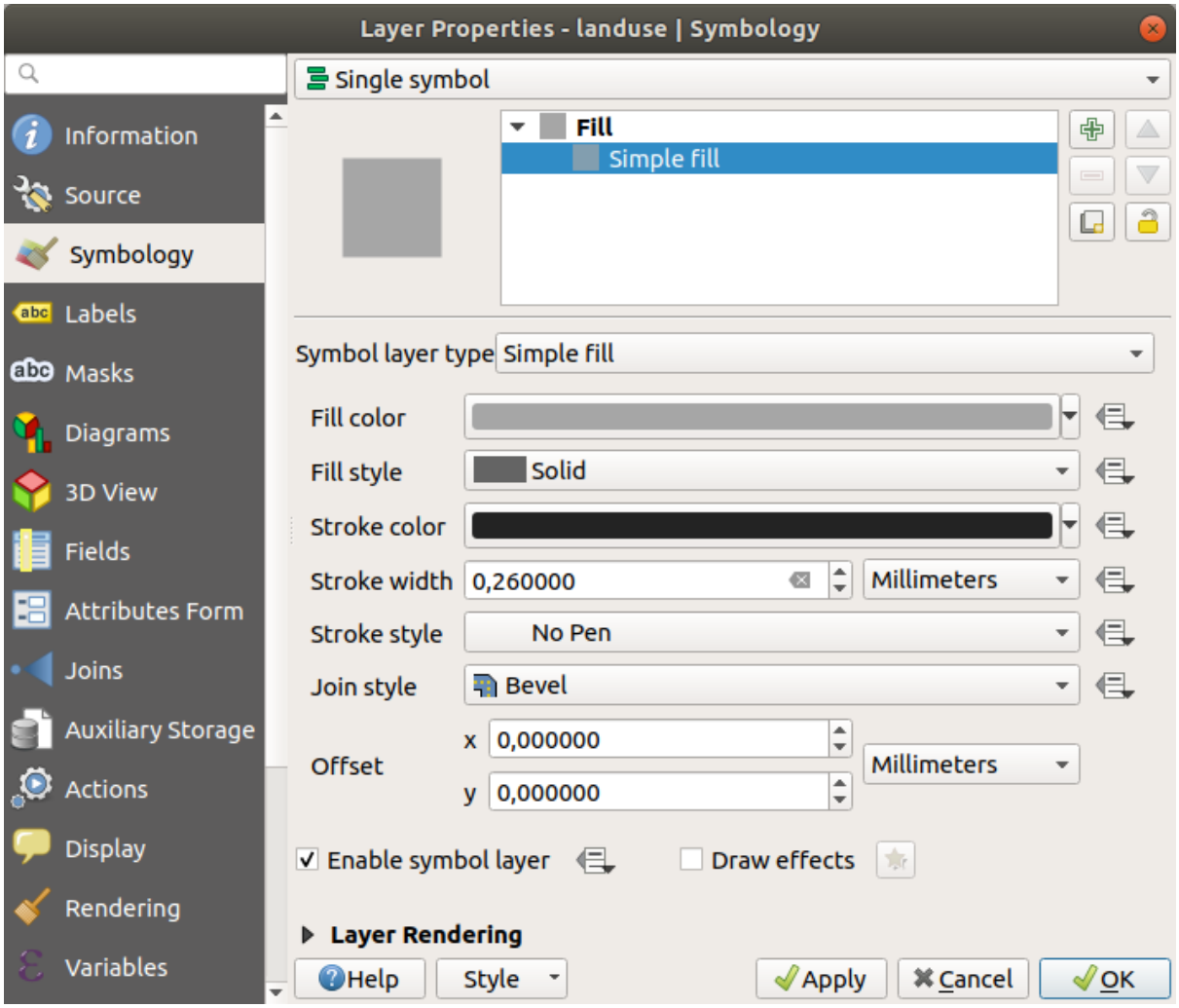
Bemerkung: You can use your mouse wheel to zoom in increments. Alternatively, use the zoom tools to zoom to a window:

2.4.6 Follow Along: Adding Symbol Layers

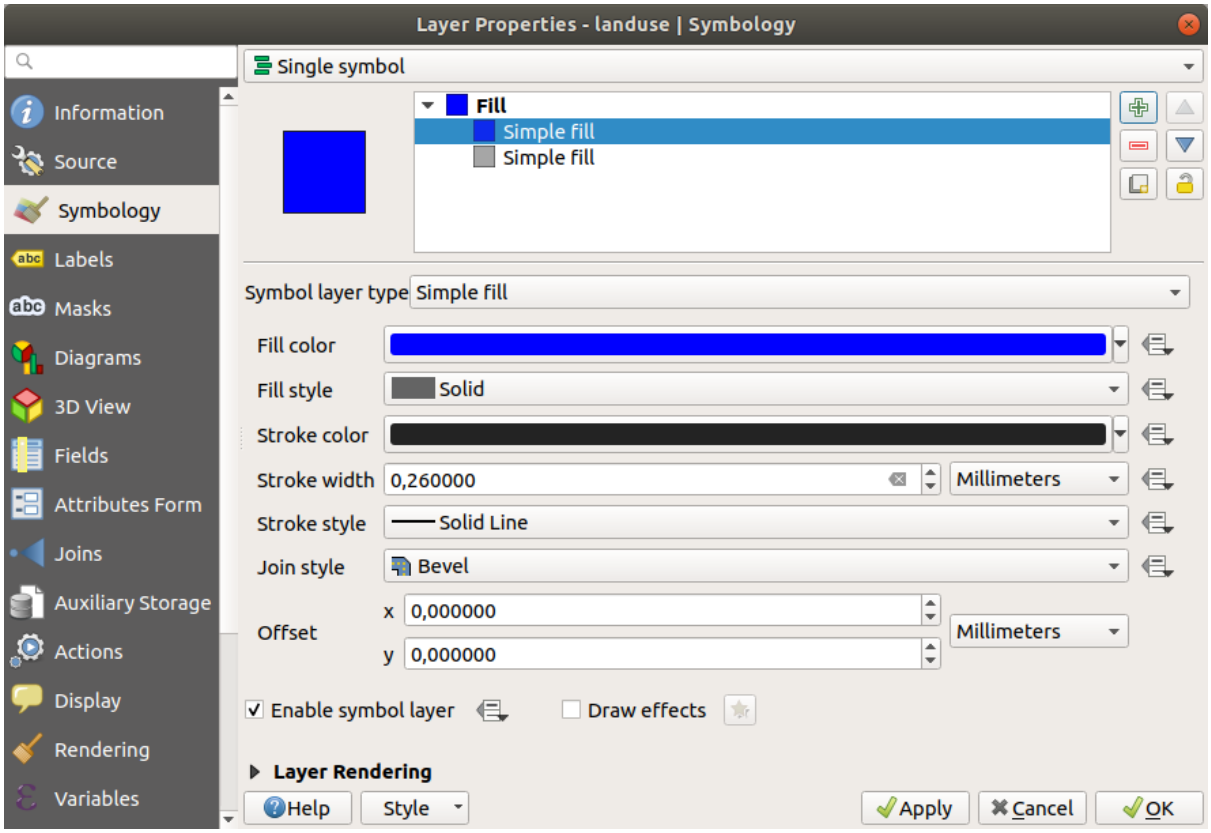
Now that you know how to change simple symbology for layers, the next step is to create more complex symbology. QGIS allows you to do this using symbol layers.

1. Go back to the *landuse* layer's symbol properties panel (by clicking *Simple fill* in the symbol layers tree).

   In this example, the current symbol has no outline (i.e., it uses the *No Pen* border style).



2. Select the *Fill* level in the tree and click the Add symbol layer button. The dialog will change to look something like this, with a new symbol layer added:



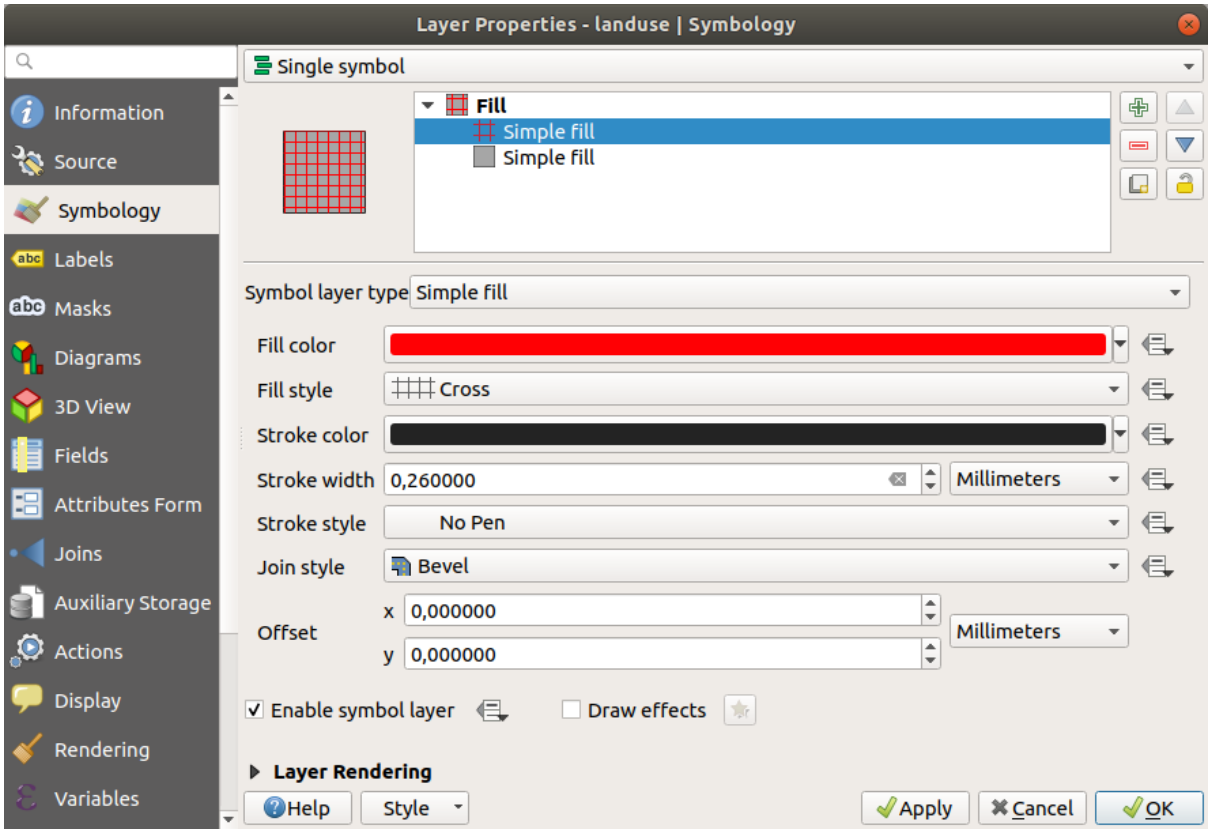
It may appear somewhat different in color, for example, but you're going to change that anyway.

Now there's a second symbol layer. Being a solid color, it will of course completely hide the previous kind of symbol. Plus, it has a *Solid Line* border style, which we don't want. Clearly this symbol has to be changed.

Bemerkung: It's important not to get confused between a map layer and a symbol layer. A map layer is a vector (or raster) that has been loaded into the map. A symbol layer is part of the symbol used to represent a map layer. This course will usually refer to a map layer as just a layer, but a symbol layer will always be called a symbol layer, to prevent confusion.

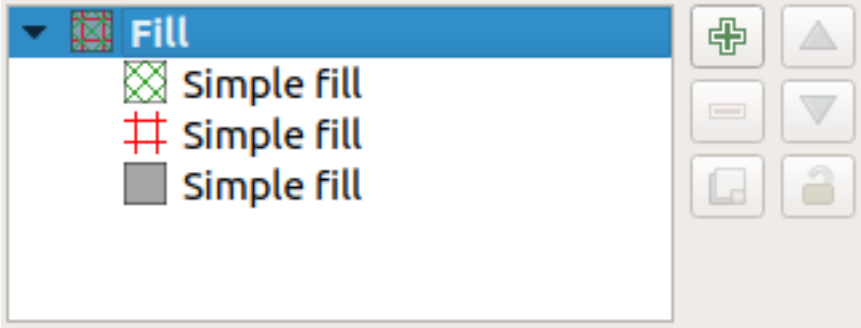
With the new *Simple Fill* symbol layer selected:

1. Set the border style to *No Pen*, as before.
2. Change the fill style to something other than *Solid* or *No brush*. For example:



3. Click *OK*.

Now you can see your results and tweak them as needed. You can even add multiple extra symbol layers and create a kind of texture for your layer that way.



It's fun! But it probably has too many colors to use in a real map…

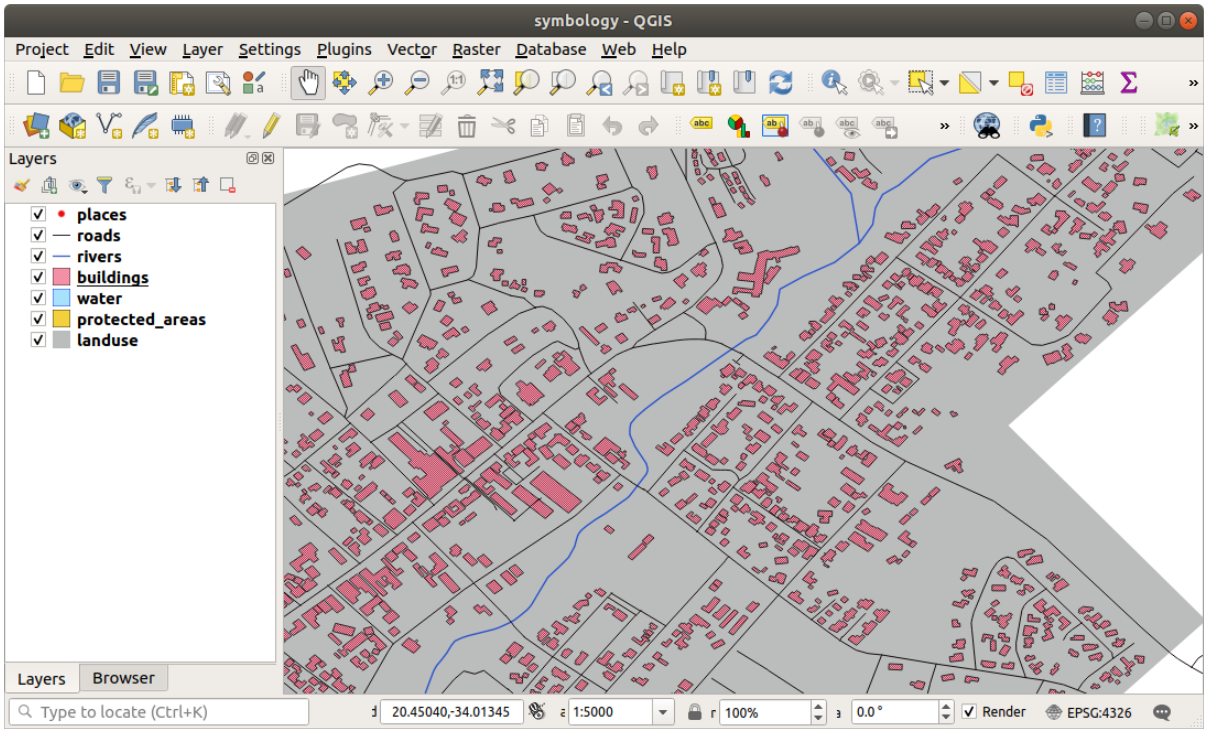
2.4.7 Try Yourself

Remembering to zoom in if necessary, create a simple, but not distracting texture for the *buildings* layer using the methods above.

Answer

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

Im Folgenden ein Beispiel:



2.4.8 Follow Along: Ordering Symbol Levels

When symbol layers are rendered, they are also rendered in a sequence, similar to the way the different map layers are rendered. This means that in some cases, having many symbol layers in one symbol can cause unexpected results.

1. Give the *roads* layer an extra symbol layer (using the method for adding symbol layers demonstrated above).
2. Give the base line a *Stroke width* of `1.5` and a black color.
3. Give the new, uppermost layer a thickness of `0.8` and a white color.

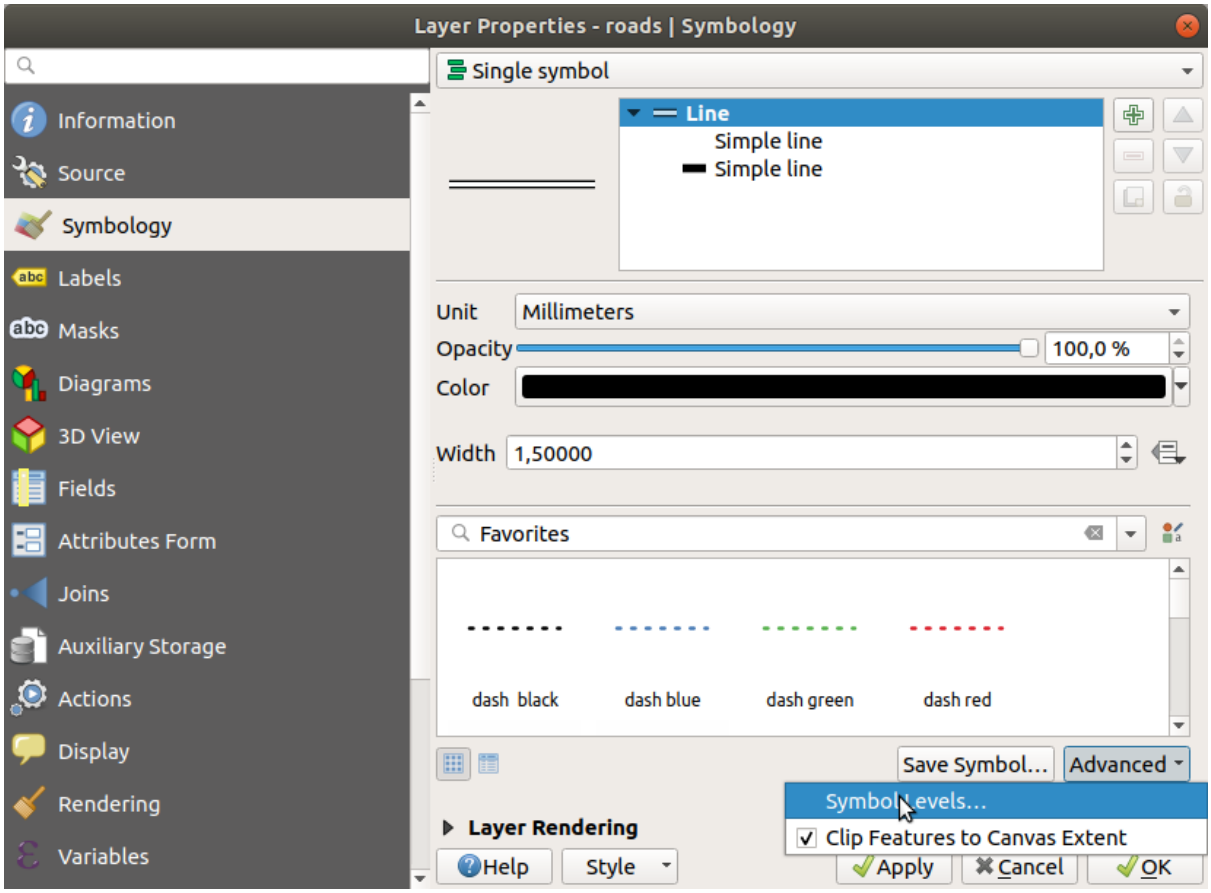
You'll notice that this happens:

Well, roads have now a *street* like symbology, but you see that lines are overlapping each others at each cross. That's not what we want at all!

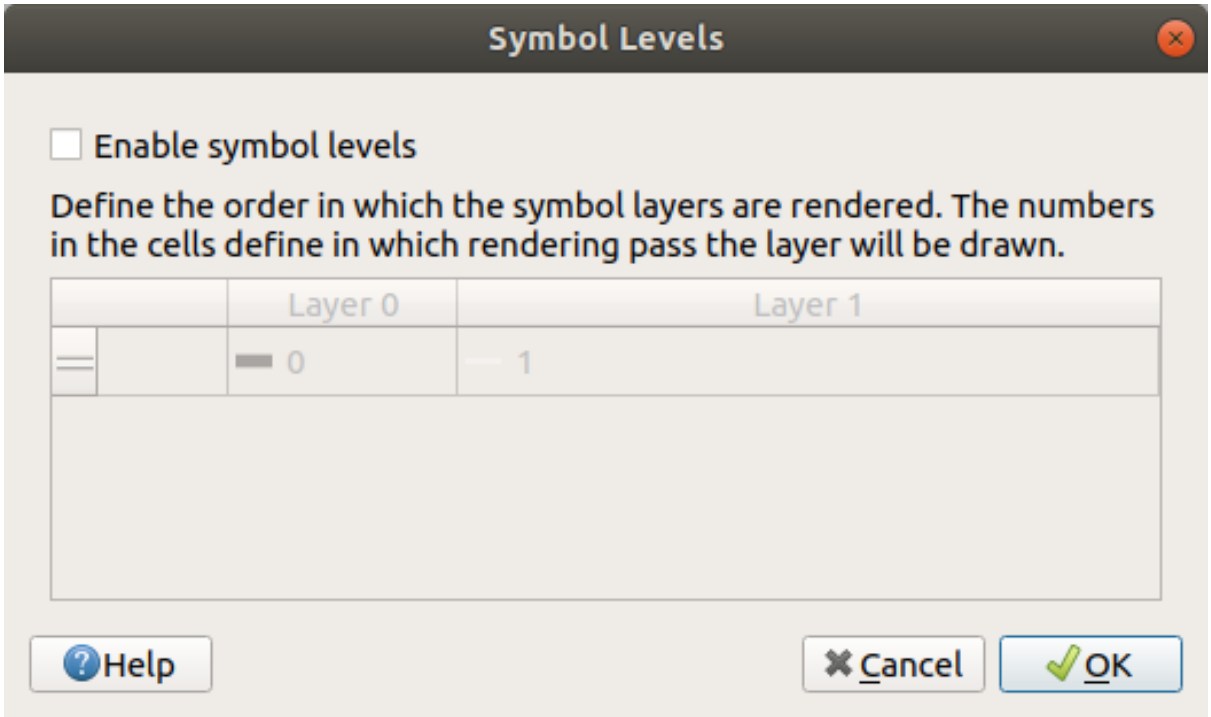
To prevent this from happening, you can sort the symbol levels and thereby control the order in which the different symbol layers are rendered.

To change the order of the symbol layers:

1. Select the topmost *Line* layer in the symbol layers tree.
2. Click *Advanced* ► *Symbol levels...* in the bottom right-hand corner of the window.

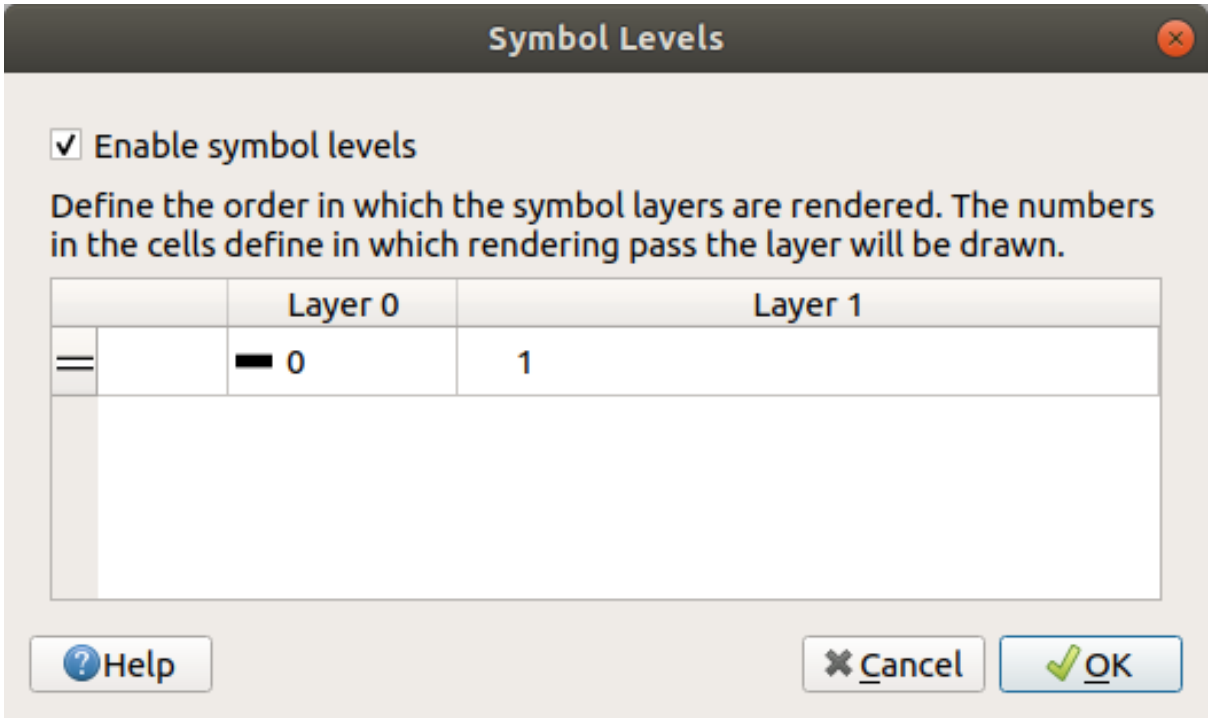


This will open a dialog like this:



3. Check *Enable symbol levels*. You can then set the layer order of each symbol by entering the corresponding level number. 0 is the bottom layer.

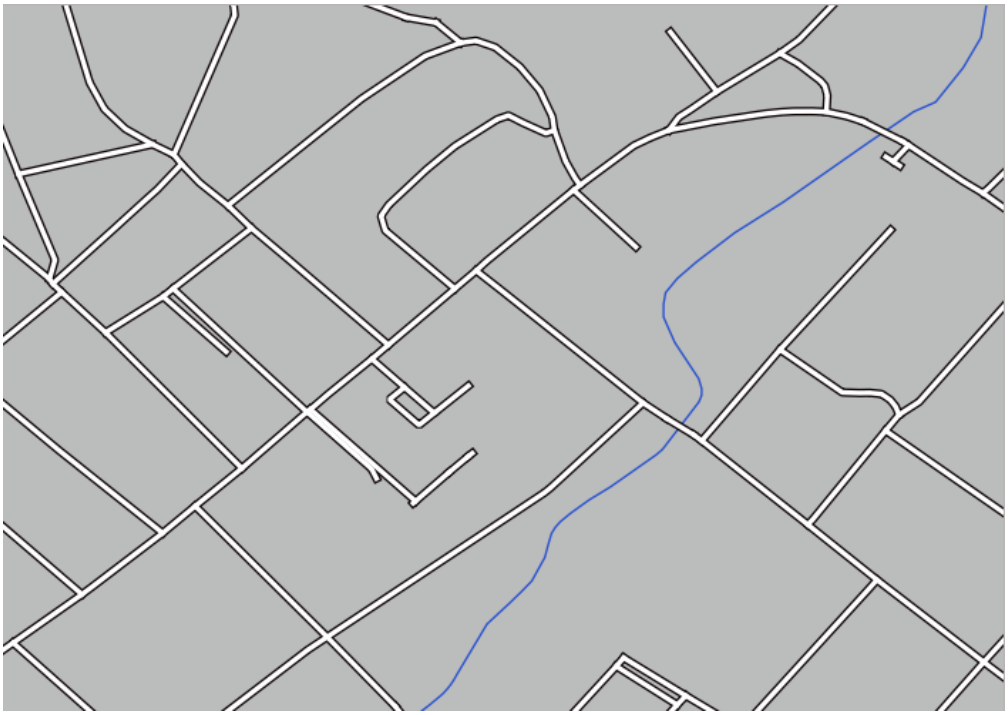
In our case, we just want to activate the option, like this:



This will render the white line above the thick black line borders:

4. Click *OK* twice to return to the map.

   The map will now look like this:



When you're done, remember to save the symbol itself so as not to lose your work if you change the symbol again in the future. You can save your current symbol style by clicking the *Save Style…* button at the bottom of the *Layer Properties* dialog. We will be using the *QGIS QML Style File* format.

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

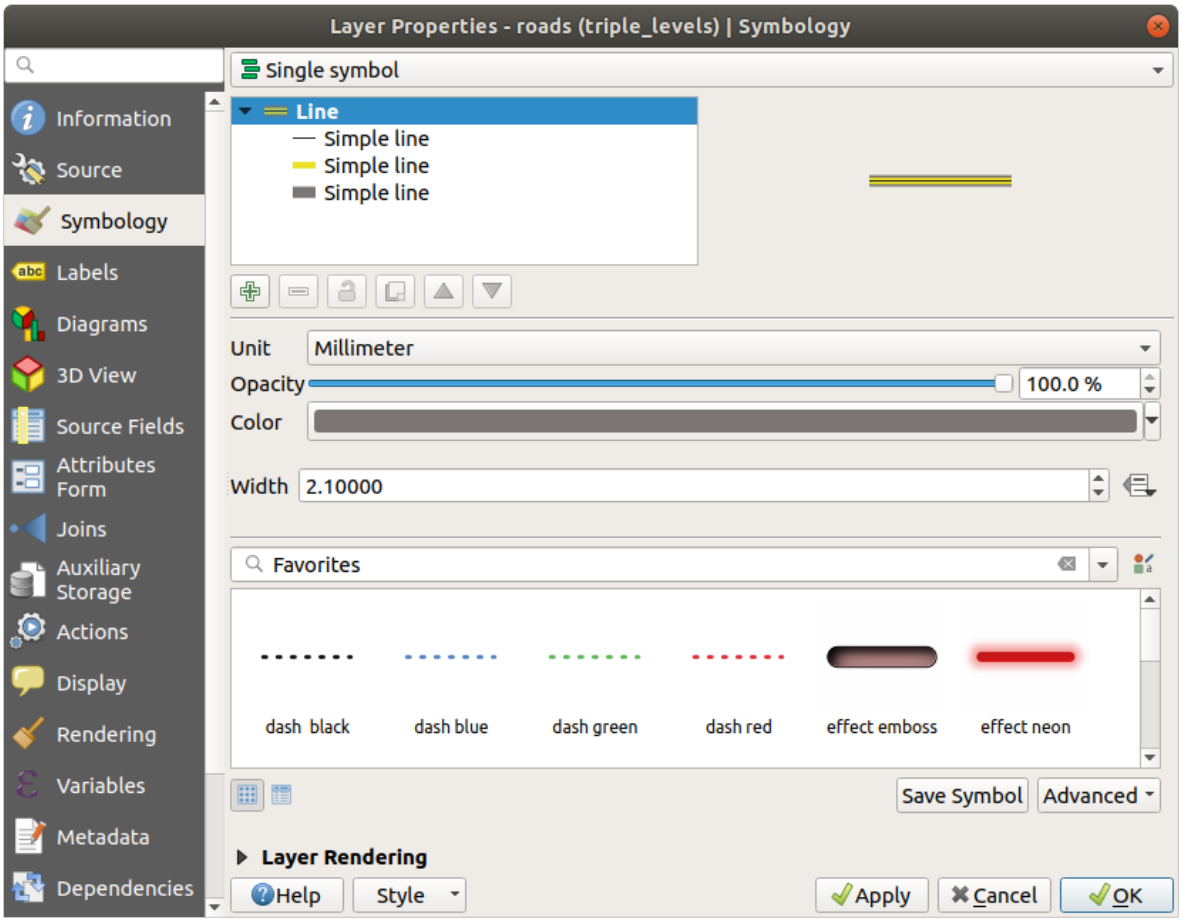
2.4.9 Try Yourself

Change the appearance of the *roads* layer again.

Make the roads narrow and yellow, with a thin, pale gray outline and a thin black line in the middle. Remember that you may need to change the layer rendering order via the *Advanced* ► *Symbol levels...* dialog.

Answer

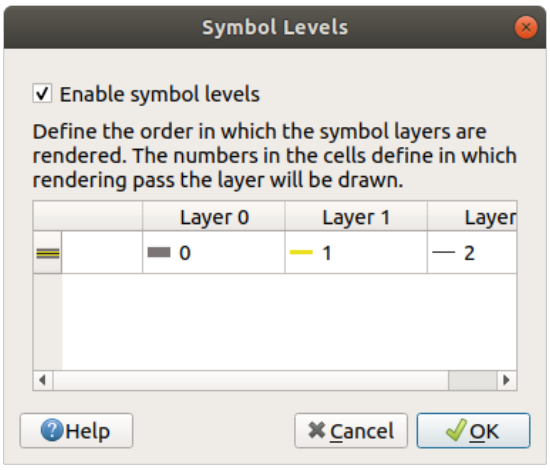
To make the required symbol, you need three symbol layers:



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:



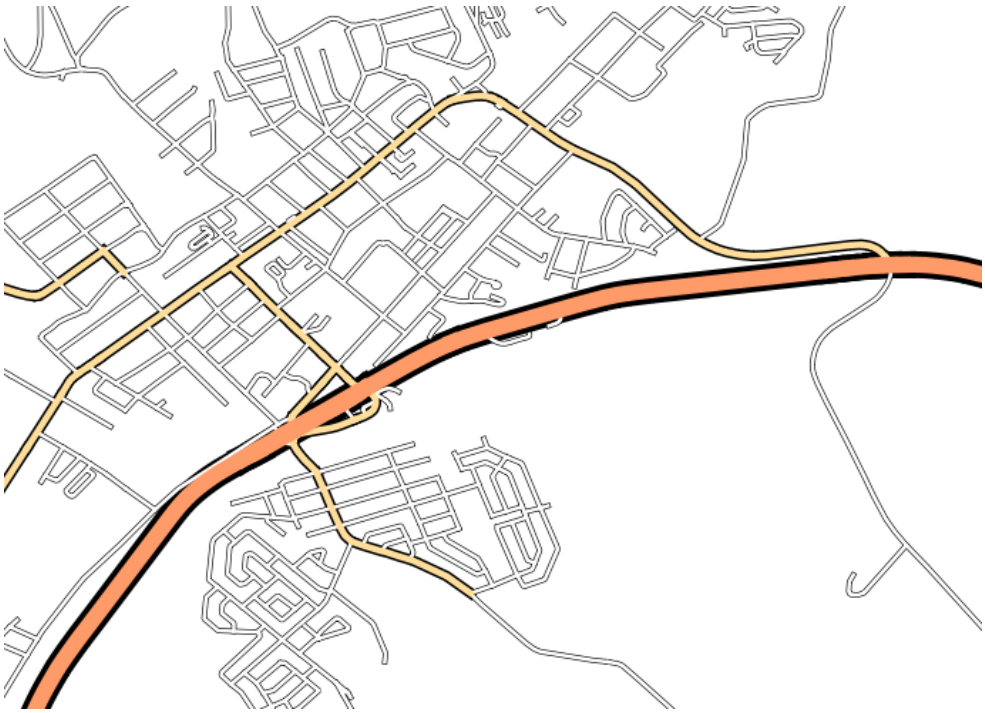
2. Now your map should look like this:

2.4.10 Try Yourself

Symbol levels also work for classified layers (i.e., layers having multiple symbols). Since we haven't covered classification yet, you will work with some rudimentary pre-classified data.

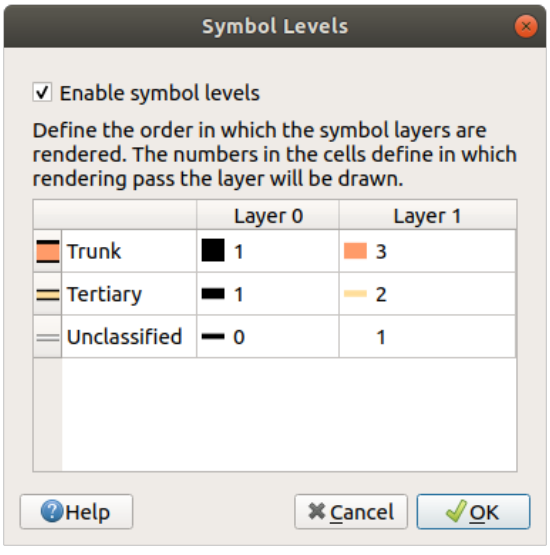
1. Create a new project and add only the *roads* dataset.
2. Apply the style file `advanced_levels_demo.qml` provided in `exercise_data/styles` to the layer. This can be done through the *Style* ► *Load Style...* combobox at the bottom of the *Layer Properties* dialog.

3. Zoom in to the Swellendam area.
4. Using symbol layers, ensure that the outlines of layers flow into one another as per the image below:



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.

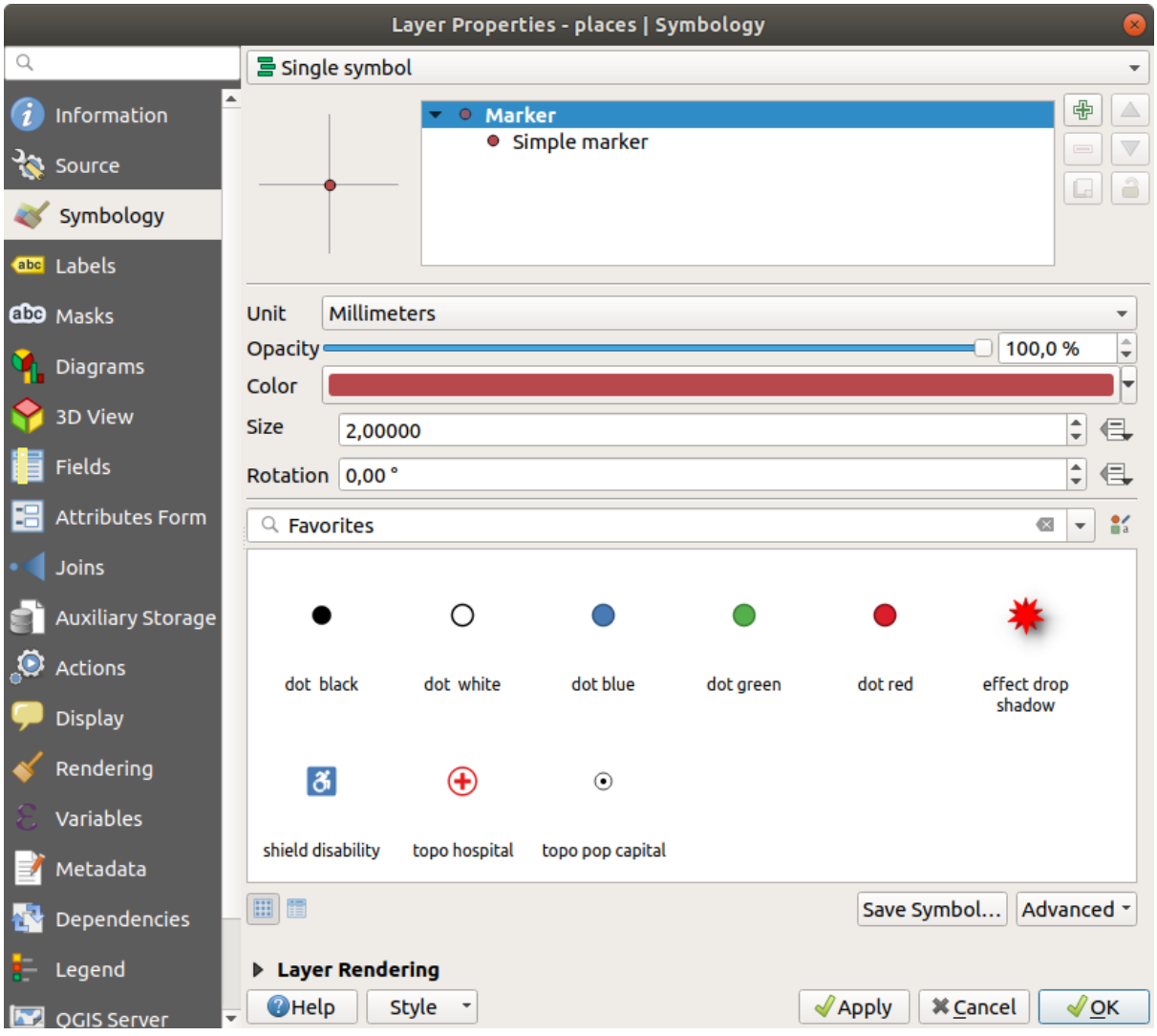
2.4.11 Follow Along: Symbol layer types

In addition to setting fill colors and using predefined patterns, you can use different symbol layer types entirely. The only type we've been using up to now was the *Simple Fill* type. The more advanced symbol layer types allow you to customize your symbols even further.

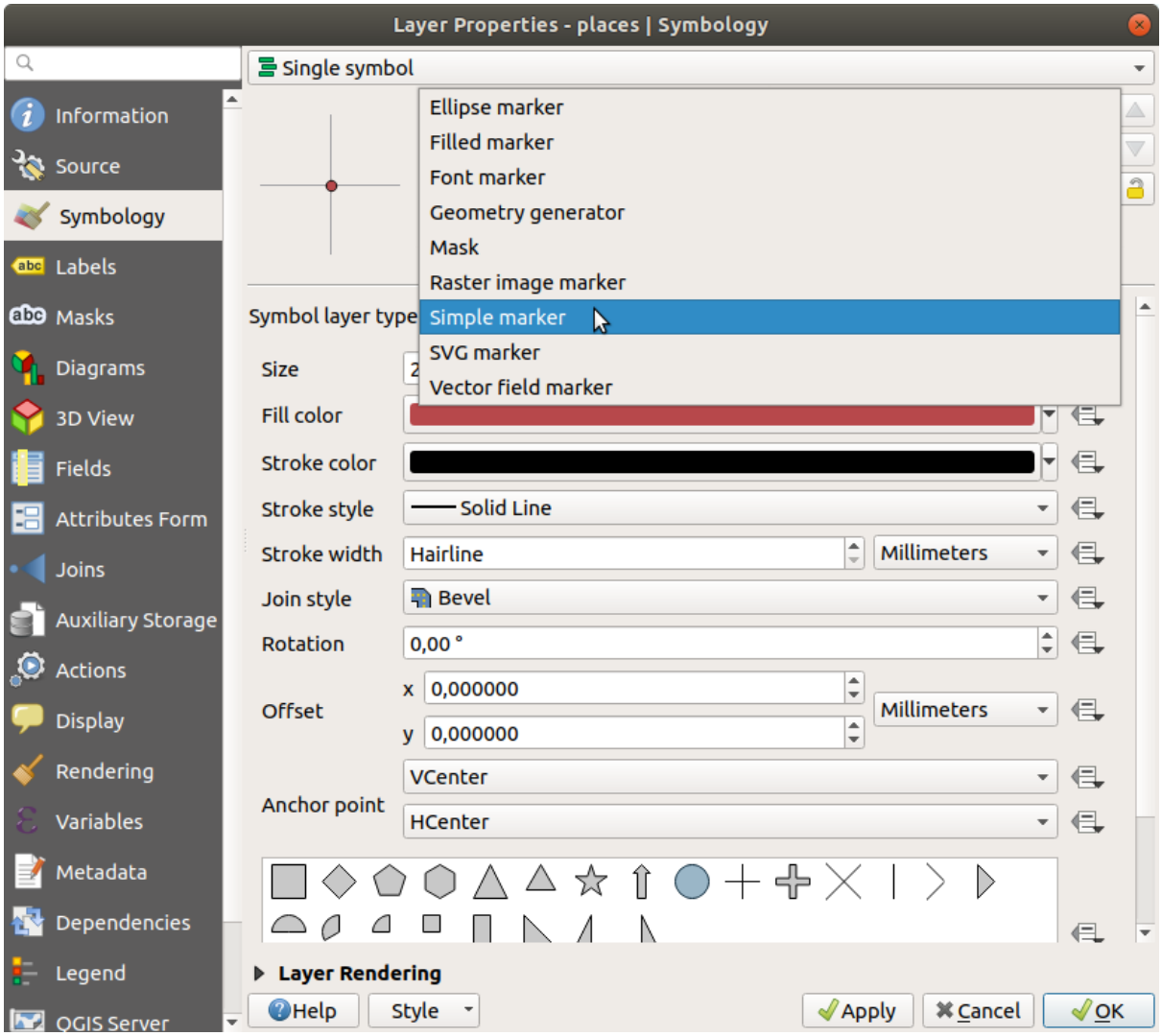
Each type of vector (point, line and polygon) has its own set of symbol layer types. First we will look at the types available for points.

Point Symbol Layer Types

1. Uncheck all the layers except for *places*.
2. Change the symbol properties for the *places* layer:



3. You can access the various symbol layer types by selecting the *Simple marker* layer in the symbol layers tree, then click the *Symbol layer type* dropdown:

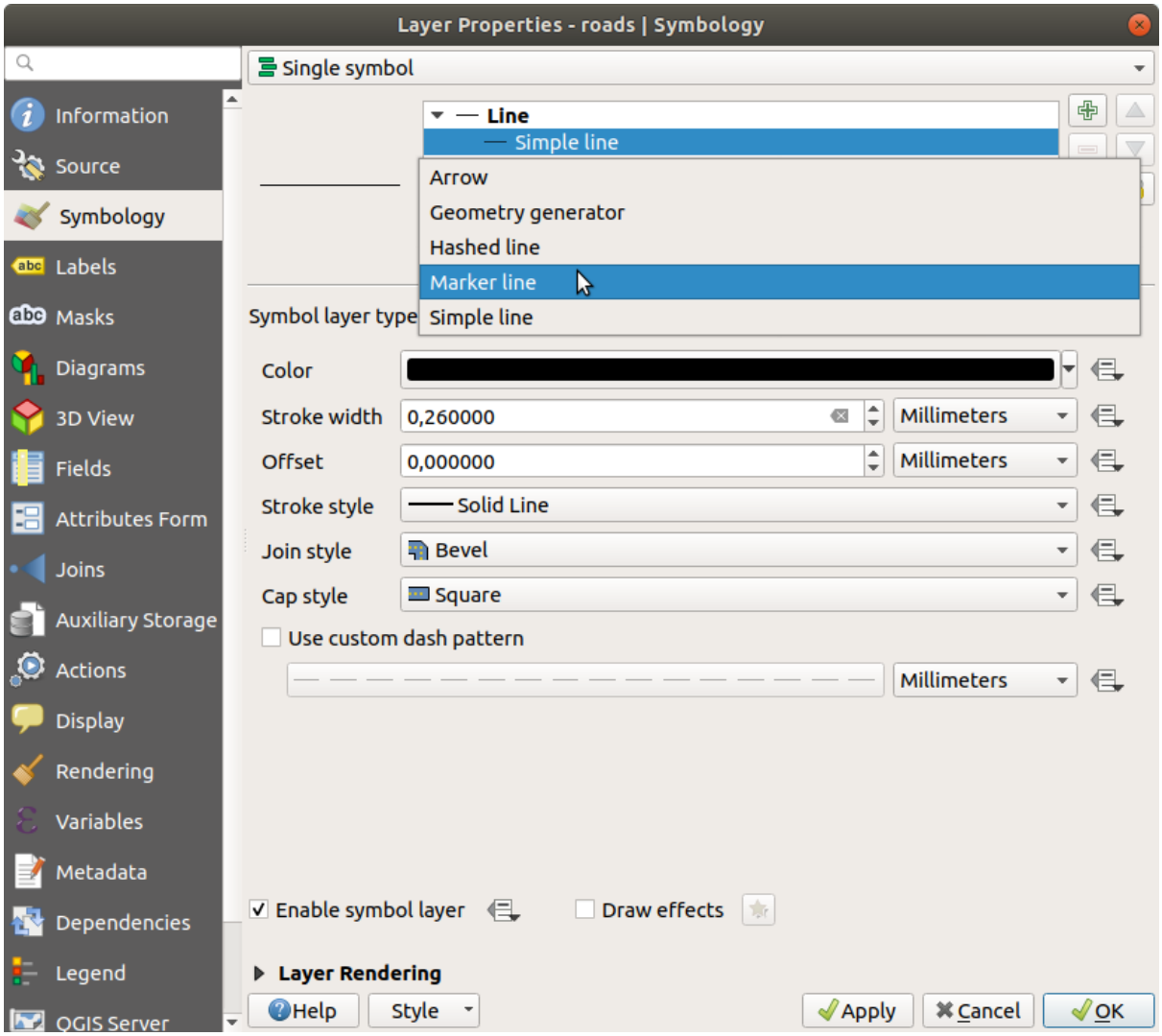


4. Investigate the various options available to you, and choose a symbol with styling you think is appropriate.
5. If in doubt, use a round *Simple marker* with a white border and pale green fill, with a *Size* of `3.00` and a *Stroke width* of `0.5`.

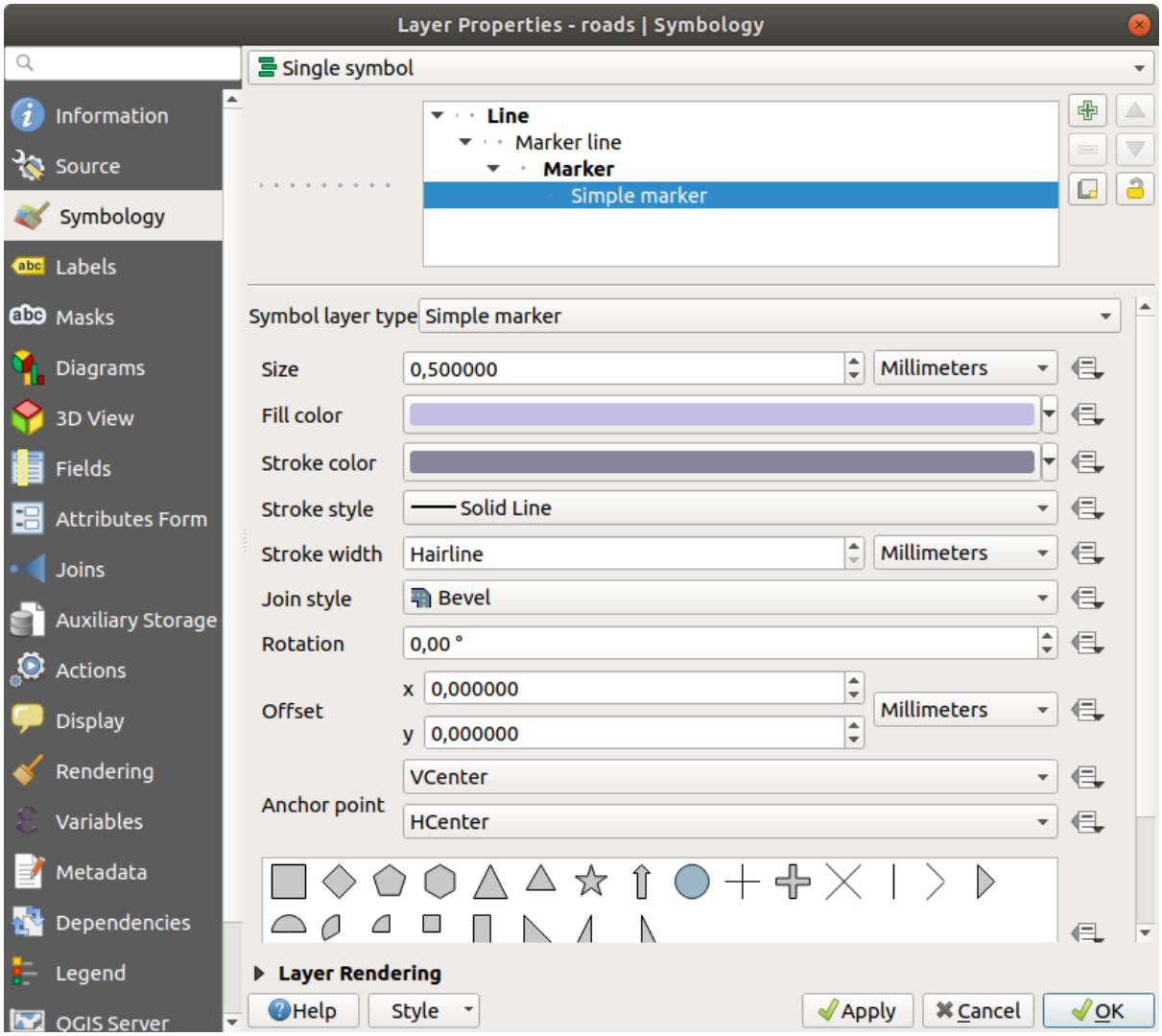
Line Symbol Layer Types

To see the various options available for line data:

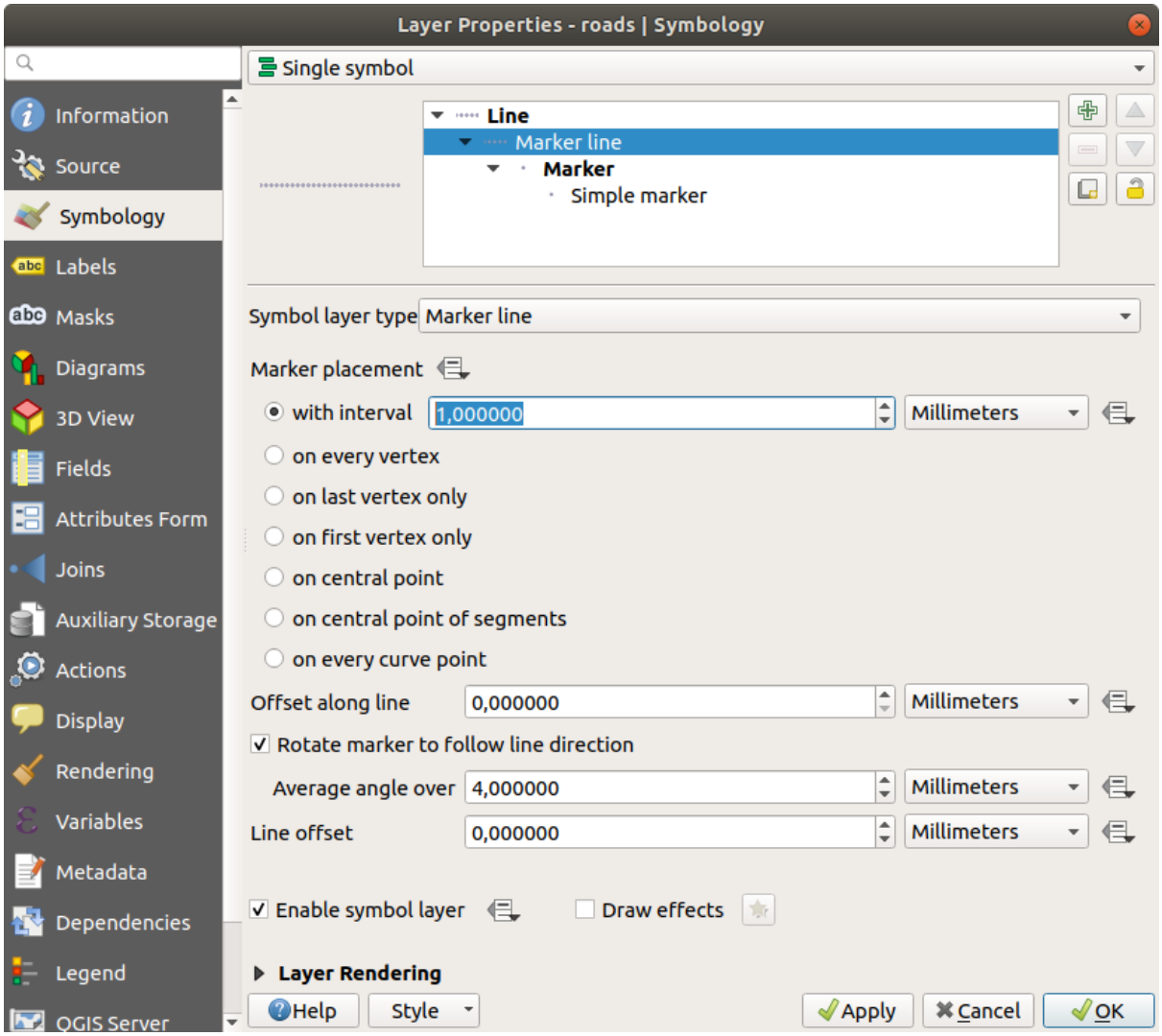
1. Change the *Symbol layer type* for the *roads* layer's topmost symbol layer to *Marker line*:



2. Select the *Simple marker* layer in the symbol layers tree. Change the symbol properties to match this dialog:



3. Select the *Marker line* layer and change the interval to `1.00`:



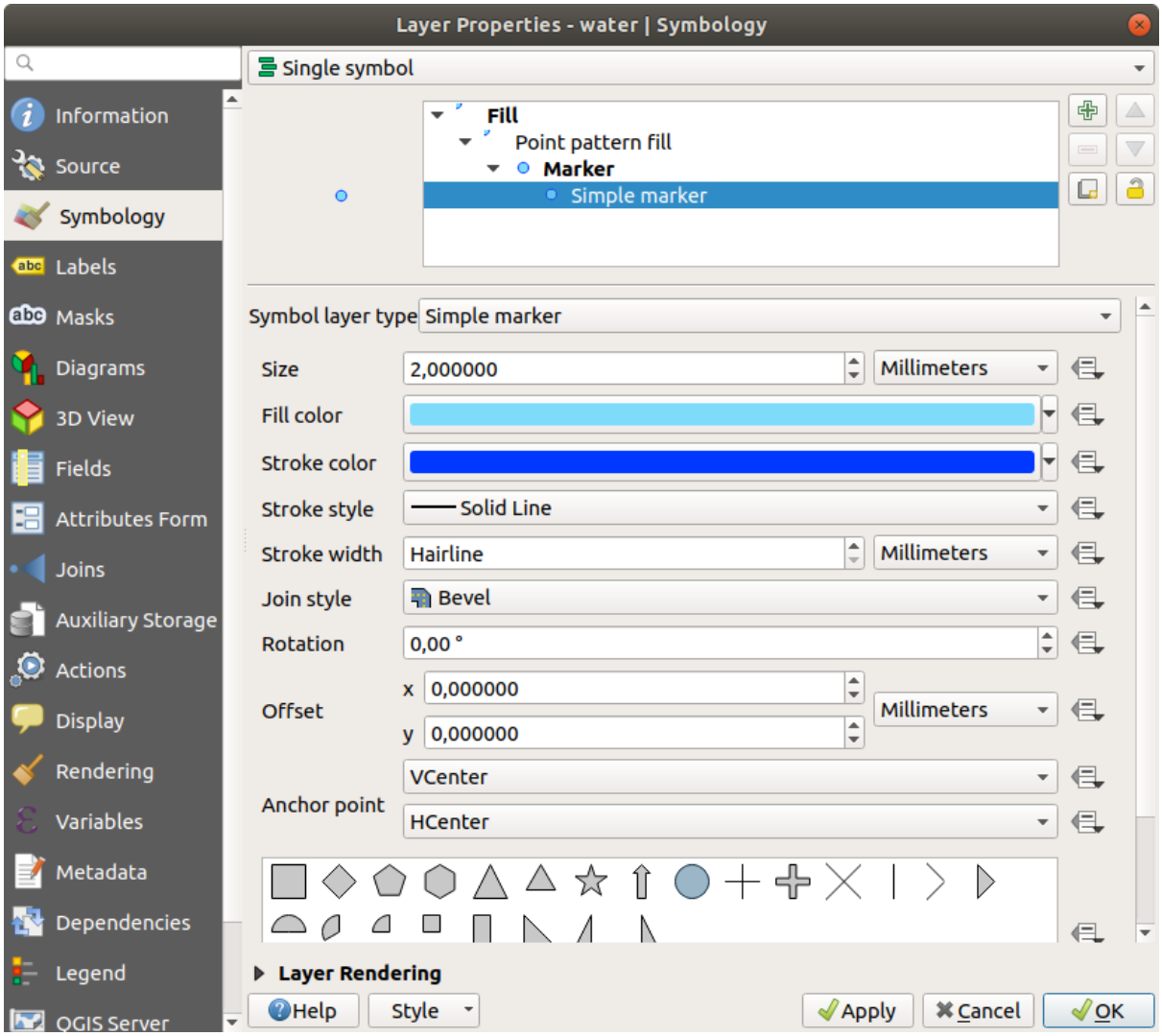
4. Ensure that the symbol levels are correct (via the *Advanced* ► *Symbol levels* dialog we used earlier) before applying the style.

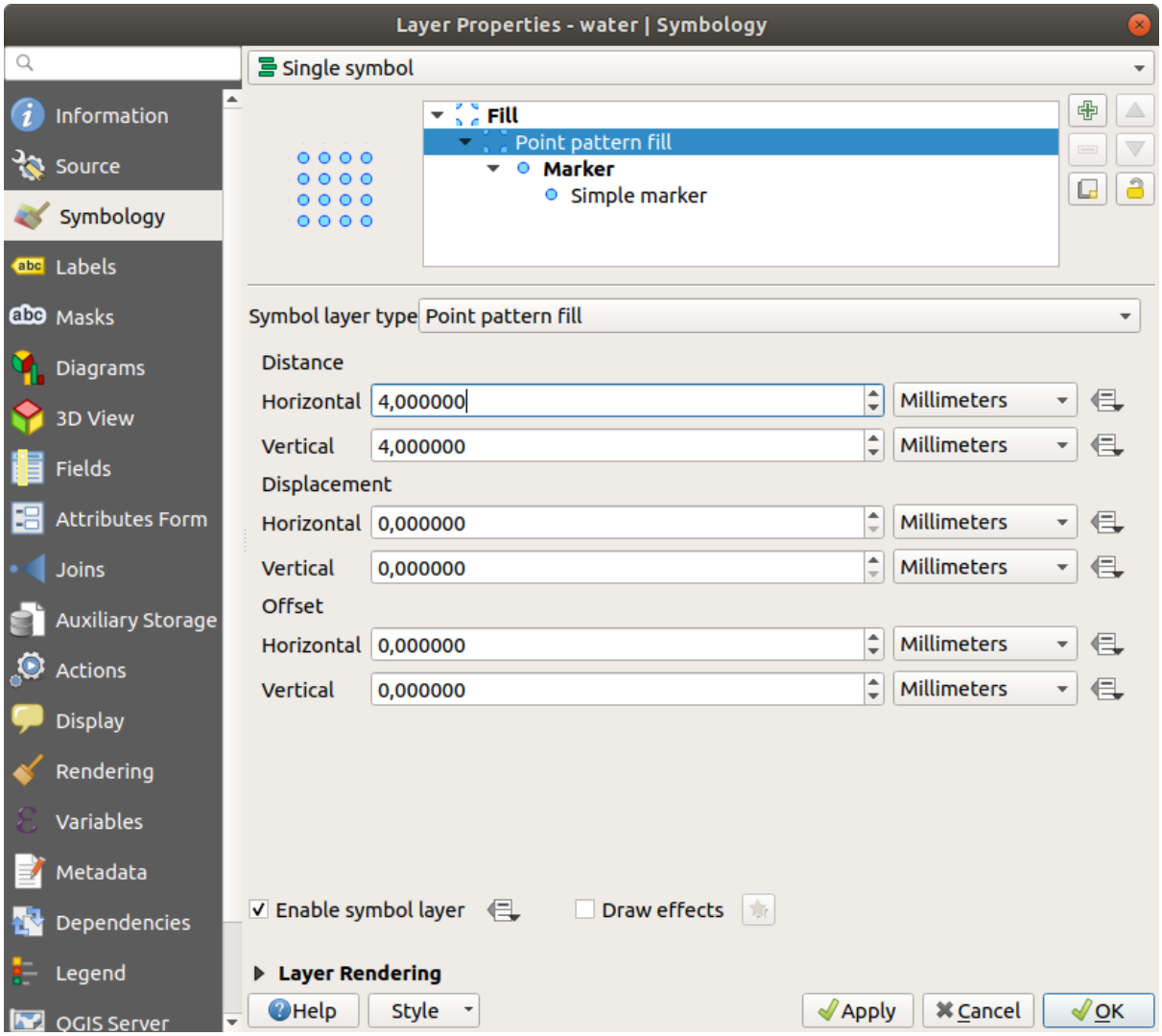
Once you have applied the style, take a look at its results on the map. As you can see, these symbols change direction along with the road but don't always bend along with it. This is useful for some purposes, but not for others. If you prefer, you can change the symbol layer in question back to the way it was before.

Polygon Symbol Layer Types

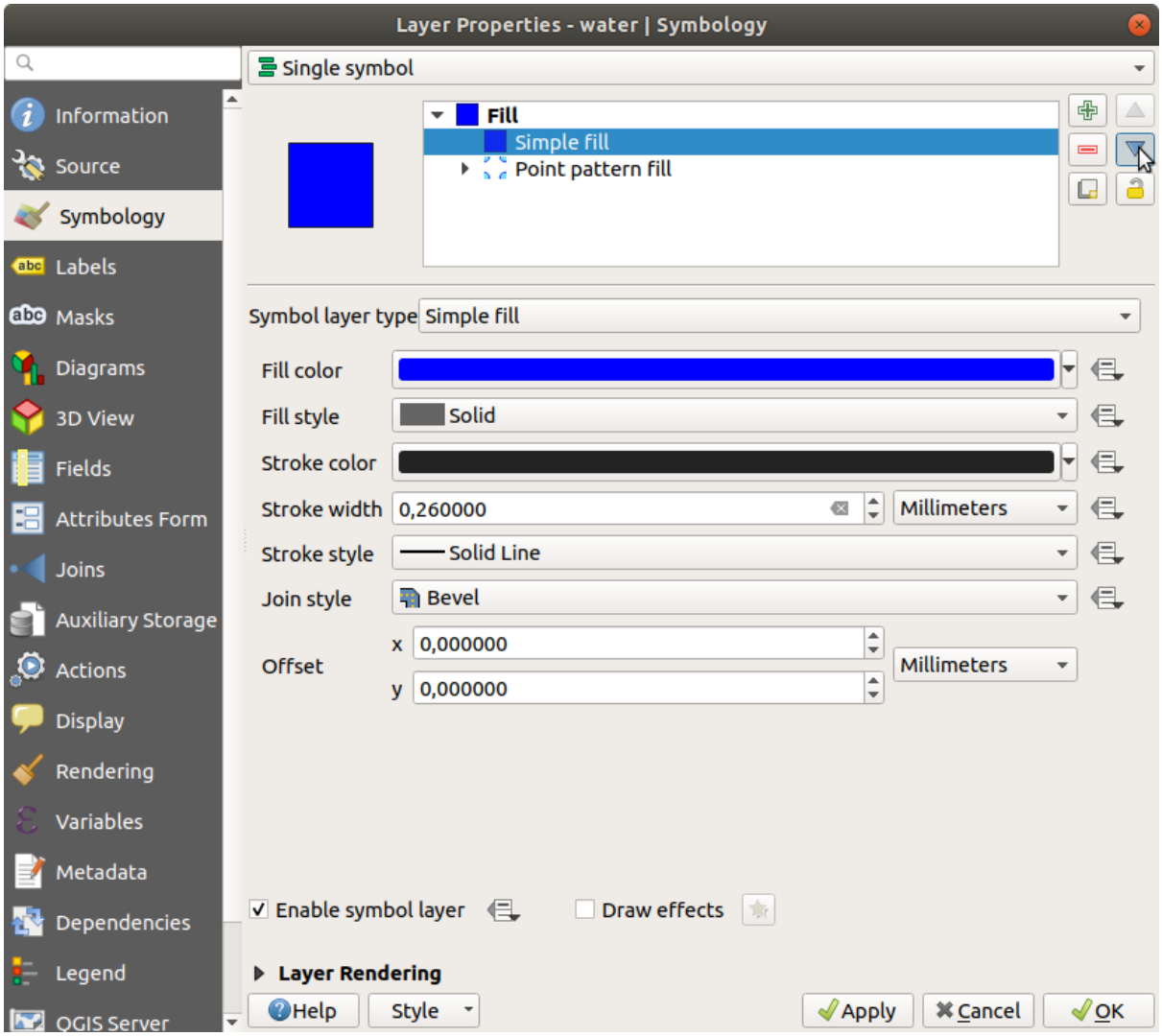
To see the various options available for polygon data:

1. Change the *Symbol layer type* for the *water* layer, as before for the other layers.
2. Investigate what the different options on the list can do.
3. Choose one of them that you find suitable.
4. If in doubt, use the *Point pattern fill* with the following options:


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



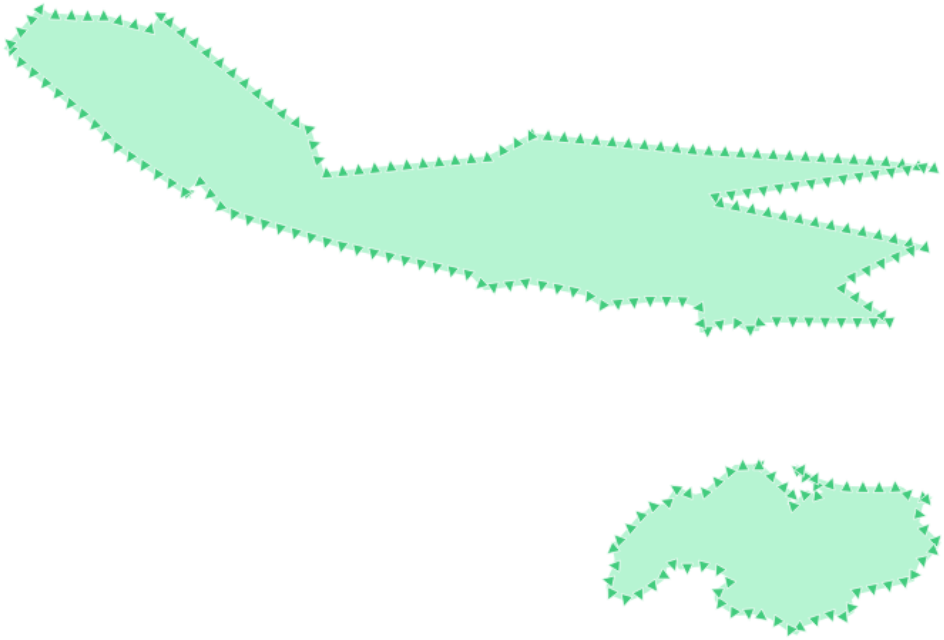
5. Add a new symbol layer with a normal *Simple fill*.
6. Make it the same light blue with a darker blue border.
7. Move it underneath the point pattern symbol layer with the *Move down* button:



As a result, you have a textured symbol for the water layer, with the added benefit that you can change the size, shape and distance of the individual dots that make up the texture.

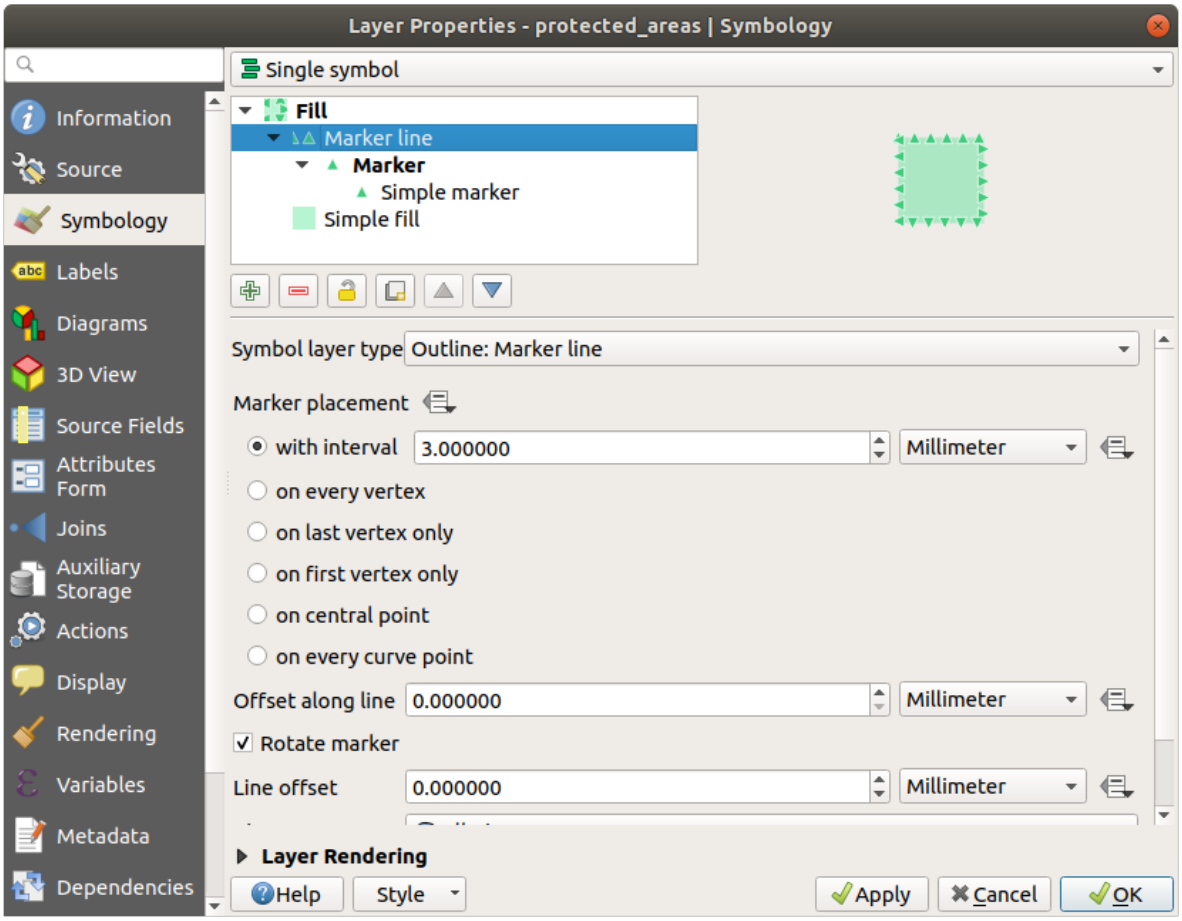
2.4.12 Try Yourself

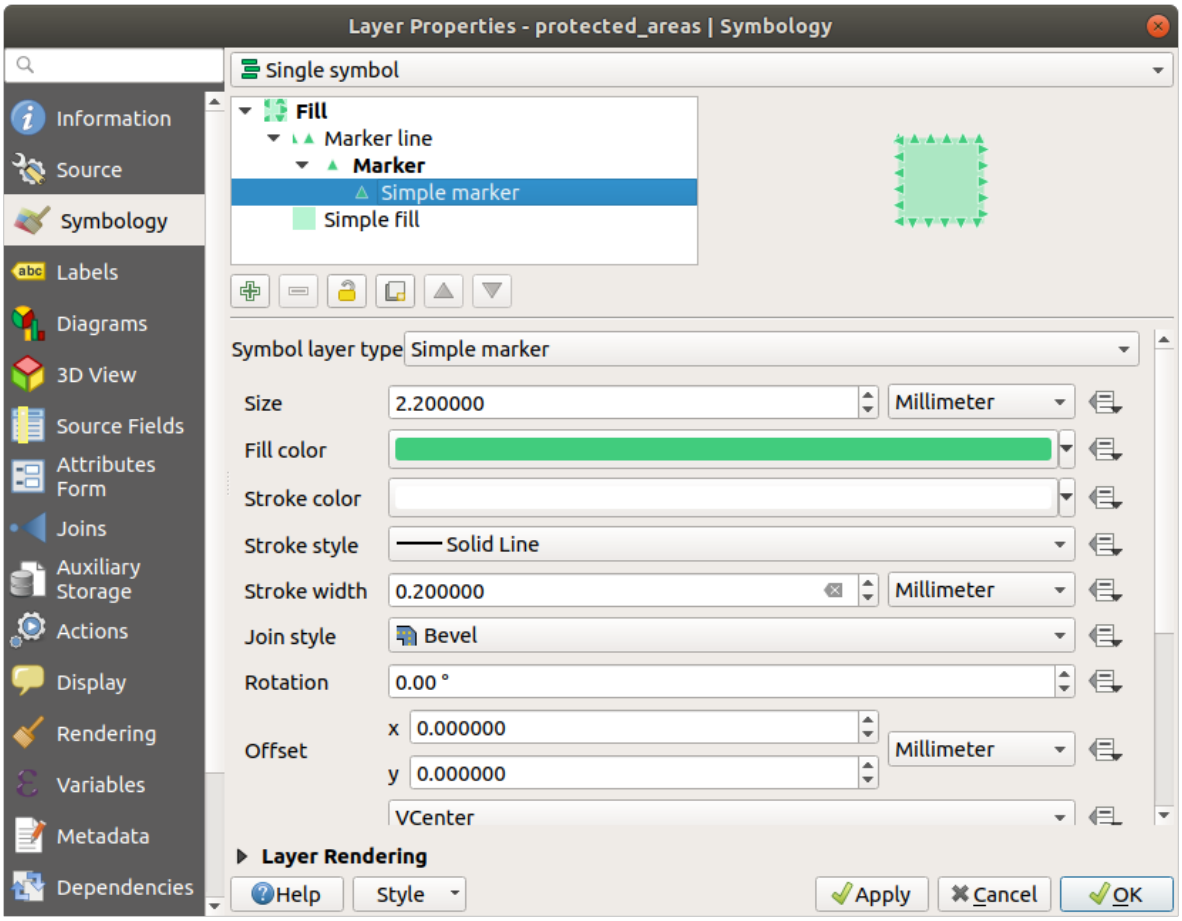
Apply a green transparent fill color to the *protected_areas* layer, and change the outline to look like this:



Answer

Here are examples of the symbol structure:





2.4.13 Follow Along: Geometry generator symbology

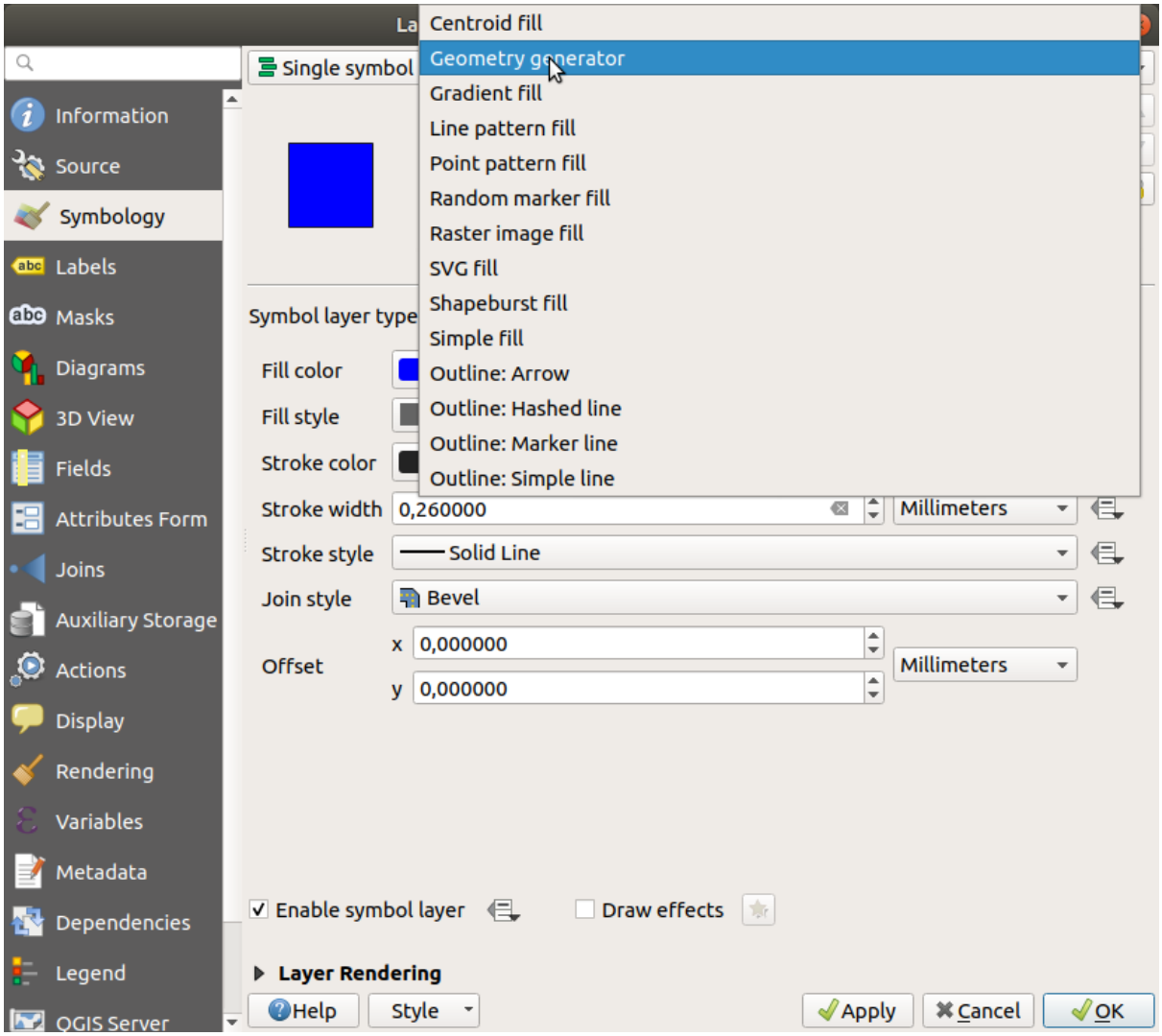
You can use the Geometry generator symbology with all layer types (points, lines and polygons). The resulting symbol depends directly on the layer type.

Very briefly, the Geometry generator symbology allows you to run some spatial operations within the symbology itself. For example you can run a real centroid spatial operation on a polygon layer without creating a point layer.

Moreover, you have all the styling options to change the appearance of the resulting symbol.

Let's give it a try!

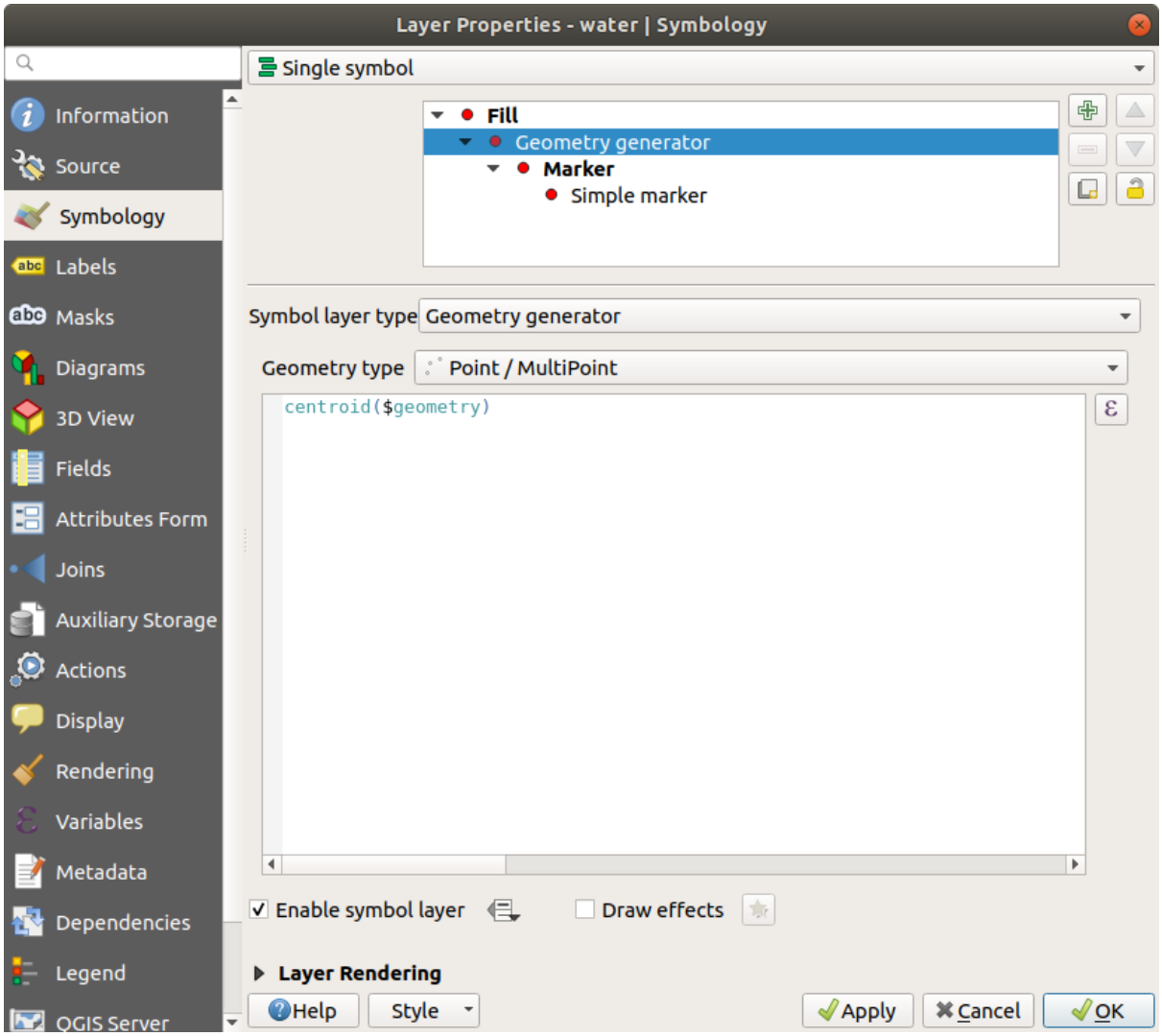
1. Select the *water* layer.
2. Click on *Simple fill* and change the *Symbol layer type* to *Geometry generator*.



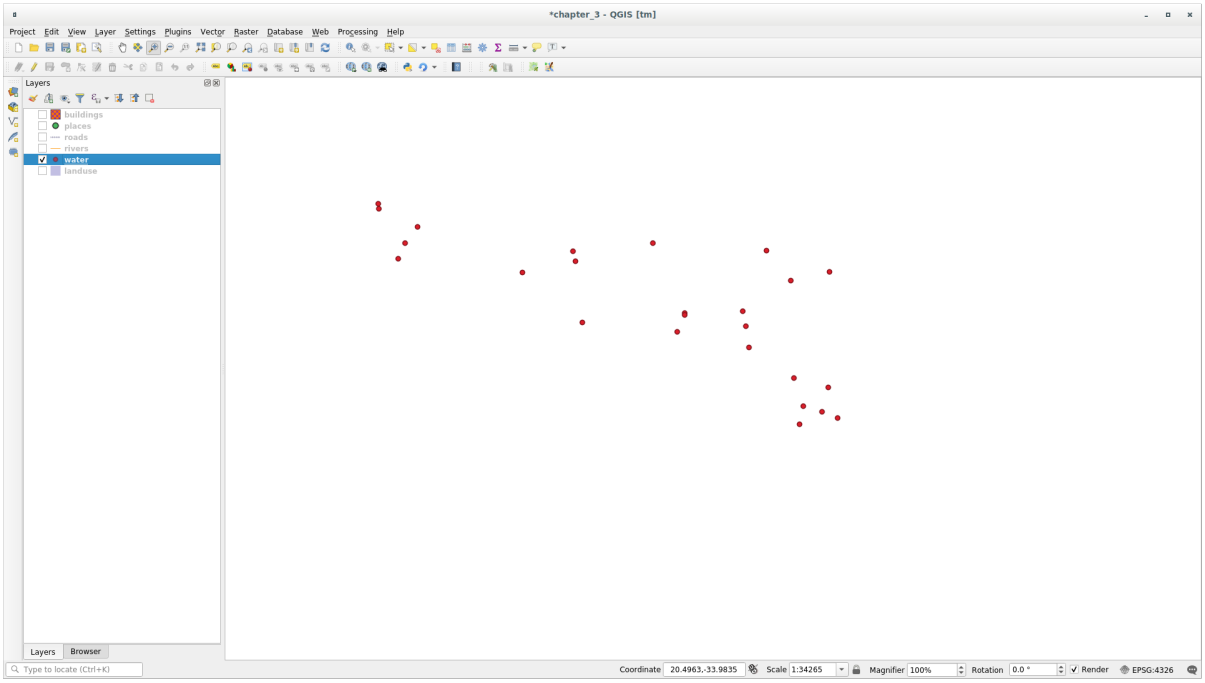
3. Before to start writing the spatial query we have to choose the Geometry Type in output. In this example we are going to create centroids for each feature, so change the Geometry Type to *Point / Multipoint*.

4. Now let's write the query in the query panel:

```
centroid($geometry)
```



5. When you click on *OK* you will see that the *water* layer is rendered as a point layer! We have just run a spatial operation within the layer symbology itself, isn't that amazing?



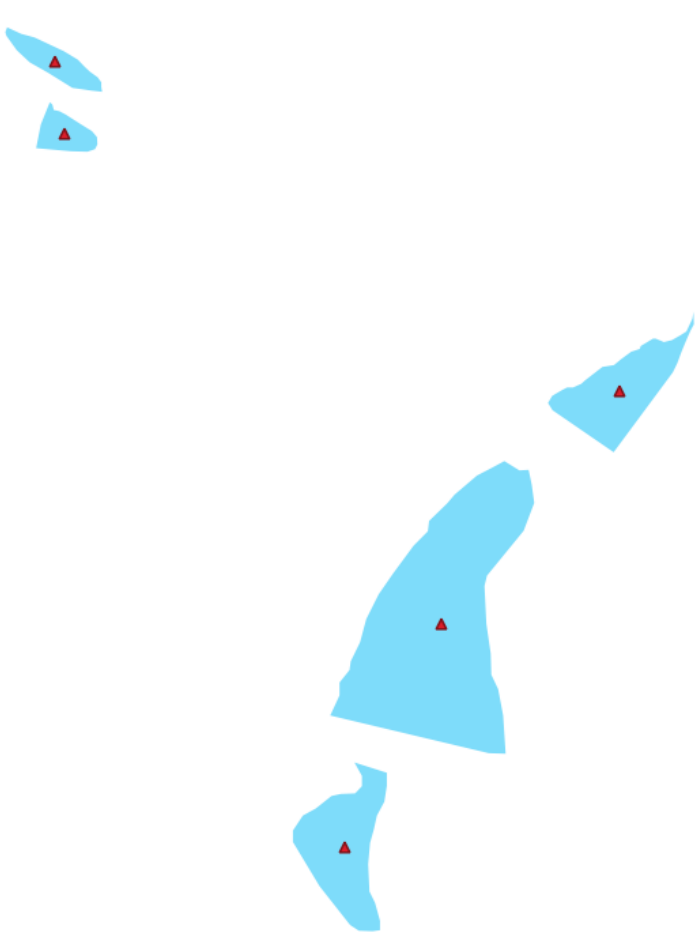
With the Geometry generator symbology you can really go over the edge of *normal* symbology.

Try Yourself

Geometry generator are just another symbol level. Try to add another *Simple fill* underneath the *Geometry generator* one.

Change also the appearance of the Simple marker of the Geometry generator symbology.

The final result should look like this:



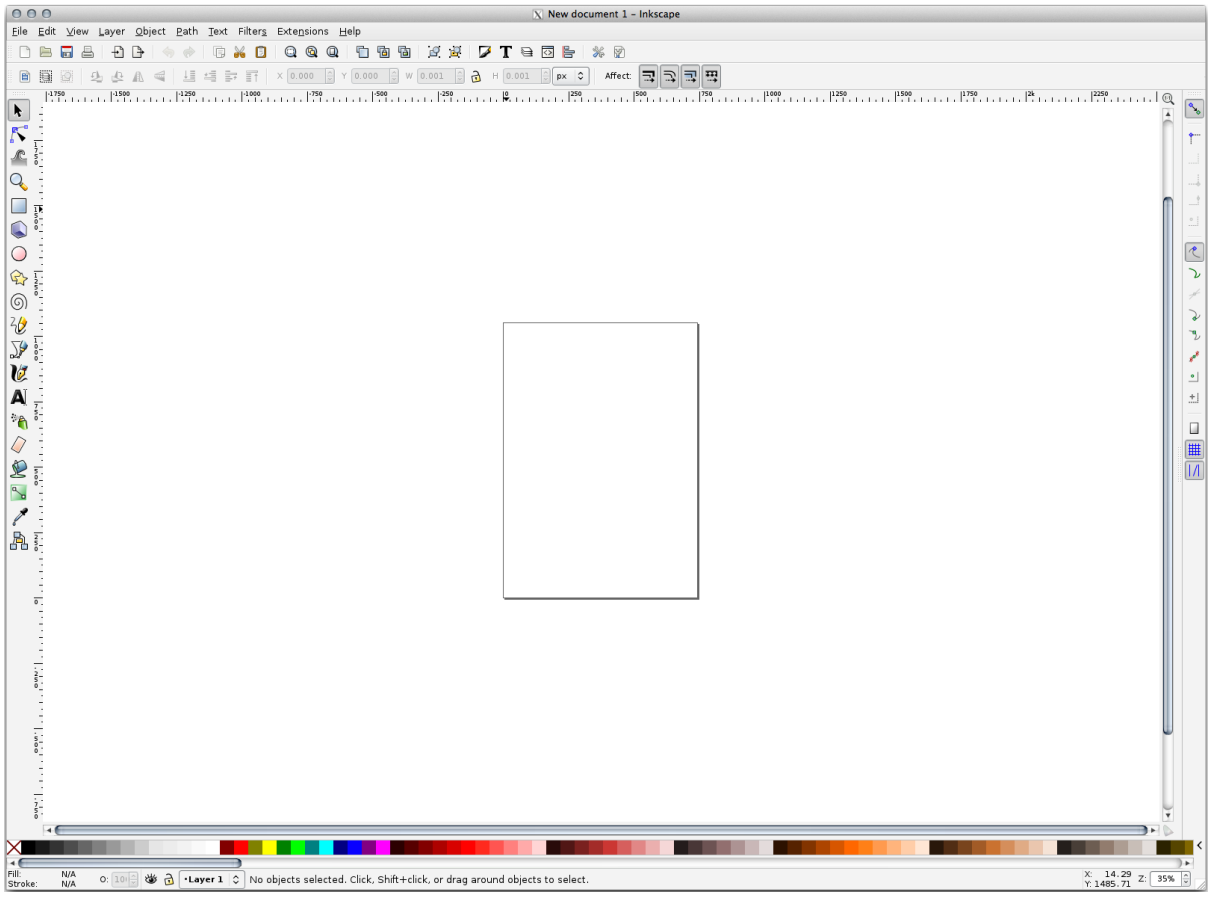
Answer

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 Follow Along: Creating a Custom SVG Fill

Bemerkung: To do this exercise, you will need to have the free vector editing software Inkscape installed.

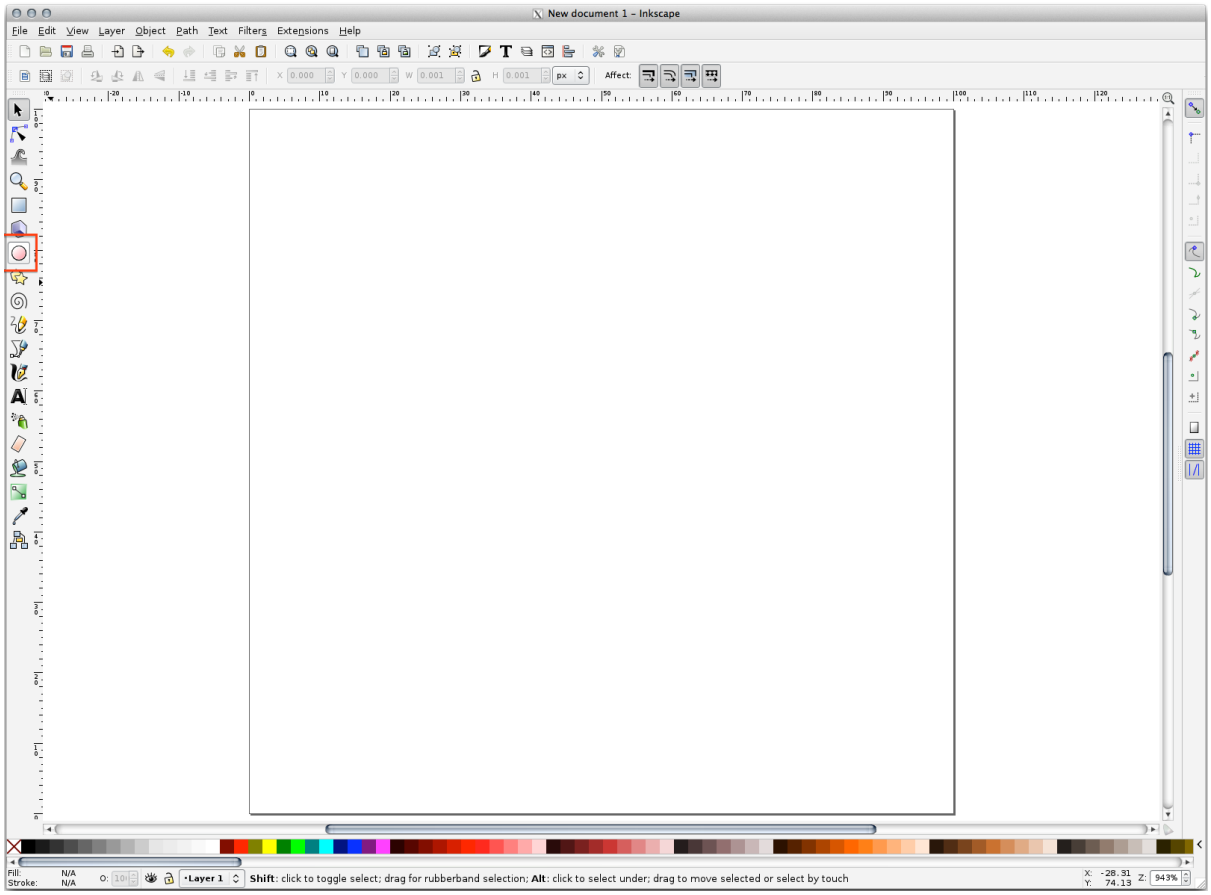
1. Start the Inkscape program. You will see the following interface:



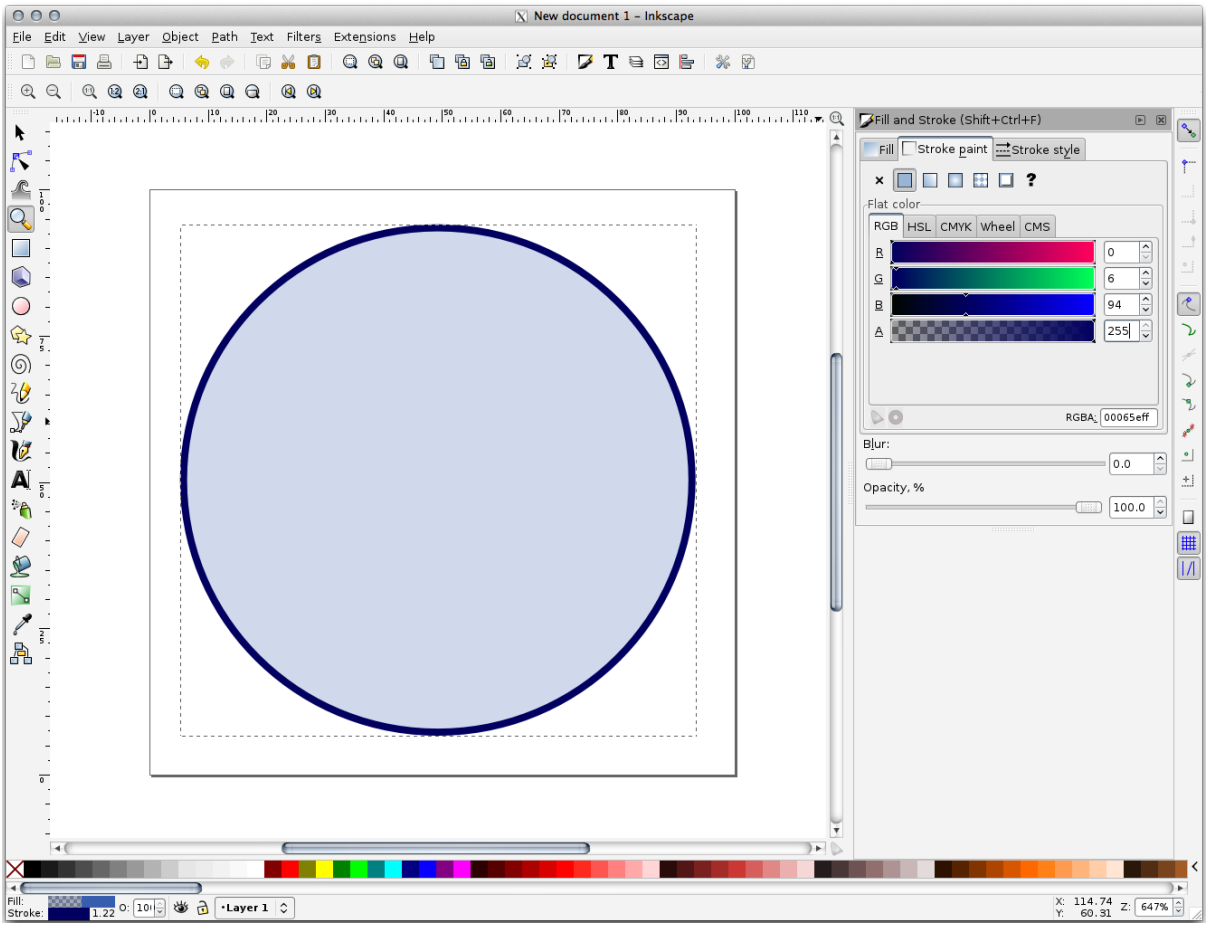
You should find this familiar if you have used other vector image editing programs, like Corel.

First, we'll change the canvas to a size appropriate for a small texture.

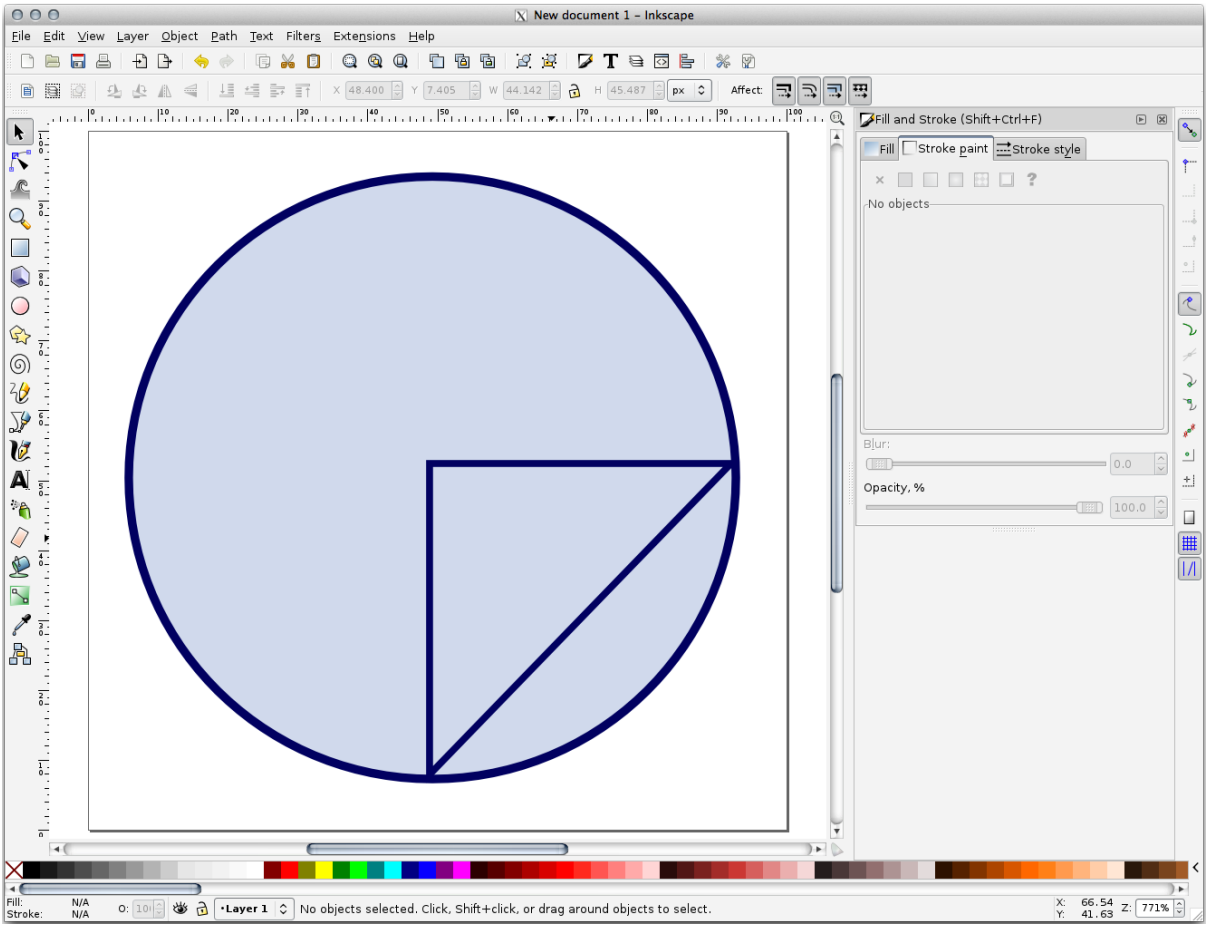
2. Click on the menu item *File* ► *Document Properties*. This will give you the *Document Properties* dialog.
3. Change the *Units* to *px*.
4. Change the *Width* and *Height* to `100`.
5. Close the dialog when you are done.
6. Click on the menu item *View* ► *Zoom* ► *Page* to see the page you are working with.
7. Select the *Circle* tool:



8. Click and drag on the page to draw an ellipse. To make the ellipse turn into a circle, hold the `Ctrl` button while you're drawing it.
9. Right-click on the circle you just created and open its *Fill and Stroke* options. You can modify its rendering, such as:
    1. Change the *Fill* color to a somehow pale grey-blue,
    2. Assign to the border a darker color in *Stroke paint* tab,
    3. And reduce the border thickness under *Stroke style* tab.



10. Draw a line using the *Pencil* tool:
    1. Click once to start the line. Hold `Ctrl` to make it snap to increments of 15 degrees.
    2. Move the pointer horizontally and place a point with a simple click.
    3. Click and snap to the vertex of the line and trace a vertical line, ended by a simple click.
    4. Now join the two end vertices.
    5. Change the color and width of the triangle symbol to match the circle's stroke and move it around as necessary, so that you end up with a symbol like this one:

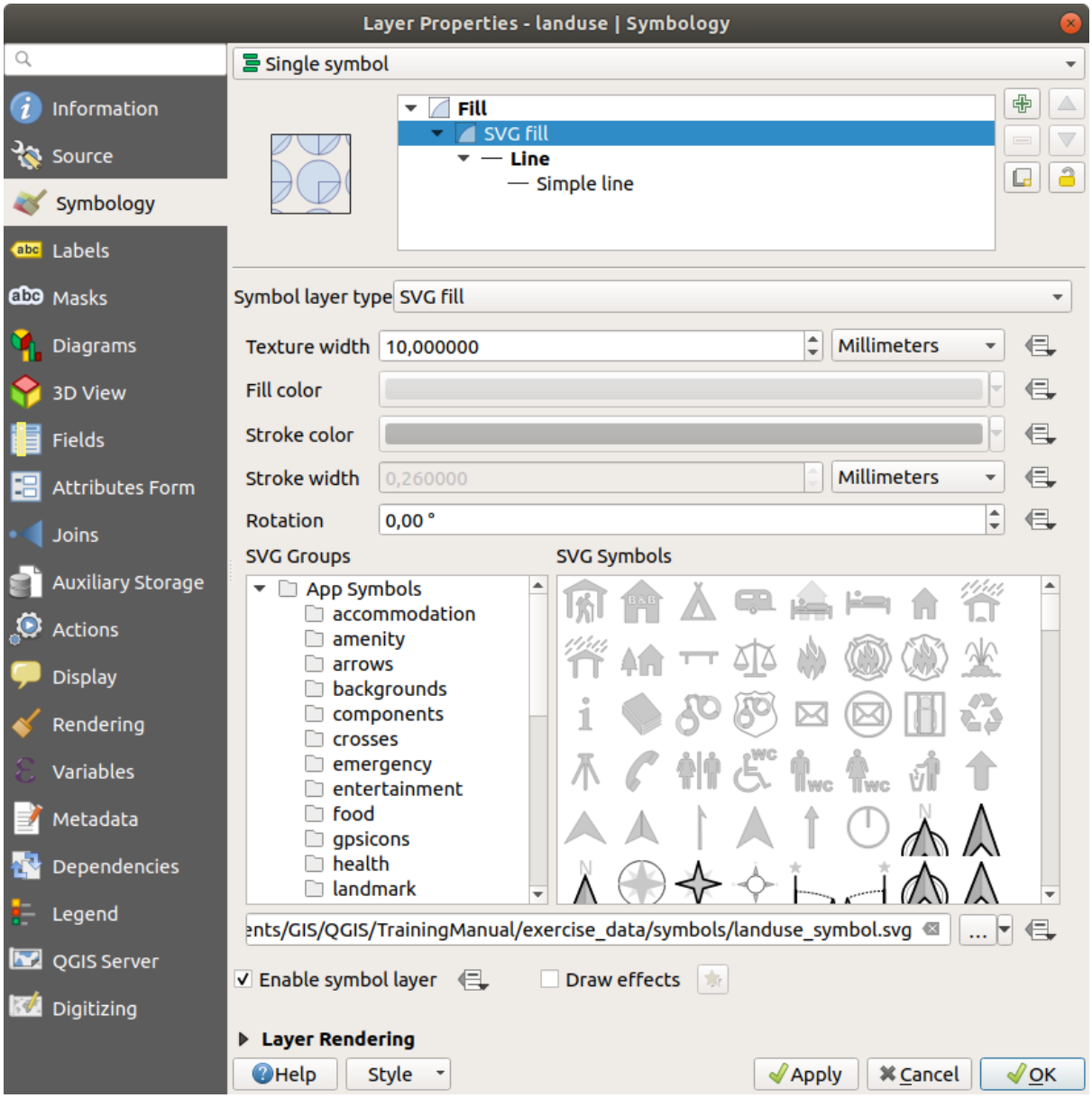


11. If the symbol you get satisfies you, then save it as *landuse_symbol* under the directory that the course is in, under `exercise_data/symbols`, as SVG file.

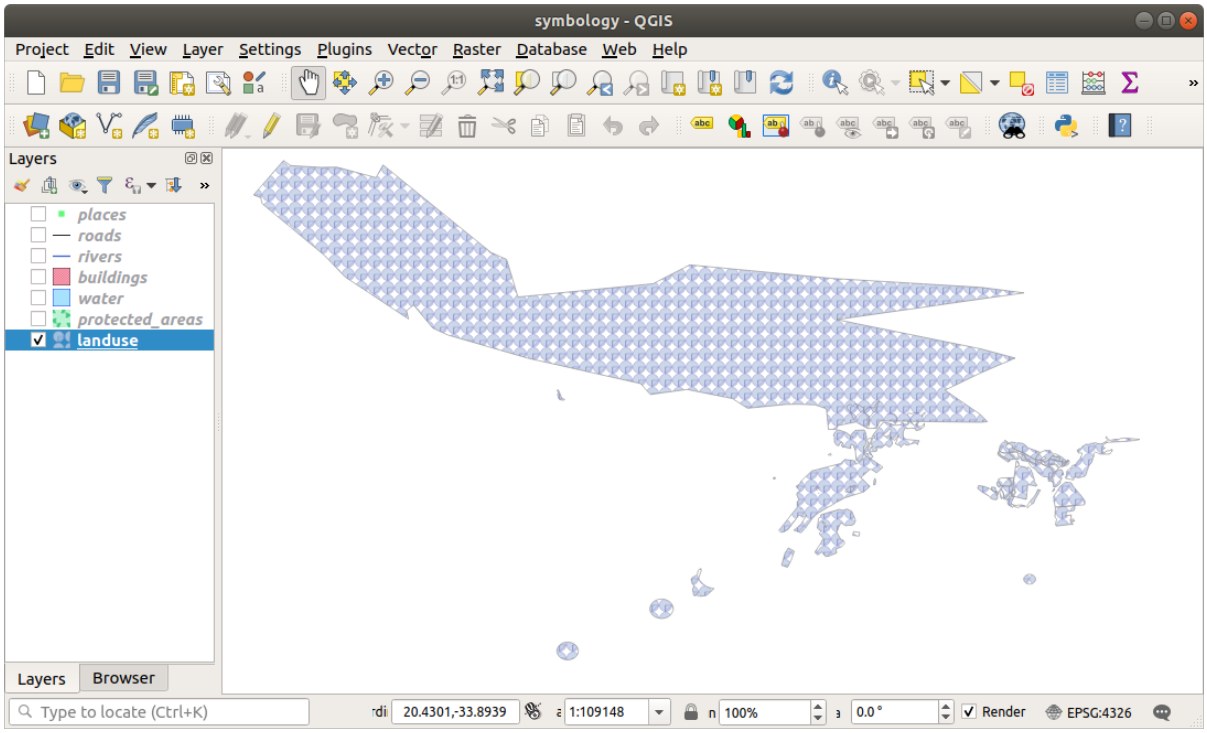
In QGIS:

1. Open the *Layer Properties* for the `landuse` layer.
2. In the *Symbology* tab, change the symbol structure by changing the *Symbol Layer Type* to *SVG Fill*``` shown below.
3. Click the … button and then *Select File…* to select your SVG image.

   It's added to the symbol tree and you can now customize its different characteristics (colors, angle, effects, units…).



Once you validate the dialog, features in *landuse* layer should now be covered by a set of symbols, showing a texture like the one on the following map. If textures are not visible, you may need to zoom in the map canvas or set in the layer properties a bigger *Texture width*.



2.4.15 In Conclusion

Changing the symbology for the different layers has transformed a collection of vector files into a legible map. Not only can you see what's happening, it's even nice to look at!

2.4.16 Further Reading

Examples of Beautiful Maps

2.4.17 What's Next?

Changing symbols for whole layers is useful, but the information contained within each layer is not yet available to someone reading these maps. What are the streets called? Which administrative regions do certain areas belong to? What are the relative surface areas of the farms? All of this information is still hidden. The next lesson will explain how to represent this data on your map.

Bemerkung: Did you remember to save your map recently?

KAPITEL 3

Module: Vektordaten klassifizieren

Das Klassifizieren von Vektordaten ermöglicht es Ihnen, den Elementen (verschiedene Objekte im selben Layer) in Abhängigkeit von ihren Attributen unterschiedliche Symbole zuzuordnen. Auf diese Weise kann jemand, der die Karte benutzt, die Attribute verschiedener Elemente leicht erkennen.

3.1 Lesson: Vector Attribute Data

Vector data is arguably the most common kind of data in the daily use of GIS. The vector model represents the location and shape of geographic features using points, lines and polygons (and for 3D data also surfaces and volumes), while their other properties are included as attributes (often presented as a table in QGIS).

Up to now, none of the changes we have made to the map have been influenced by the objects that are being shown. In other words, all the land use areas look alike, and all the roads look alike. When looking at the map, the viewers don't know anything about the roads they are seeing; only that there is a road of a certain shape in a certain area.

But the whole strength of GIS is that all the objects that are visible on the map also have attributes. Maps in a GIS aren't just pictures. They represent not only objects in locations, but also information about those objects.

The goal for this lesson: To learn about the structure of vector data and explore the attribute data of an object

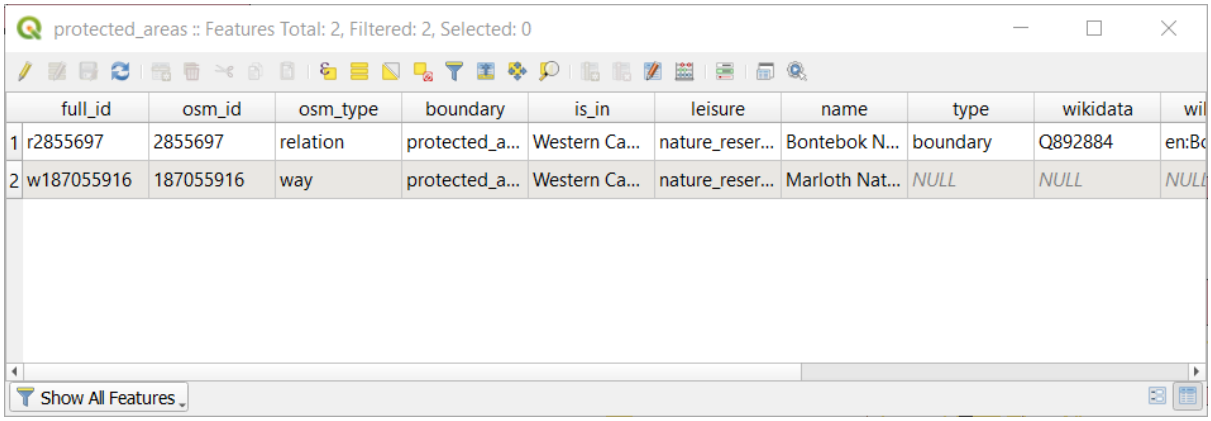
3.1.1 Follow Along: Viewing Layer Attributes

It's important to know that the data you will be working with does not only represent **where** objects are in space, but also tells you **what** those objects are.

From the previous exercise, you should have the `protected_areas` layer loaded in your map. If it is not loaded, then you can find the `protected_areas.shp` *ESRI Shapefile* format dataset in directory `exercise_data/shapefile`.

The polygons representing the protected areas constitute the **spatial data**, but we can learn more about the protected areas by exploring the **attribute table**.

1. In the *Layers* panel, click on the `protected_areas` layer to select it.
2. In the *Layer* menu, click the Open Attribute Table button (also accessible from top toolbars buttons). This will open a new window showing the attribute table of the `protected_areas` layer.



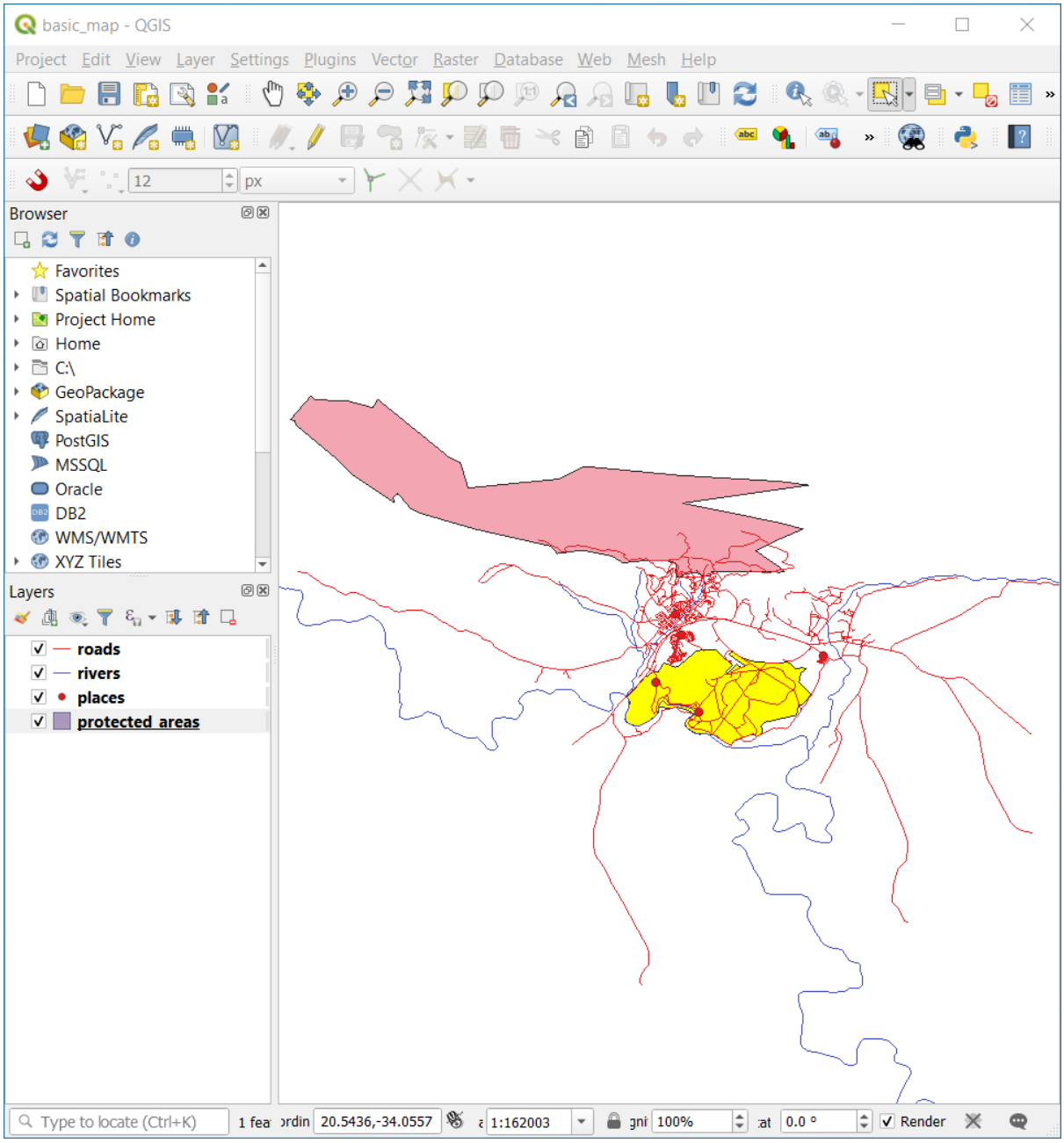
A row is called a **record** and is associated with a **feature** in the Canvas Map, such as a polygon. A column is called a **field** (or an **attribute**), and has a name that helps describe it, such as `name` or `id`. Values in the cells are known as **attribute values**. These definitions are commonly used in GIS, so it is good to become familiar with them.

In the `protected_areas` layer, there are two **features**, which are represented by the two polygons we see on the Map Canvas.

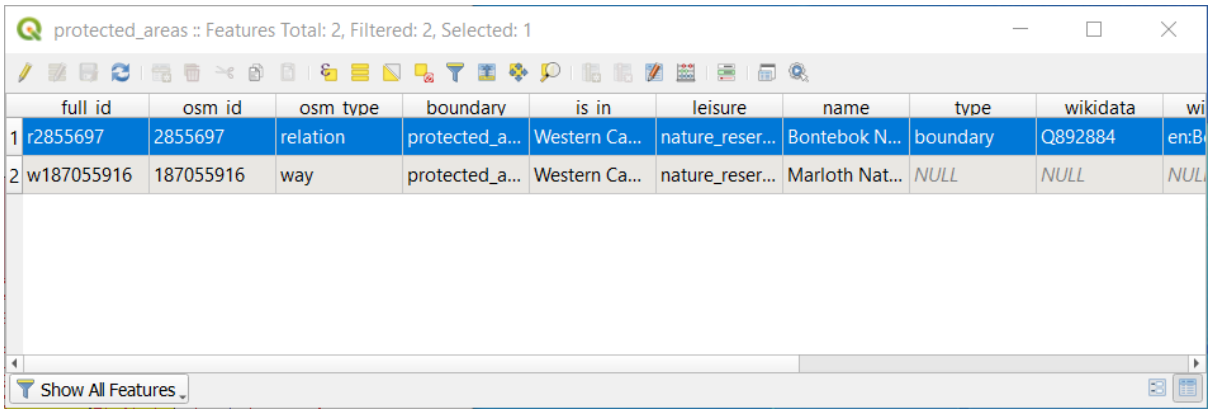
Bemerkung: In order to understand what the **fields** and **attribute values** represent, one may need to find documentation (or metadata) describing the meaning of the attribute values. This is usually available from the creator of the data set.

Next, let's see how a record in the attribute table is linked to a polygon feature that we see on the Map Canvas.

1. Go back to the main QGIS window.
2. In the *Edit –> Select* menu, click on the *Select Feature(s)* button.
3. Make sure the `protected_areas` layer is still selected in the *Layers* panel.
4. Move your mouse to the Map Canvas and left click on the smaller of the two polygons. The polygon will turn yellow indicating it is selected.

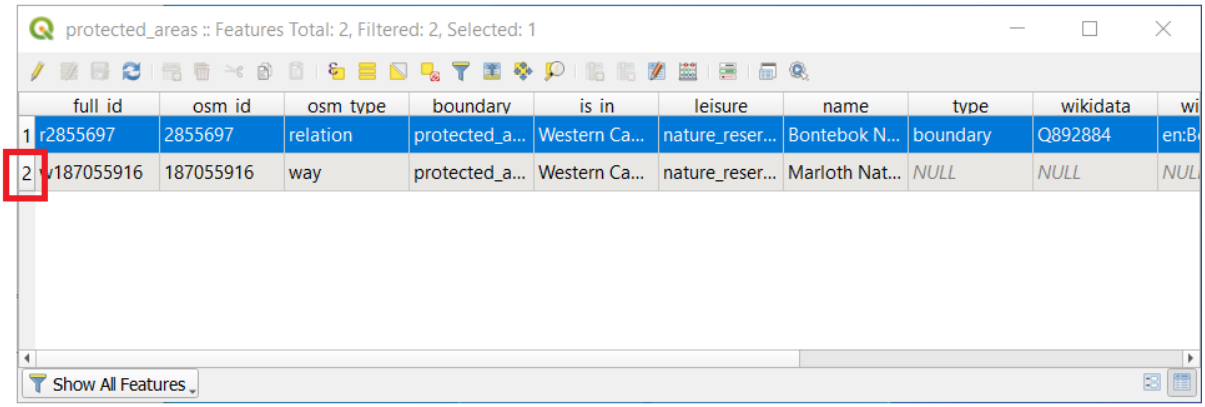


5. Go back to the *Attribute Table* window, and you should see a record (row) highlighted. These are the attribute values of the selected polygon.



You can also select a feature using the Attribute Table.

1. In the *Attribute Table* window, on the far left, click on the row number of the record that is currently not selected.



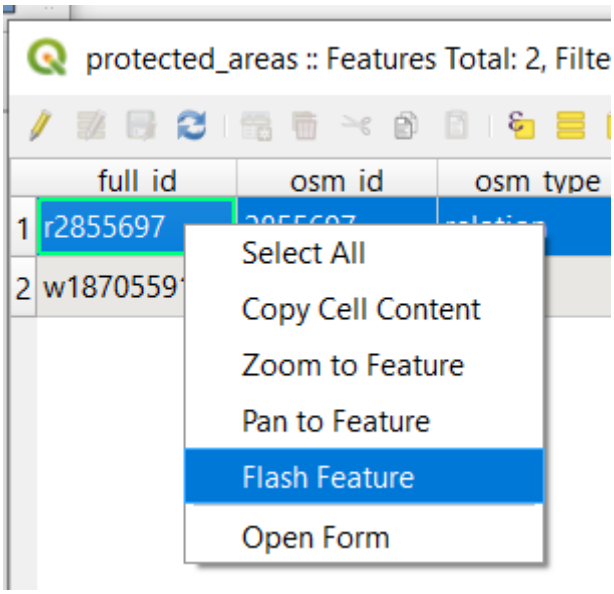
2. Go back to the main QGIS window and look at the Map Canvas. You should see the larger of the two polygons colored yellow.

3. To deselect the feature, go to the *Attribute Table* window and click on Deselect all features from the layer button.

Sometimes there are many features shown on the Map Canvas and it might be difficult to see which feature is selected from the Attribute Table. Another way to identify the location of a feature is to use the *Flash Feature* tool.

1. In the *Attribute Table*, right-click on any cell in the row that has the attribute value `r2855697` for the field `full_id`.

2. In the context menu, click on *Flash Feature* and watch the Map Canvas.

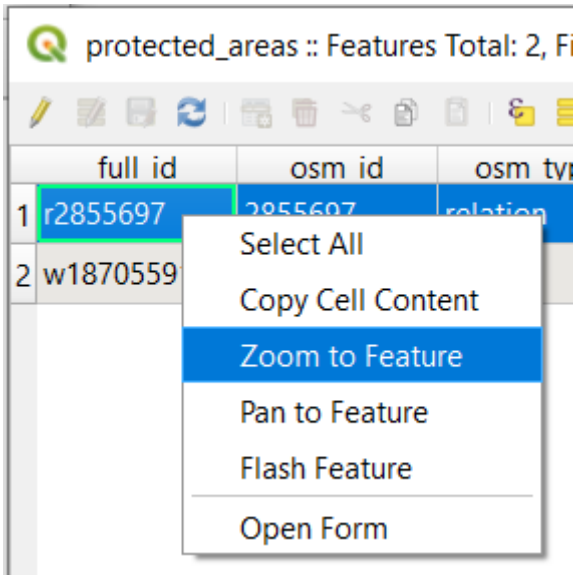


You should see the polygon flash red a few times. If you missed it, try it again.

Another useful tool is the *Zoom to Feature* tool, that tells QGIS to zoom to the feature of interest.

1. In the *Attribute Table*, right-click on any cell in the row that has the attribute value `r2855697` for the field `full_id`.

2. In the context menu, click on *Zoom to Feature*



Look at the Map Canvas. The polygon should now occupy the extent of the Map Canvas area.

You may now close the attribute table.

3.1.2 Try Yourself Exploring Vector Data Attributes

1. How many fields are available in the *rivers* layer?
2. Tell us a bit about the `town` places in your dataset.
3. Open the attribute table for the *places* layer. Which field would be the most useful to represent in label form, and why?

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* (it can be enabled from *View* ► *Toolbars* menu).
  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 In Conclusion

You now know how to use the attribute table to see what is actually in the data you're using. Any dataset will only be useful to you if it has the attributes that you care about. If you know which attributes you need, you can quickly decide if you're able to use a given dataset, or if you need to look for another one that has the required attribute data.

3.1.4 What's Next?

Different attributes are useful for different purposes. Some of them can be represented directly as text for the map user to see. You'll learn how to do this in the next lesson.

3.2 Lesson: Labels

Labels can be added to a map to show any information about an object. Any vector layer can have labels associated with it. These labels rely on the attribute data of a layer for their content.

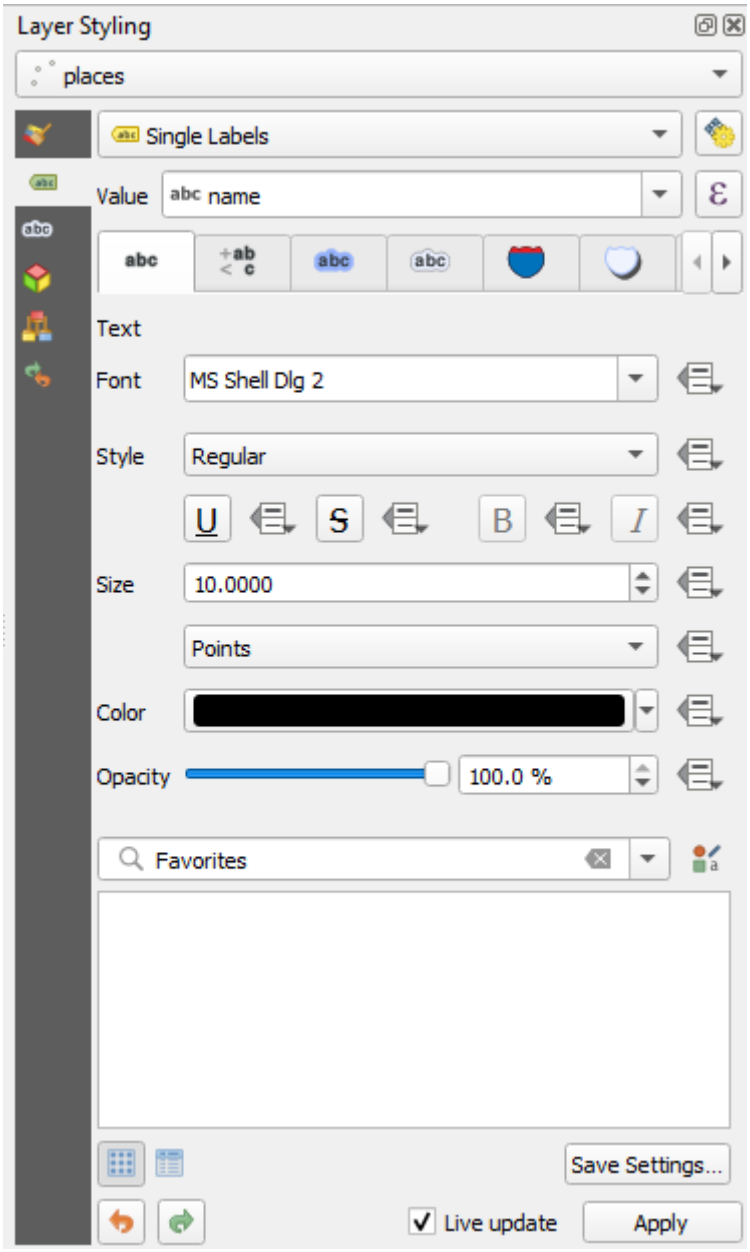
The goal for this lesson: To apply useful and good-looking labels to a layer.

3.2.1 Follow Along: Using Labels

First, ensure that the button is visible in the GUI:

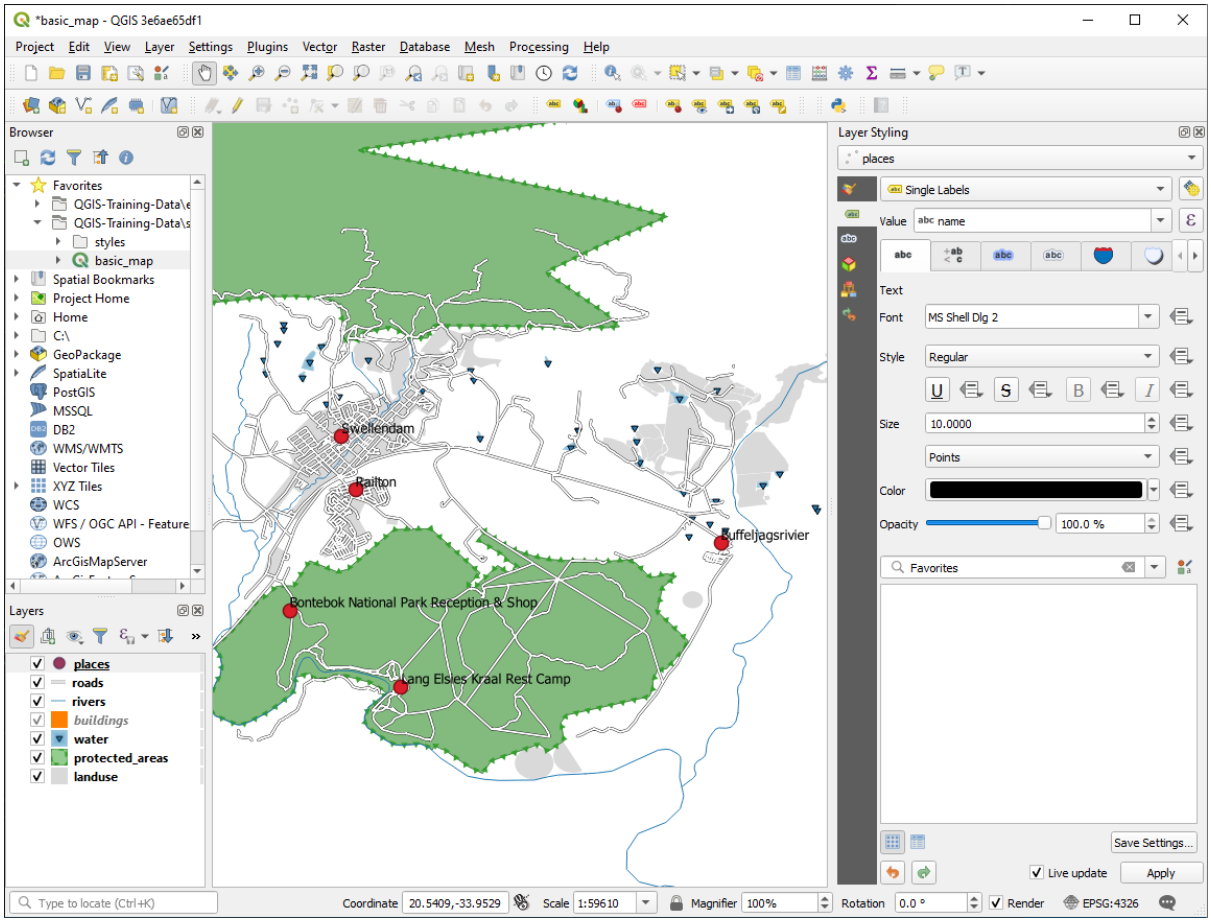
1. Go to the menu item *View* ► *Toolbars*
2. Ensure that the *Label Toolbar* item has a check mark next to it. If it doesn't, click on the *Label Toolbar* item to activate it.
3. Click on the `places` layer in the *Layers* panel so that it is highlighted
4. Click on the toolbar button to open the *Labels* tab of the *Layer Styling* panel
5. Switch from *No Labels* to *Single Labels*

   You'll need to choose which field in the attributes will be used for the labels. In the previous lesson, you decided that the `name` field was the most suitable one for this purpose.
6. Select `name` from the Value list:



7. Klicken Sie auf *Anwenden*

The map should now have labels like this:



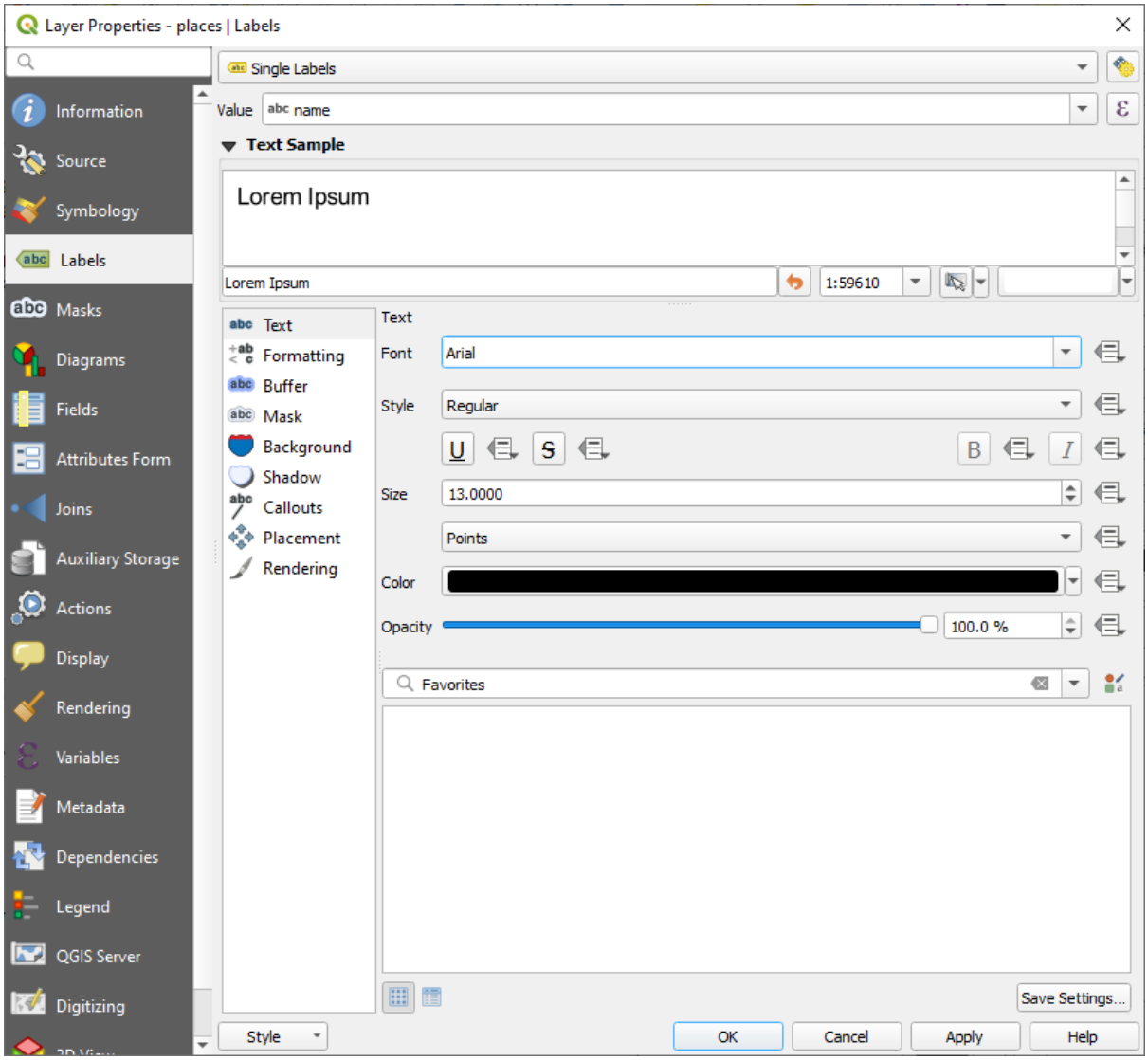
3.2.2 Follow Along: Changing Label Options

Depending on the styles you chose for your map in earlier lessons, you might find that the labels are not appropriately formatted and either overlap or are too far away from their point markers.

Bemerkung: Above, you used the abc button in the *Label Toolbar* to open the *Layer Styling* panel. As with *Symbology*, the same label options are available via both the *Layer Styling* panel and the *Layer Properties* dialog. Here, you'll use the *Layer Properties* dialog.

---

1. Open the *Layer Properties* dialog by double-clicking on the `places` layer
2. Select the abc *Labels* tab
3. Make sure *Text* is selected in the left-hand options list, then update the text formatting options to match those shown here:

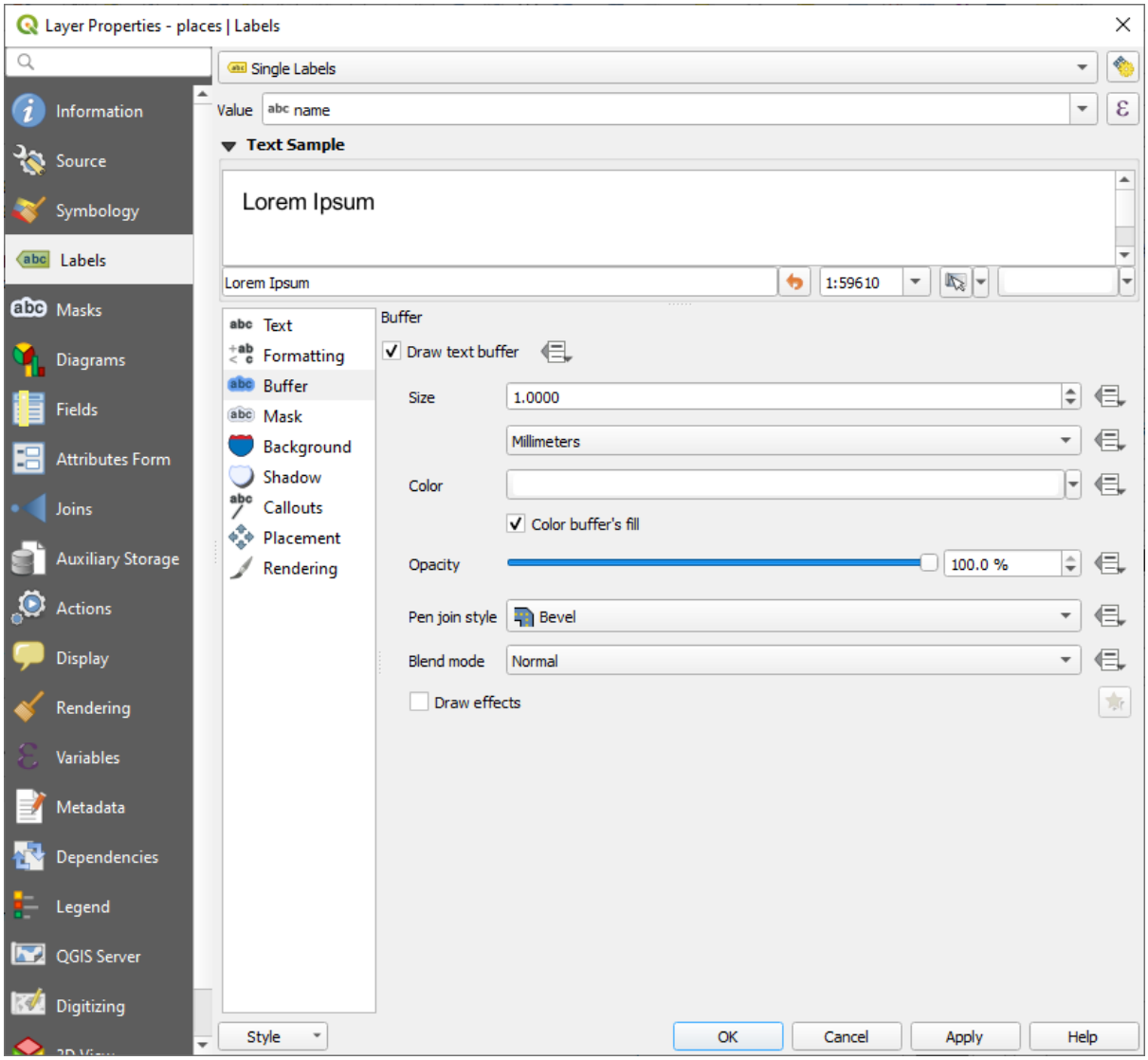


4. Klicken Sie auf *Anwenden*

   That font may be larger and more familiar to users, but its readability is still dependent on what layers are rendered beneath it. To solve this, let's take a look at the *Buffer* option.

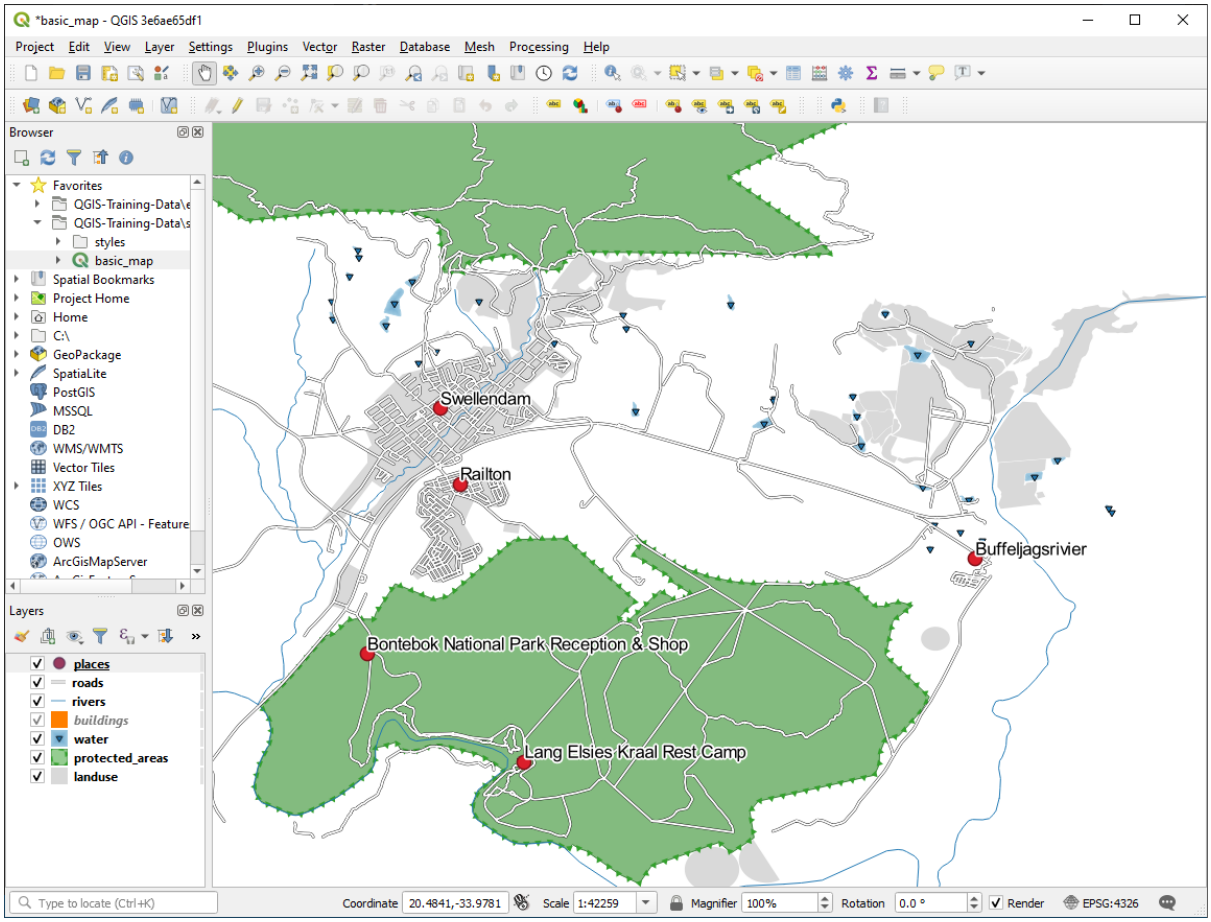
5. Select *Buffer* from the left-hand options list

6. Select the checkbox next to *Draw text buffer*, then choose options to match those shown here:



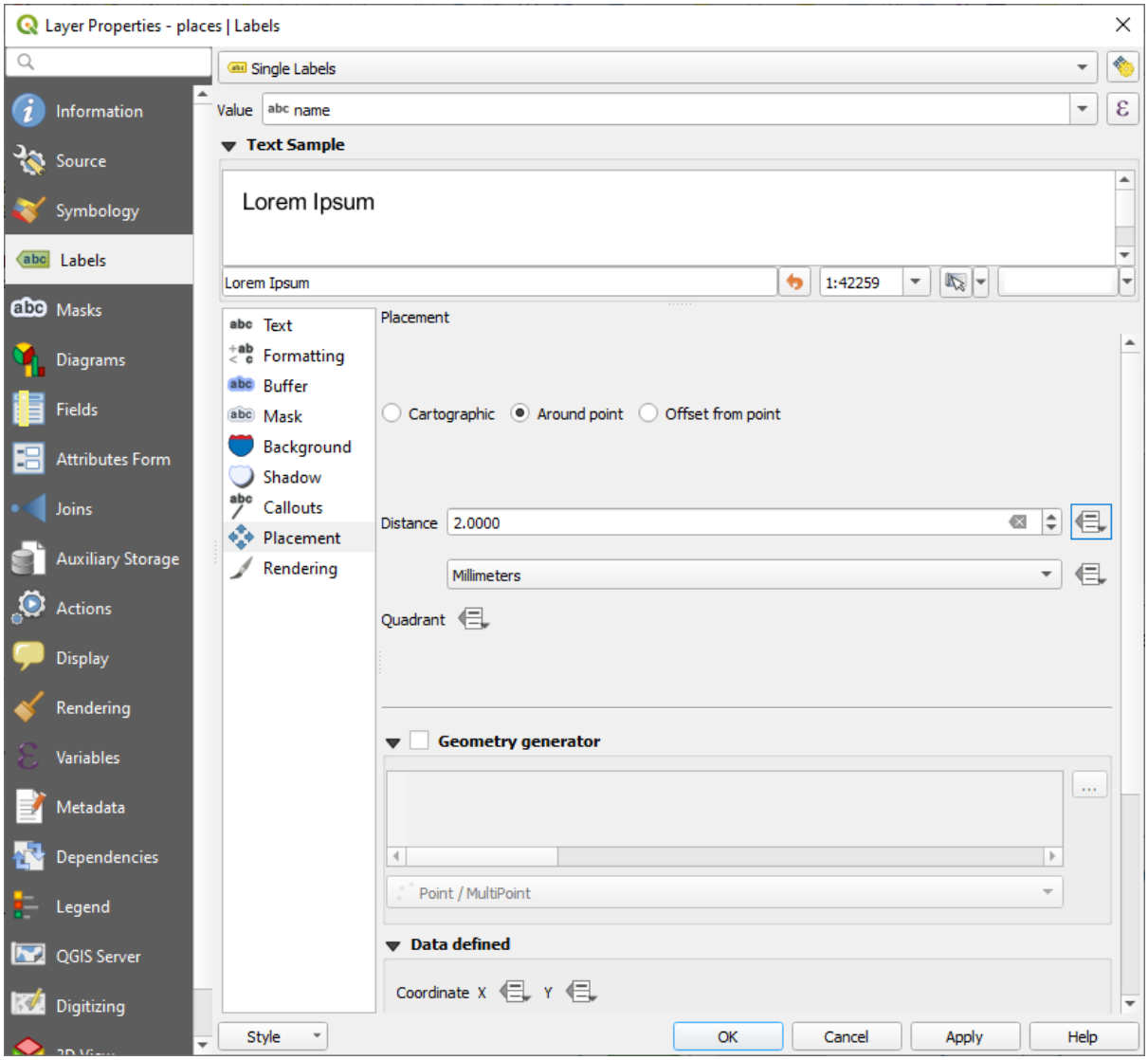
7. Klicken Sie auf *Anwenden*

   You'll see that this adds a colored buffer or border to the place labels, making them easier to pick out on the map:



Now we can address the positioning of the labels in relation to their point markers.

8. Select *Placement* from the left-hand options list
9. Select *Around point* and change the value of *Distance* to `2.0` Millimeters:



10. Klicken Sie auf *Anwenden*

    You'll see that the labels are no longer overlapping their point markers.

3.2.3 Follow Along: Using Labels Instead of Layer Symbology

In many cases, the location of a point doesn't need to be very specific. For example, most of the points in the `places` layer refer to entire towns or suburbs, and the specific point associated with such features is not that specific on a large scale. In fact, giving a point that is too specific is often confusing for someone reading a map.

To name an example: on a map of the world, the point given for the European Union may be somewhere in Poland, for instance. To someone reading the map, seeing a point labeled *European Union* in Poland, it may seem that the capital of the European Union is therefore in Poland.

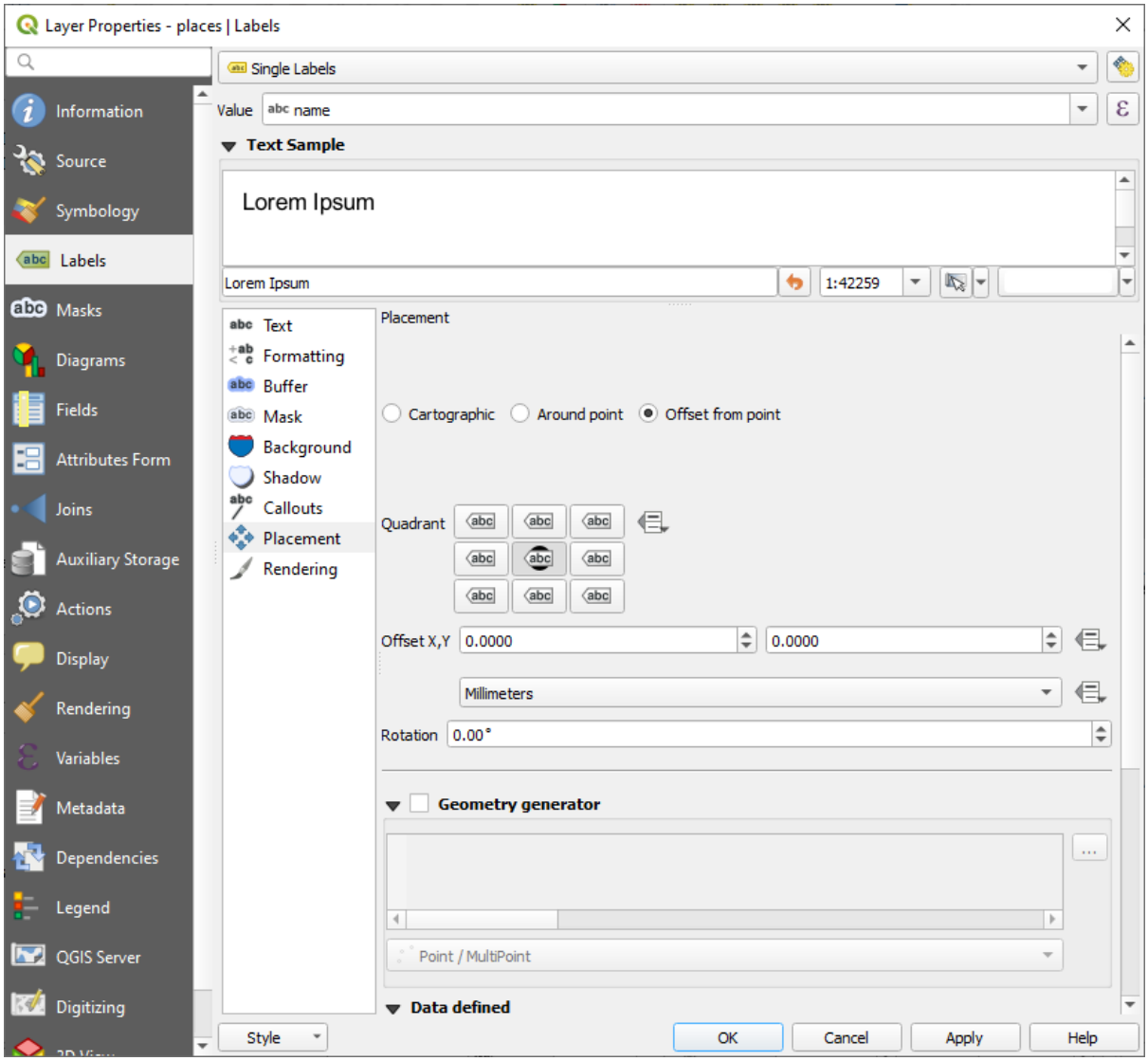
So, to prevent this kind of misunderstanding, it's often useful to deactivate the point symbols and replace them completely with labels.

In QGIS, you can do this by changing the position of the labels to be rendered directly over the points they refer to.

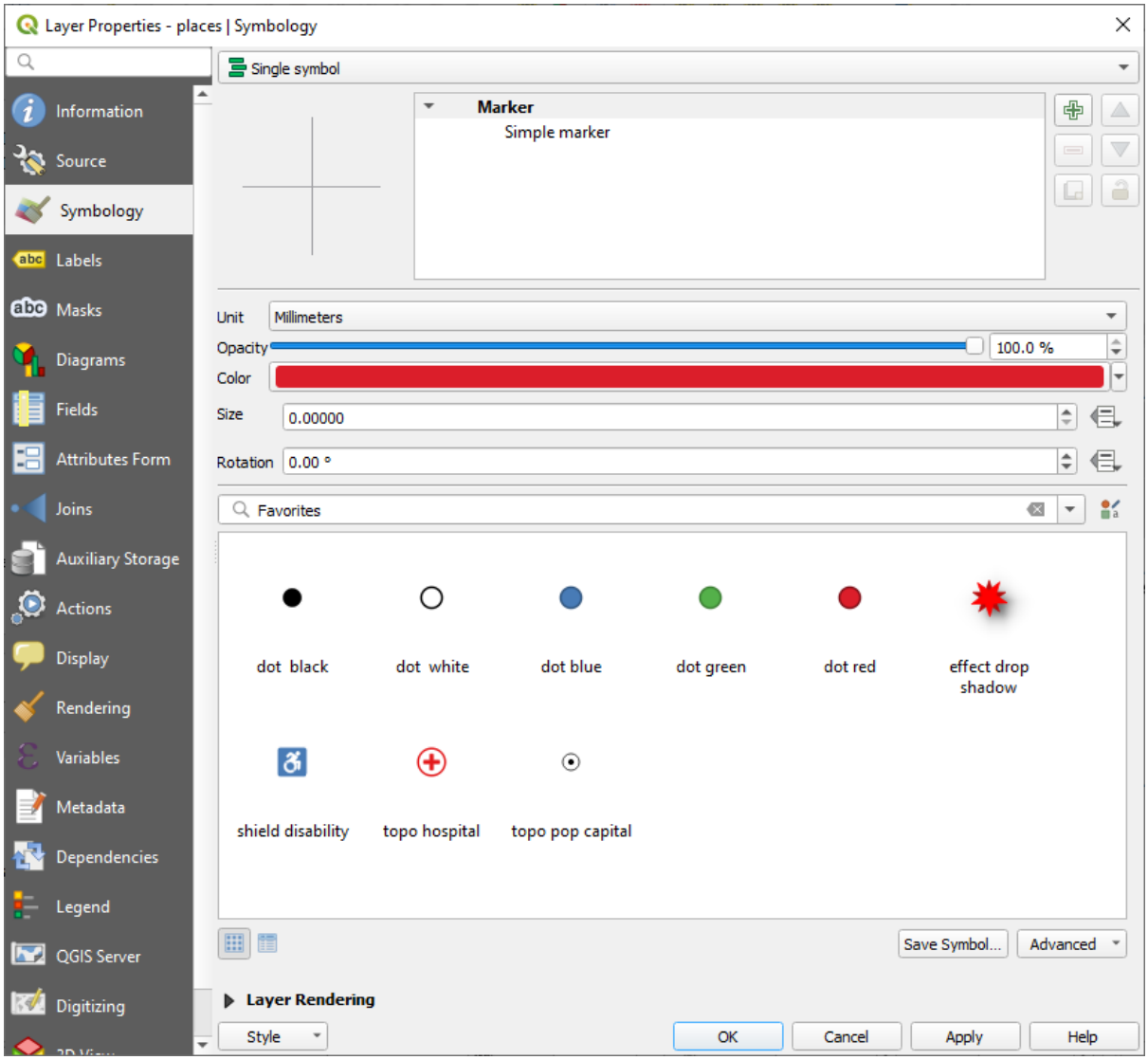
1. Open the *Labels* tab of the *Layer Properties* dialog for the `places` layer
2. Select the *Placement* option from the options list

3. Click on the *Offset from point* button

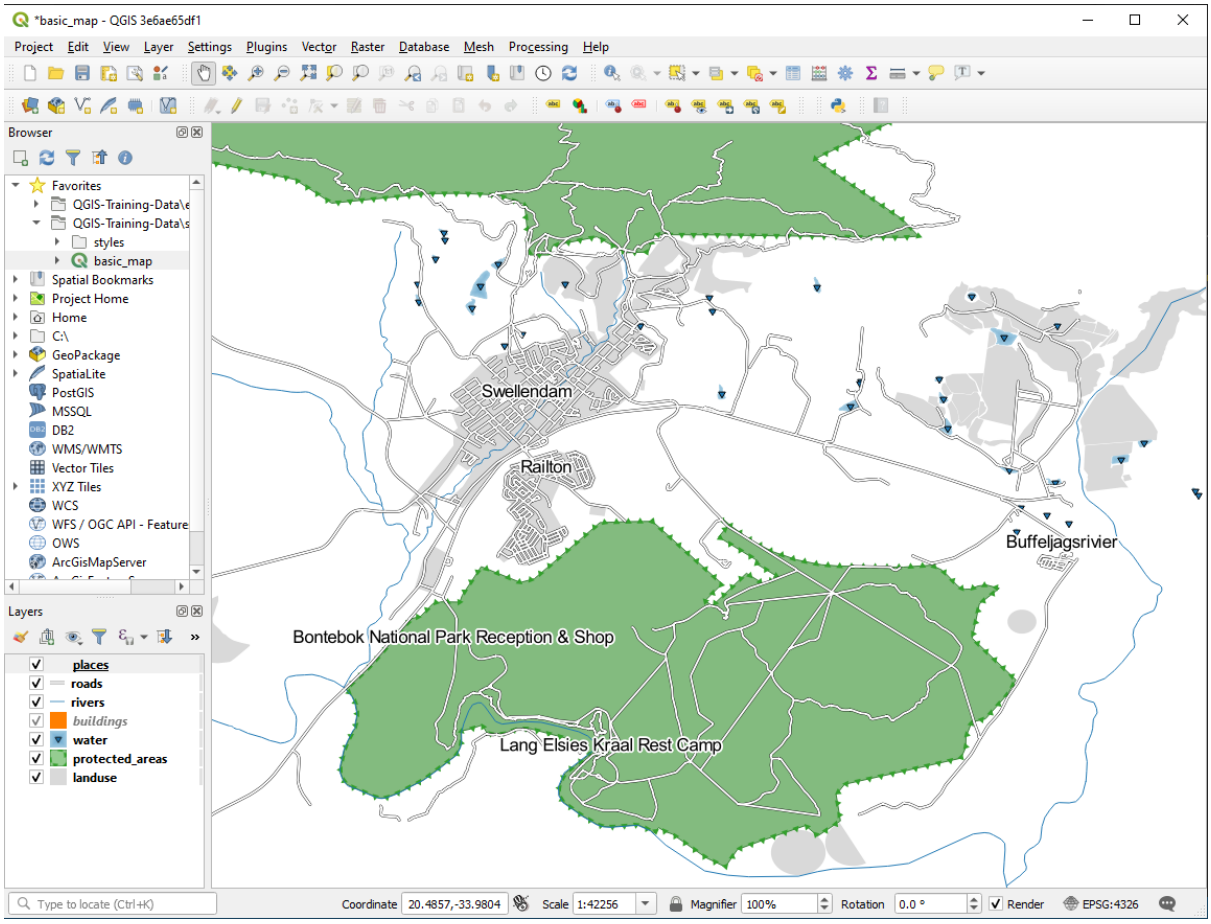
   This will reveal the *Quadrant* options which you can use to set the position of the label in relation to the point marker. In this case, we want the label to be centered on the point, so choose the center quadrant:



4. Hide the point symbols by editing the layer *Symbology* as usual, and setting the size of the *Marker* size to `0.0`:



5. Click *Apply* and you'll see this result:



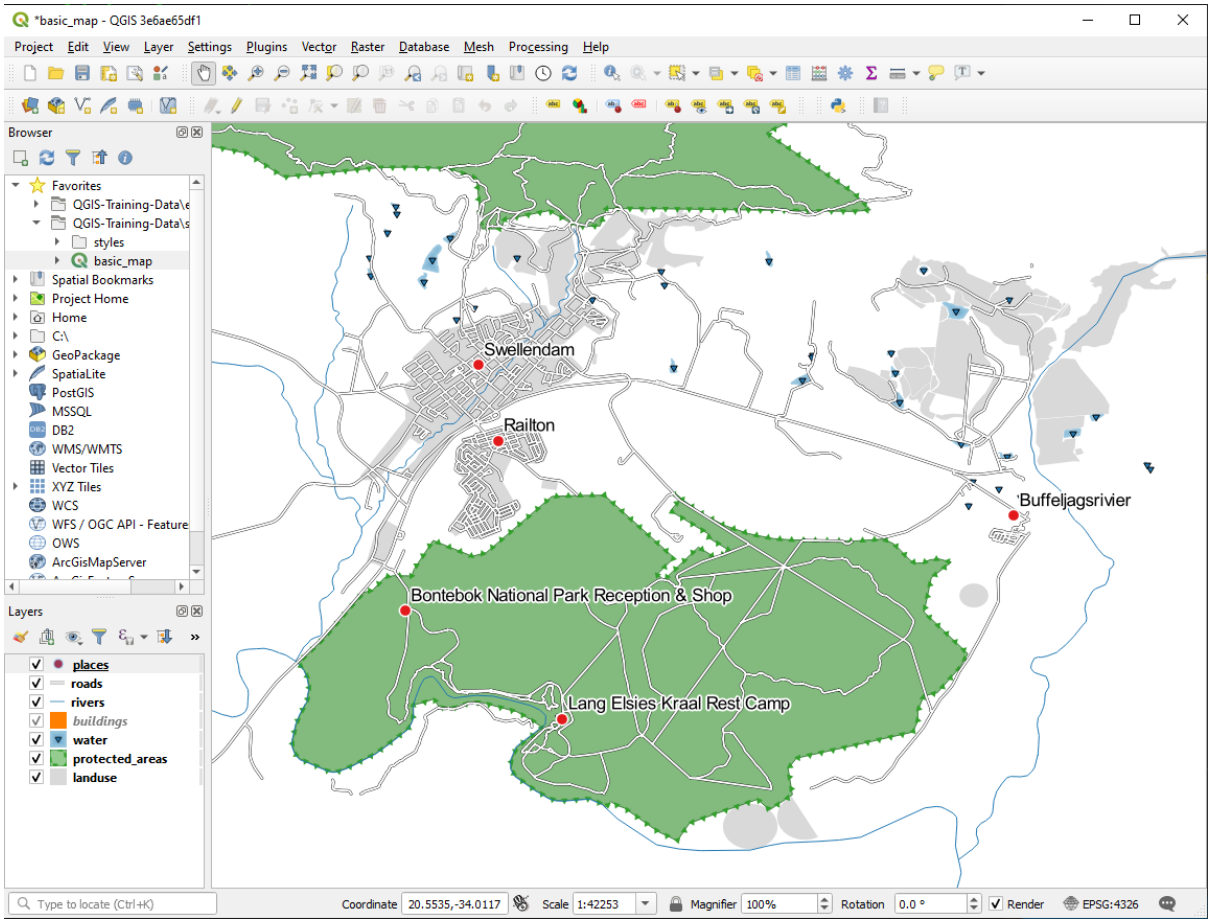
If you were to zoom out on the map, you would see that some of the labels disappear at larger scales to avoid overlapping. Sometimes this is what you want when dealing with datasets that have many points, but at other times you will lose useful information this way. There is another possibility for handling cases like this, which we'll cover in a later exercise in this lesson. For now, zoom out and click on the button in the toolbar and see what happens.

3.2.4 Try Yourself Customize the Labels

- Reset the label and symbol settings to have a point marker and a label offset of `2.0` Millimeters.

Answer

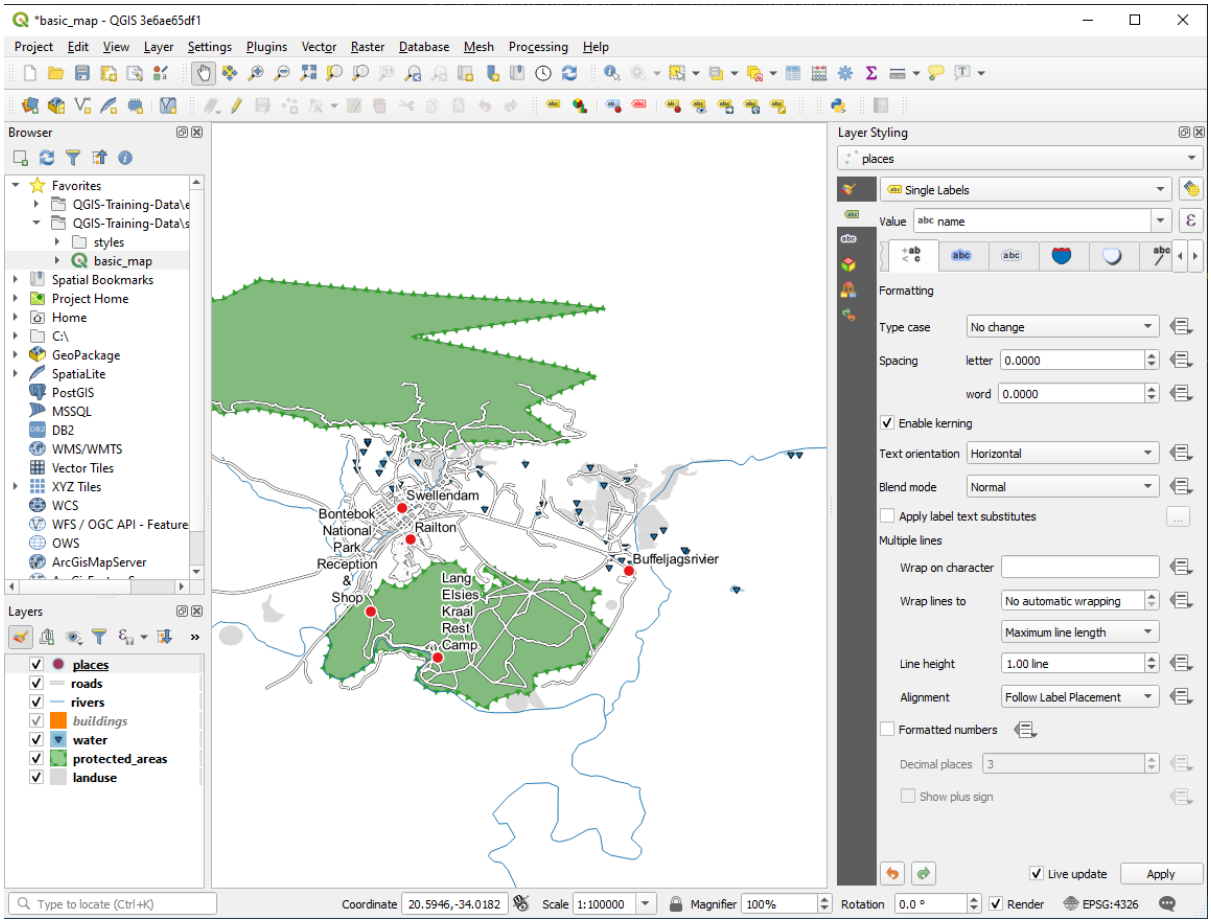
Your map should now show the marker points and the labels should be offset by 2mm. The style of the markers and labels should allow both to be clearly visible on the map:



- Set the map to the scale `1:100000`. You can do this by typing it into the *Scale* box in the *Status Bar*. Modify your labels to be suitable for viewing at this scale.

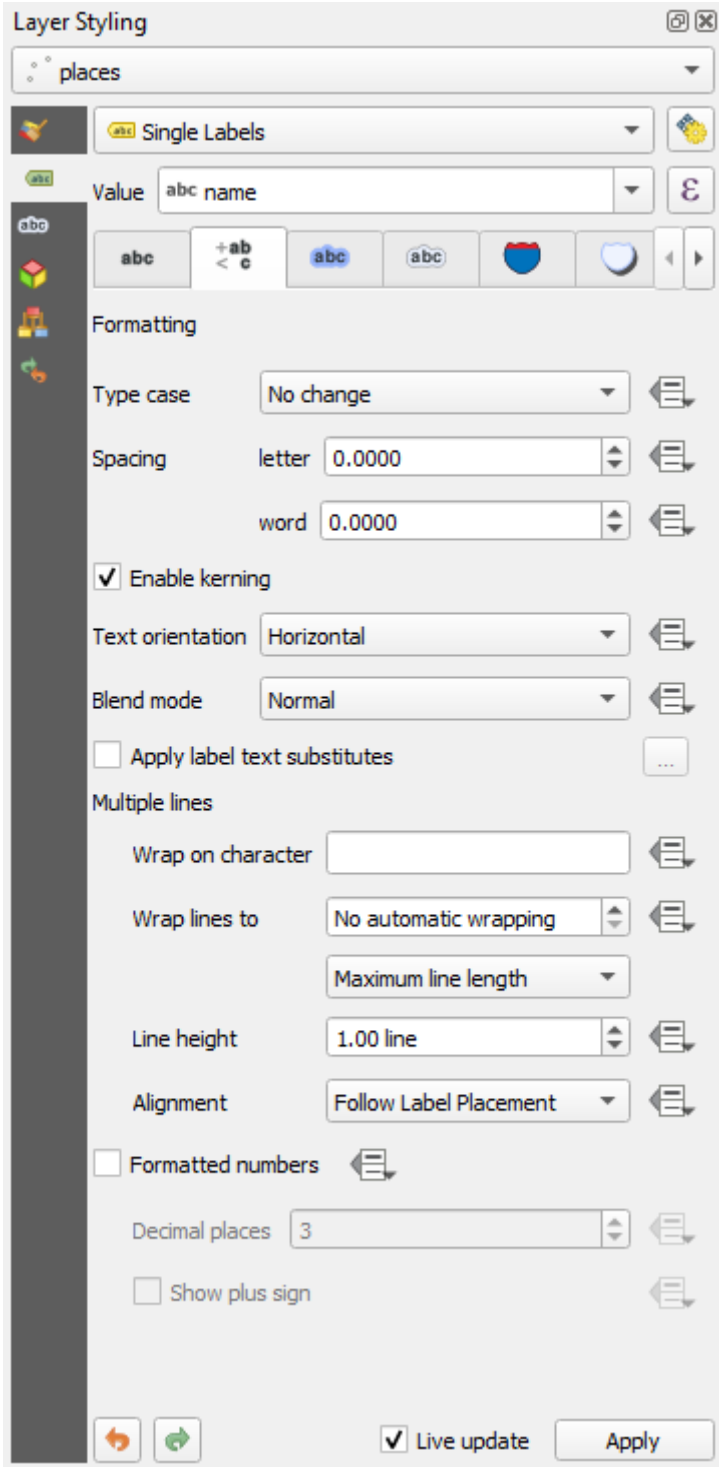
Answer

One possible solution has this final product:



To arrive at this result:

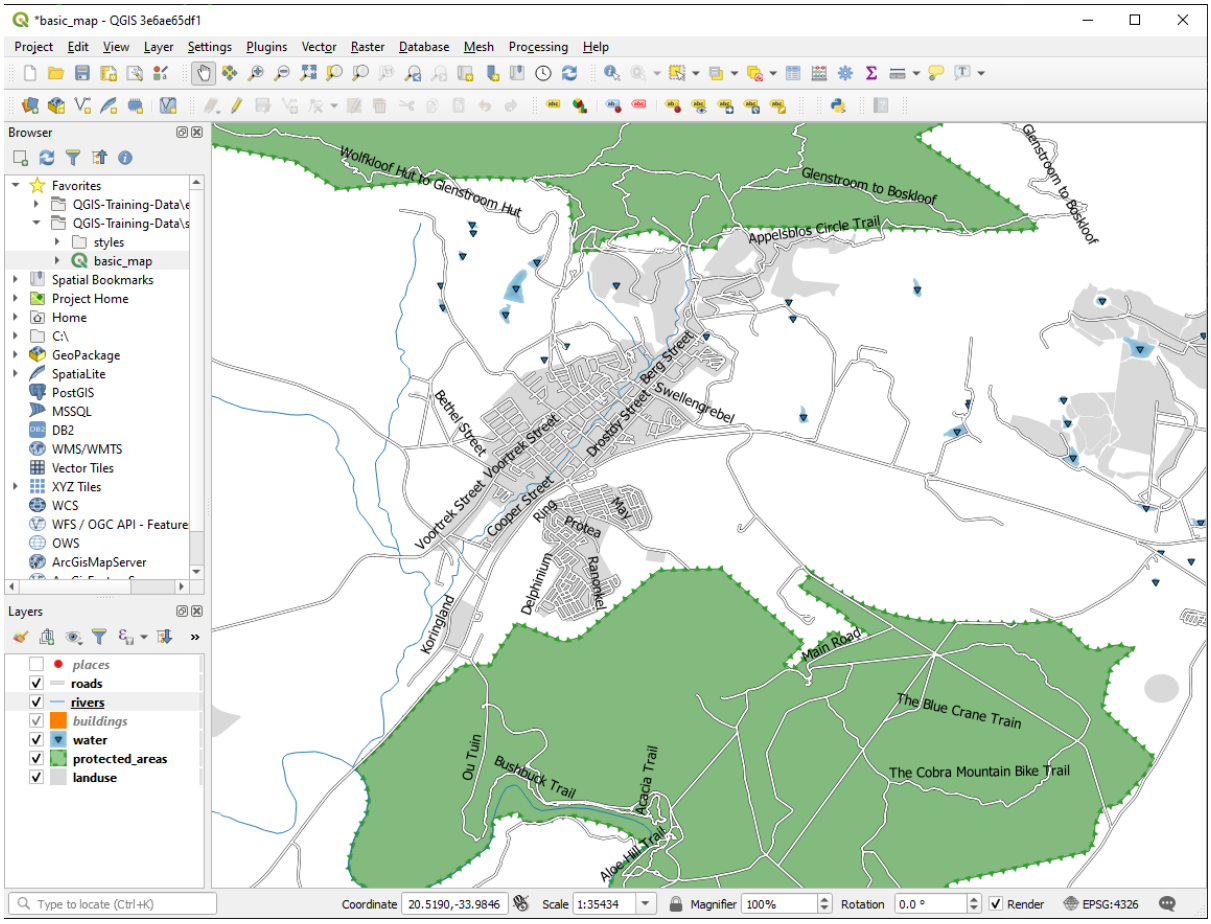
- Use a font size of `10`
- Use an around point placement distance of `1.5 mm`
- Use a marker size of `3.0 mm`
- In addition, this example uses the *Wrap on character* option:



- 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.

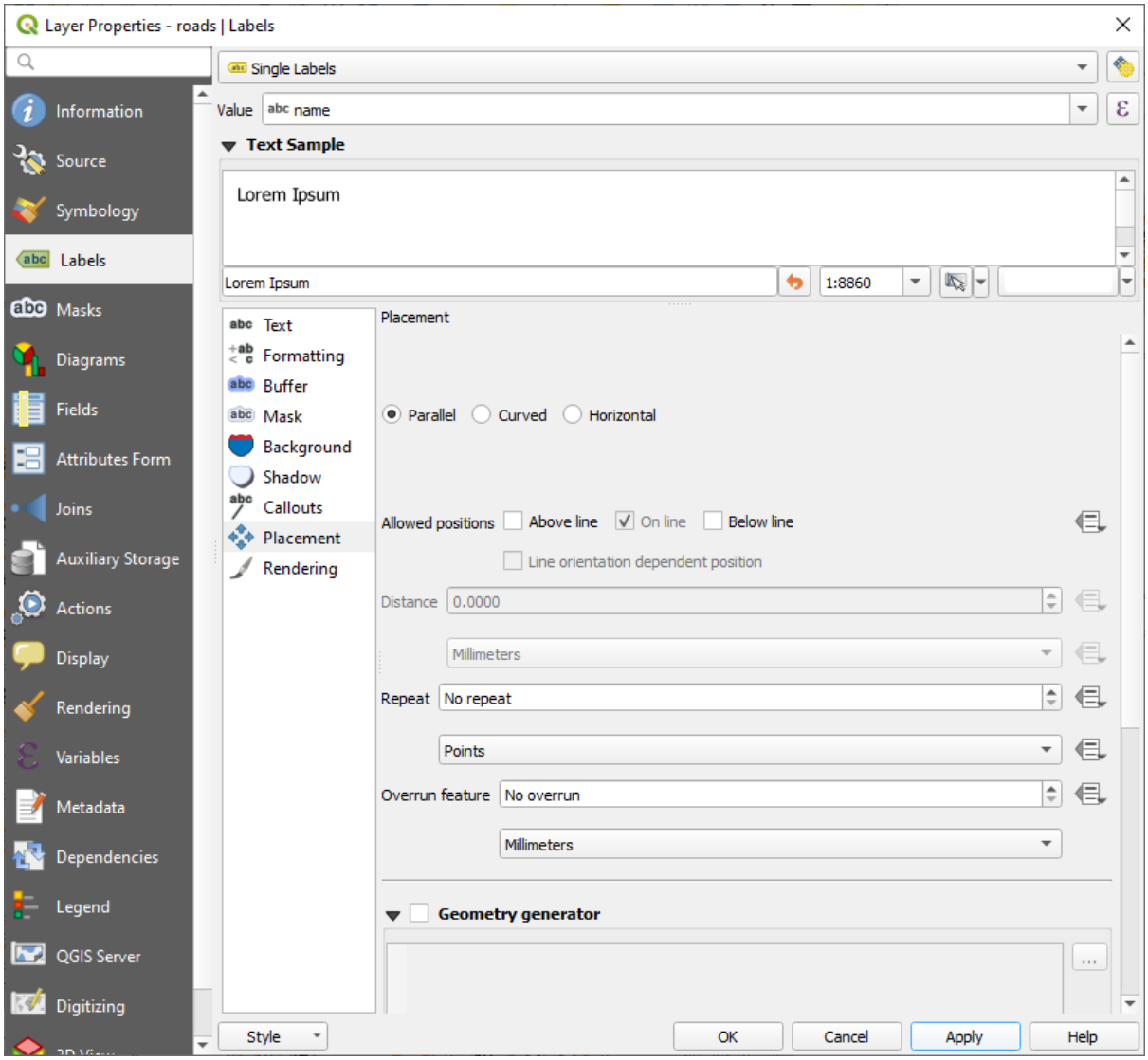
3.2.5 Follow Along: Labeling Lines

Now that you know how labeling works, there's an additional problem. Points and polygons are easy to label, but what about lines? If you label them the same way as the points, your results would look like this:



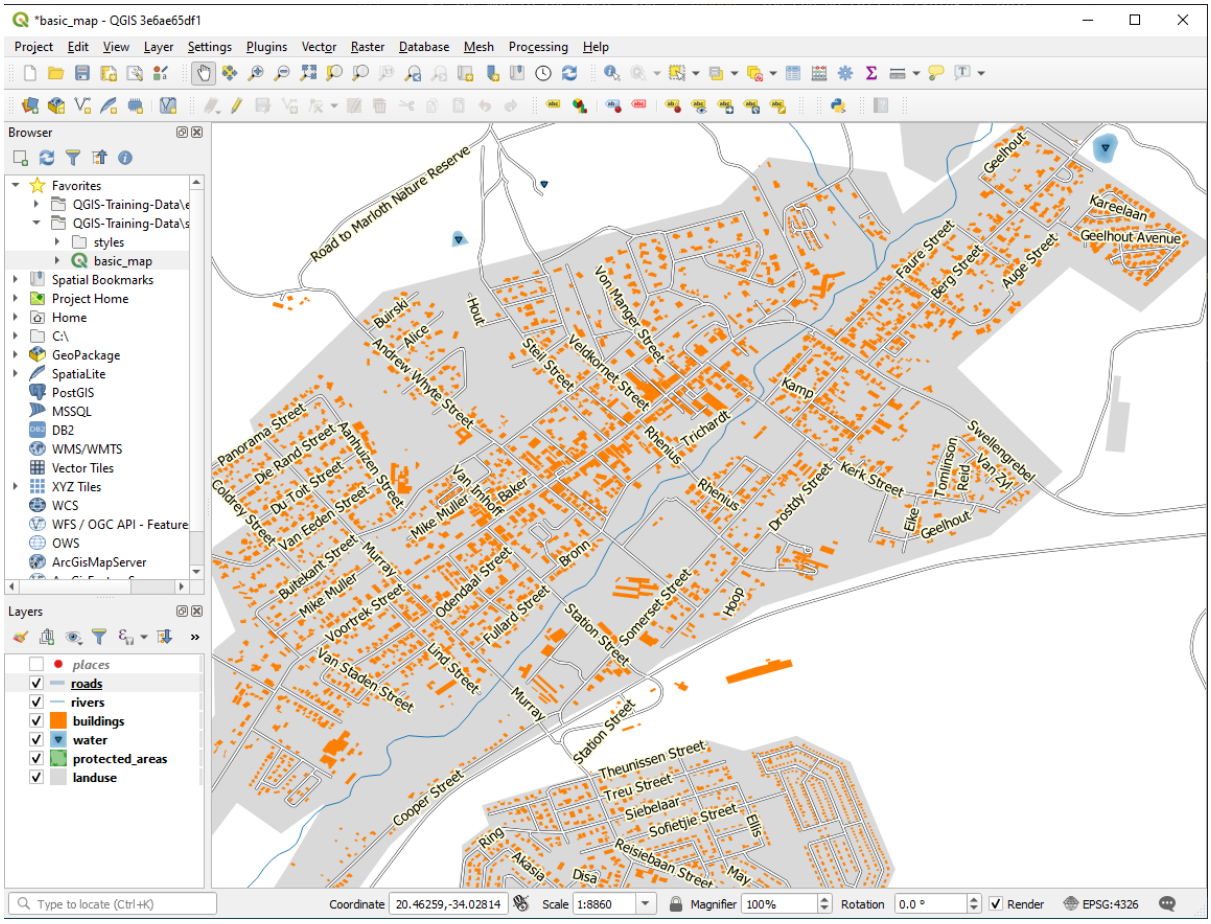
We will now reformat the `roads` layer labels so that they are easy to understand.

1. Hide the `places` layer so that it doesn't distract you
2. Activate *Single Labels* for the `roads` layer as you did above for `places`
3. Set the font *Size* to `10` so that you can see more labels
4. Zoom in on the Swellendam town area
5. In the *Labels* tab's *Placement* tab, choose the following settings:



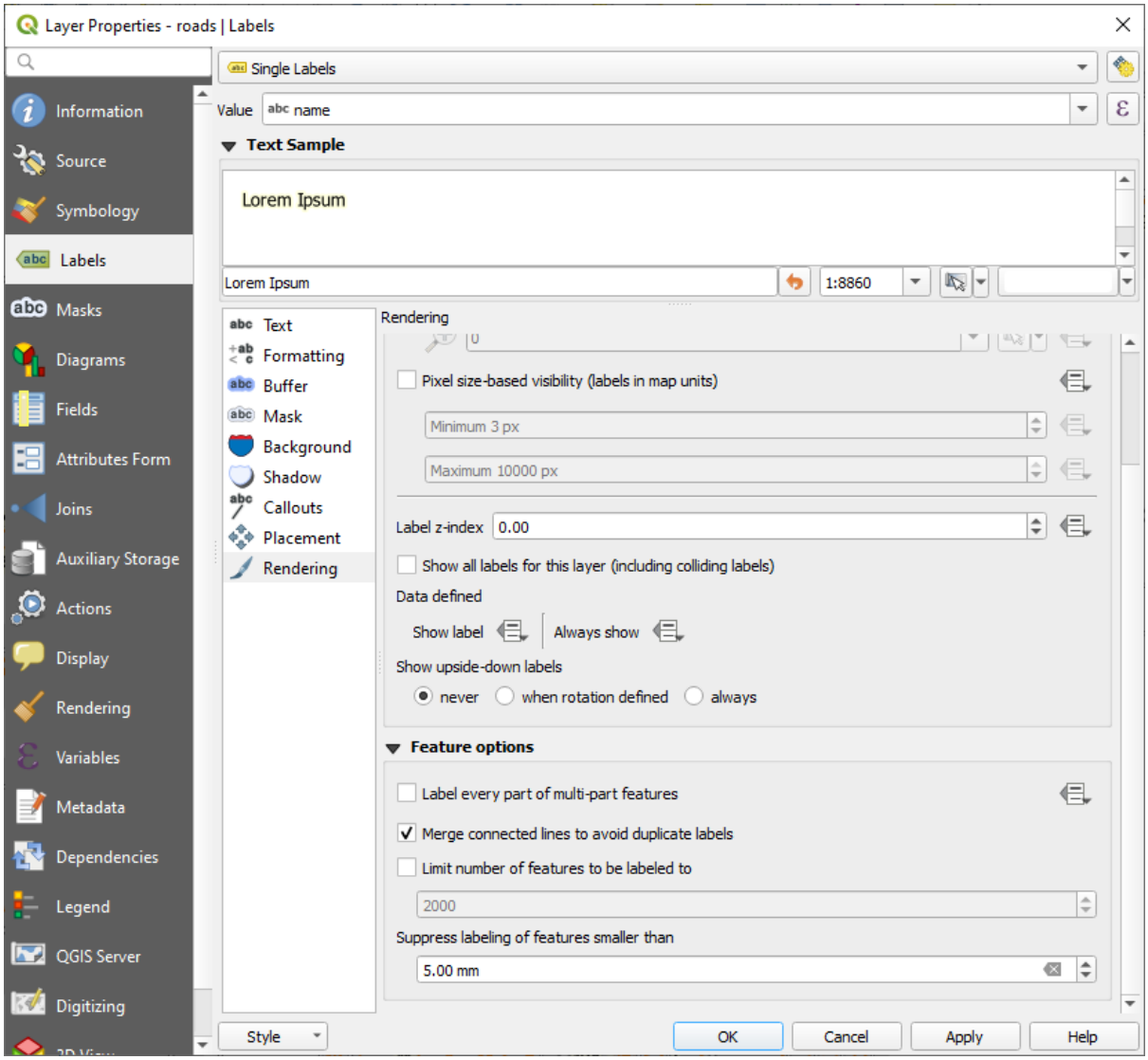
You'll probably find that the text styling has used default values and the labels are consequently very hard to read. Update the *Text* to use a dark-grey or black *Color* and the *Buffer* to use a light-yellow *Color*.

The map will look somewhat like this, depending on scale:



You'll see that some of the road names appear more than once and that's not always necessary. To prevent this from happening:

6. In the *Labels* tab of the *Layer Properties* dialog, choose the *Rendering* option and select *Merge connected lines to avoid duplicate labels* as shown:



7. Klicken Sie auf *OK*

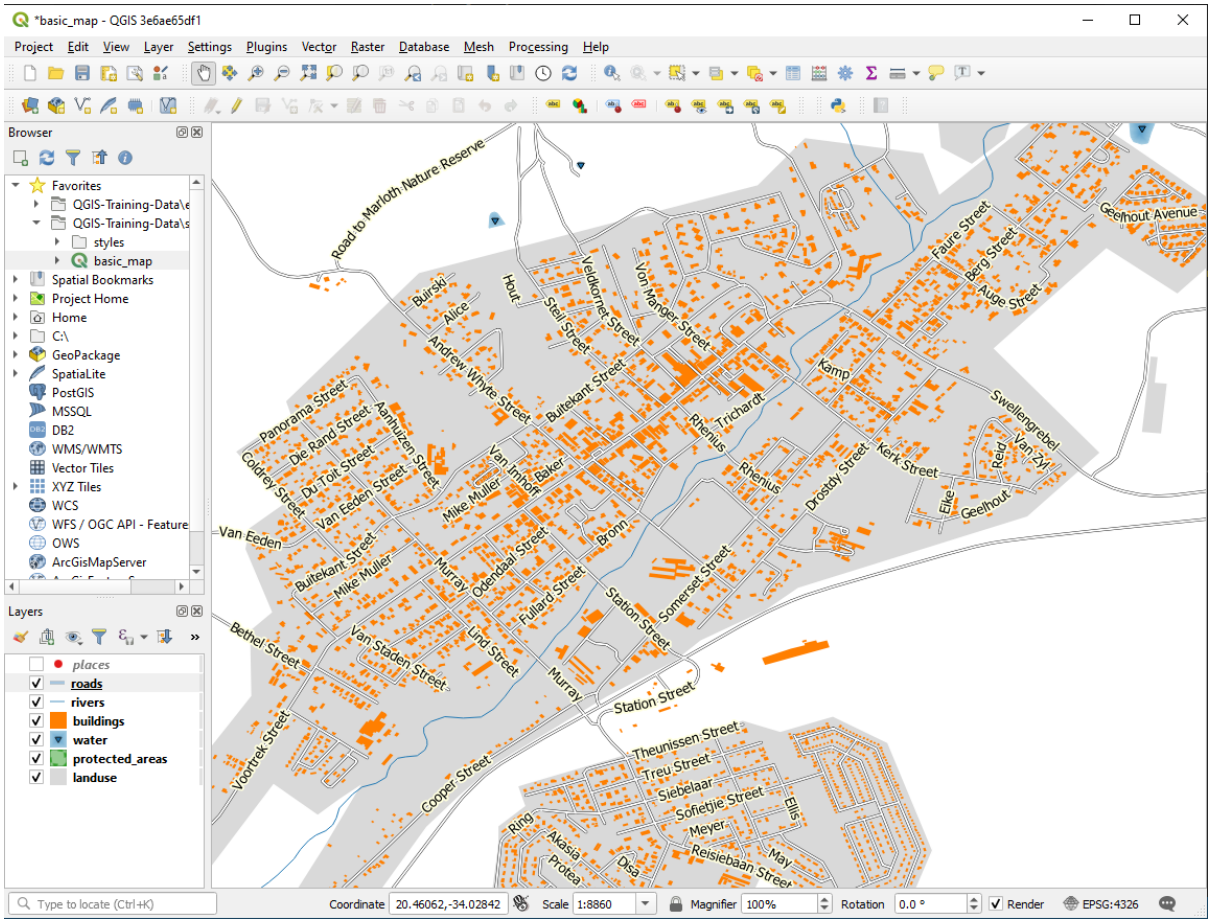
   Another useful function is to prevent labels being drawn for features too short to be of notice.

8. In the same *Rendering* panel, set the value of *Suppress labeling of features smaller than …* to `5.00 mm` and note the results when you click *Apply*

   Try out different *Placement* settings as well. As we've seen before, the *Horizontal* option is not a good idea in this case, so let's try the *Curved* option instead.

9. Select the *Curved* option in the *Placement* panel of the *Labels* tab

Here's the result:

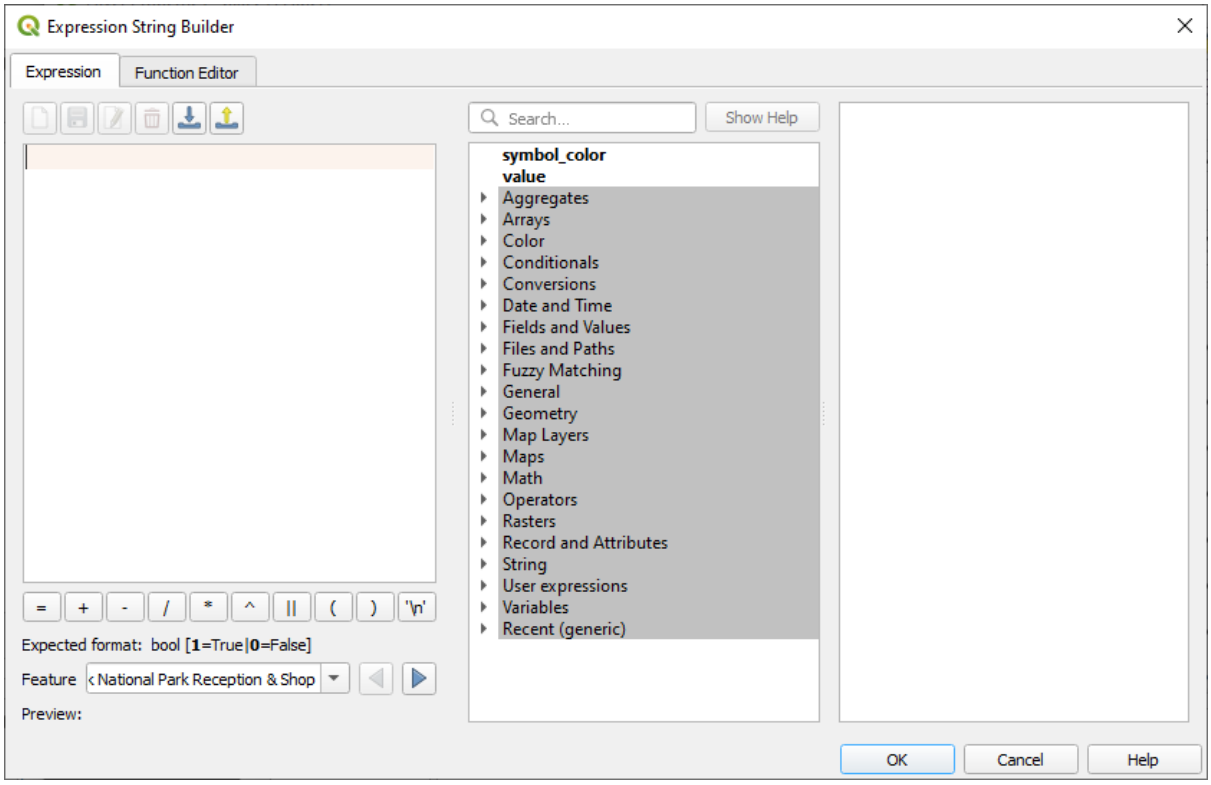


As you can see, this hides some labels that were previously visible, because of the difficulty of making some of them follow twisting street lines while still being legible. It makes other labels much more useful since they track the roads rather than float in space between them. You can decide which of these options to use, depending on what you think seems more useful or what looks better.

3.2.6 Follow Along: Data Defined Settings

1. Deactivate labeling for the `roads` layer
2. Reactivate labeling for the `places` layer
3. Open the attribute table for `places` via the button

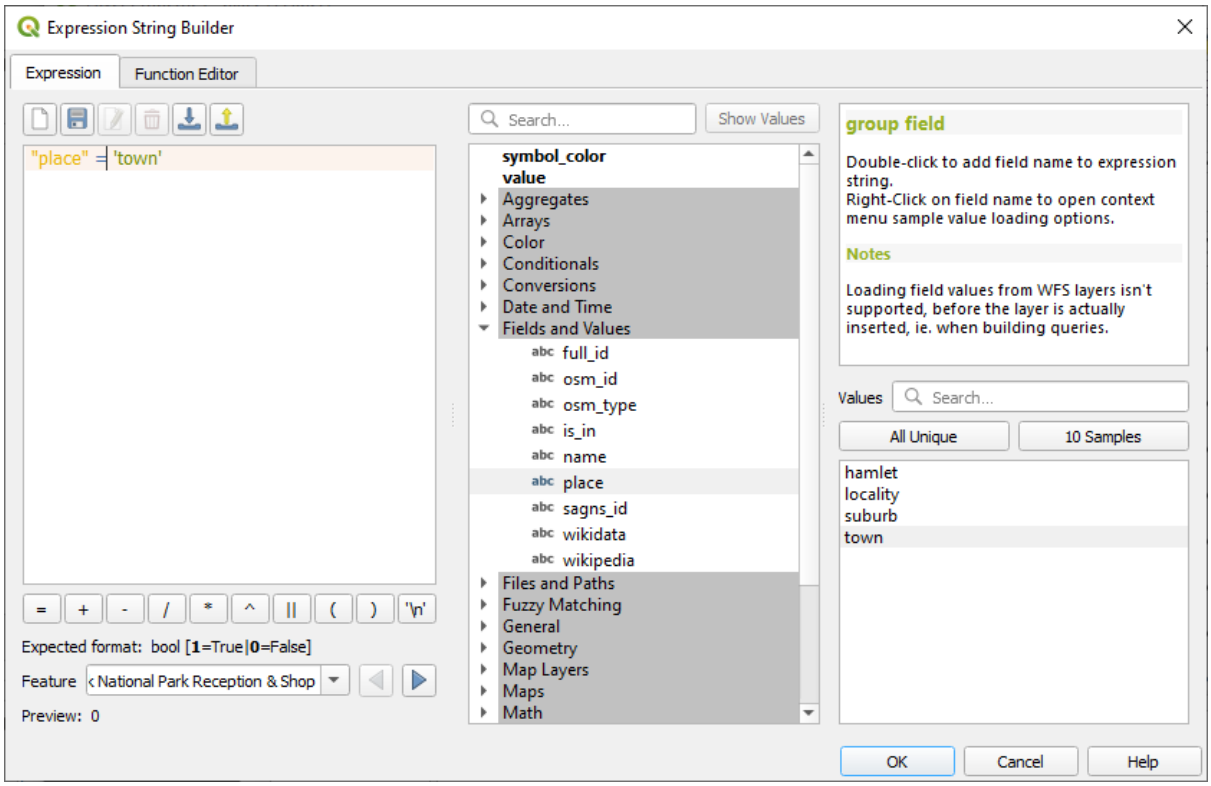
   It has one *field* which is of interest to us now: `place` which defines the type of urban area for each *record*. We can use this data to influence the label styles.
4. Navigate to the *Text* panel in the `places` *Labels* panel
5. Click the button next to the Italic text button beneath *Style* and select *Edit…* to open the *Expression String Builder*:



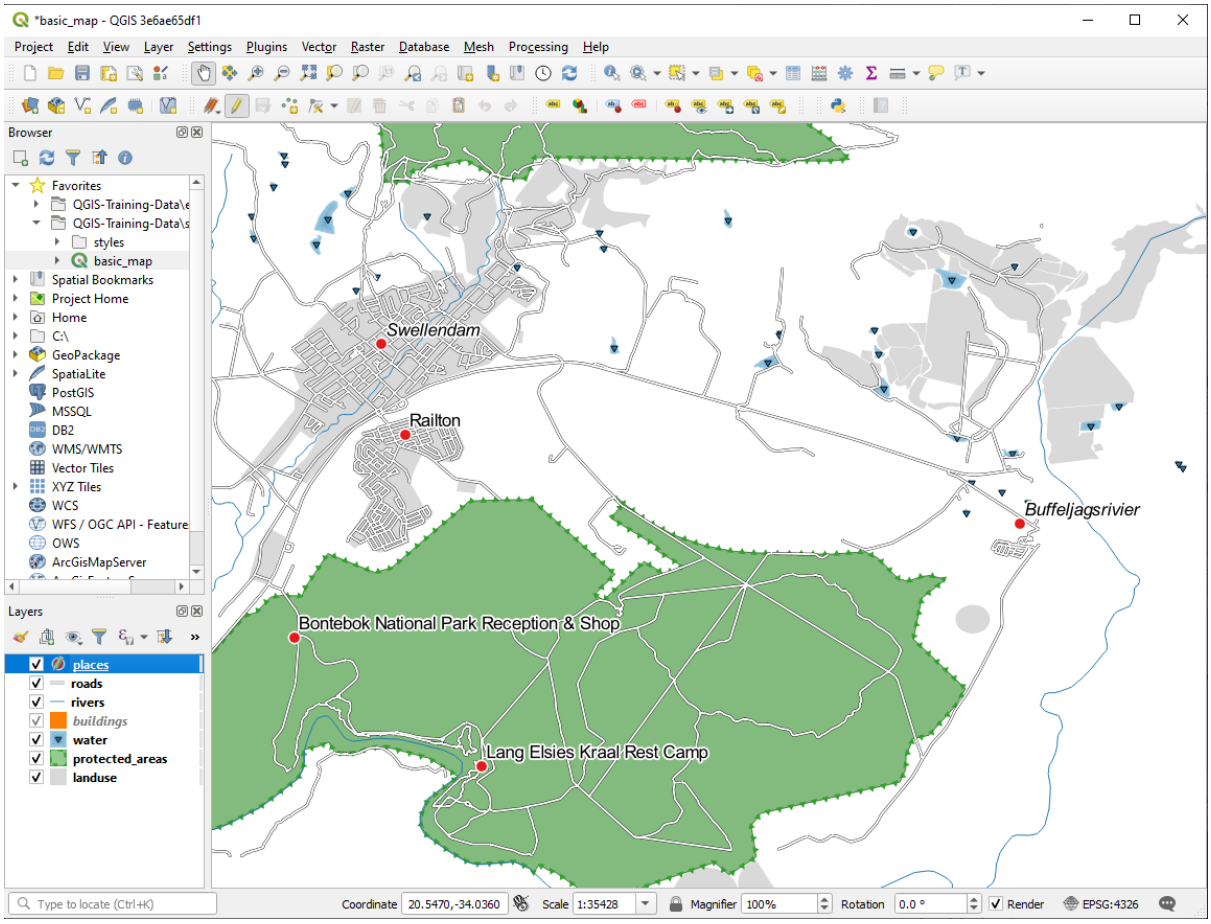
6. Under *Fields and Values*, double click on `place` and then click *All Unique*. This will list all unique values of the `place` field of this layer. Add a = in the text editor and then double click on `town`.

   Alternatively, you can type: `"place" = 'town'` directly in the text editor.

7. Click *OK* twice:



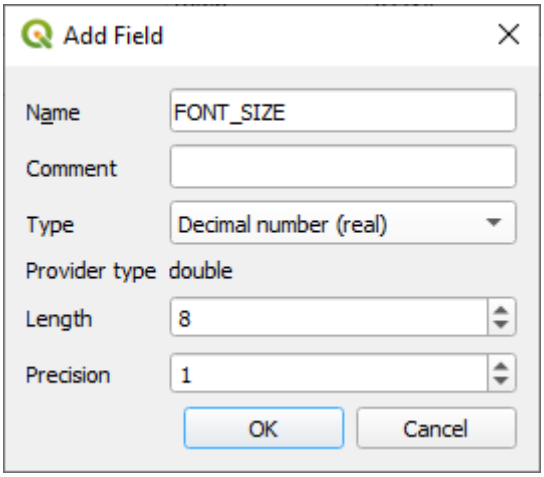
Notice that the labels for all places whose `place` field matches `town` are displayed in italics.



3.2.7 Try Yourself Using Data Defined Settings

Bemerkung: We're jumping ahead a bit here to demonstrate some advanced labeling settings. At the advanced level, it's assumed that you'll know what the following means. If you don't, feel free to leave out this section and come back later when you've covered the requisite materials.

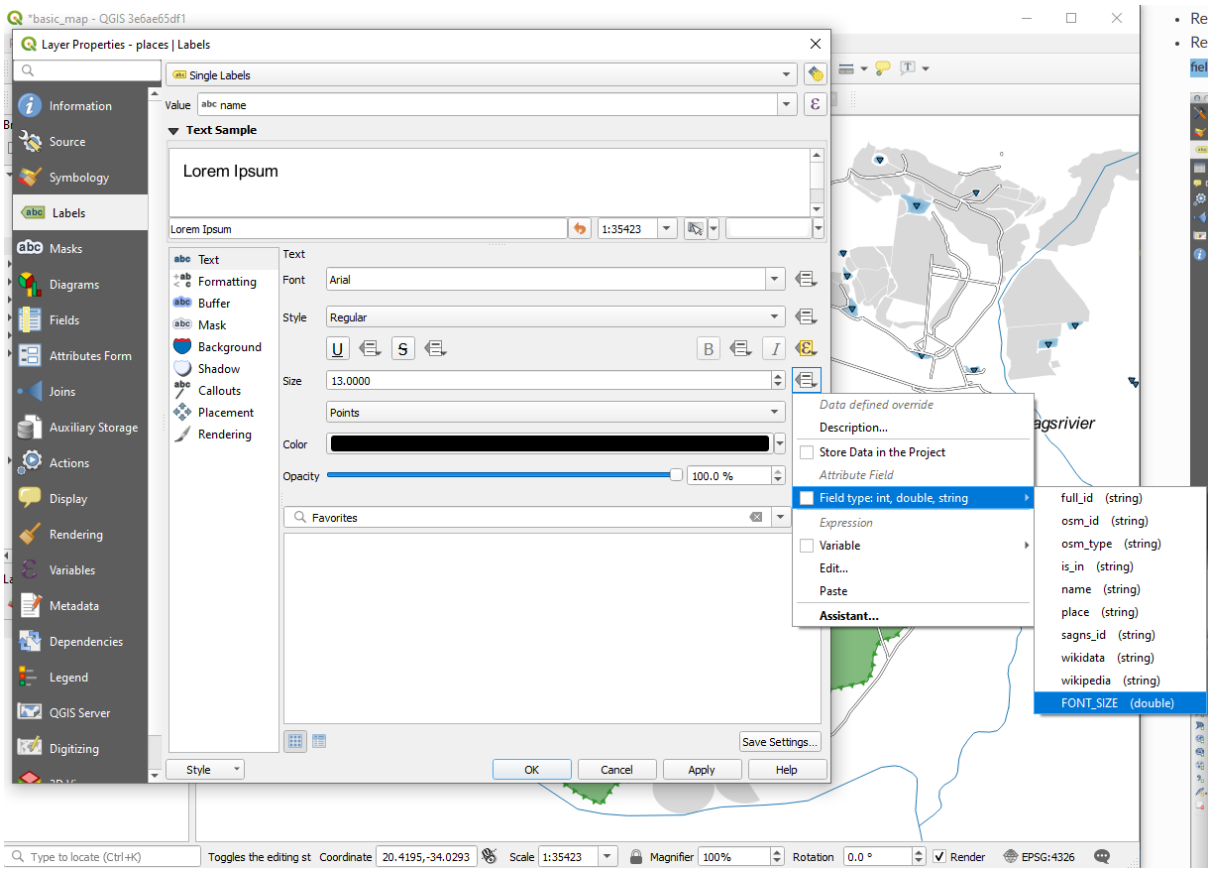
1. Open the Attribute Table for `places`
2. Enter edit mode by clicking the button
3. Add a new column with the button
4. Configure it like this:



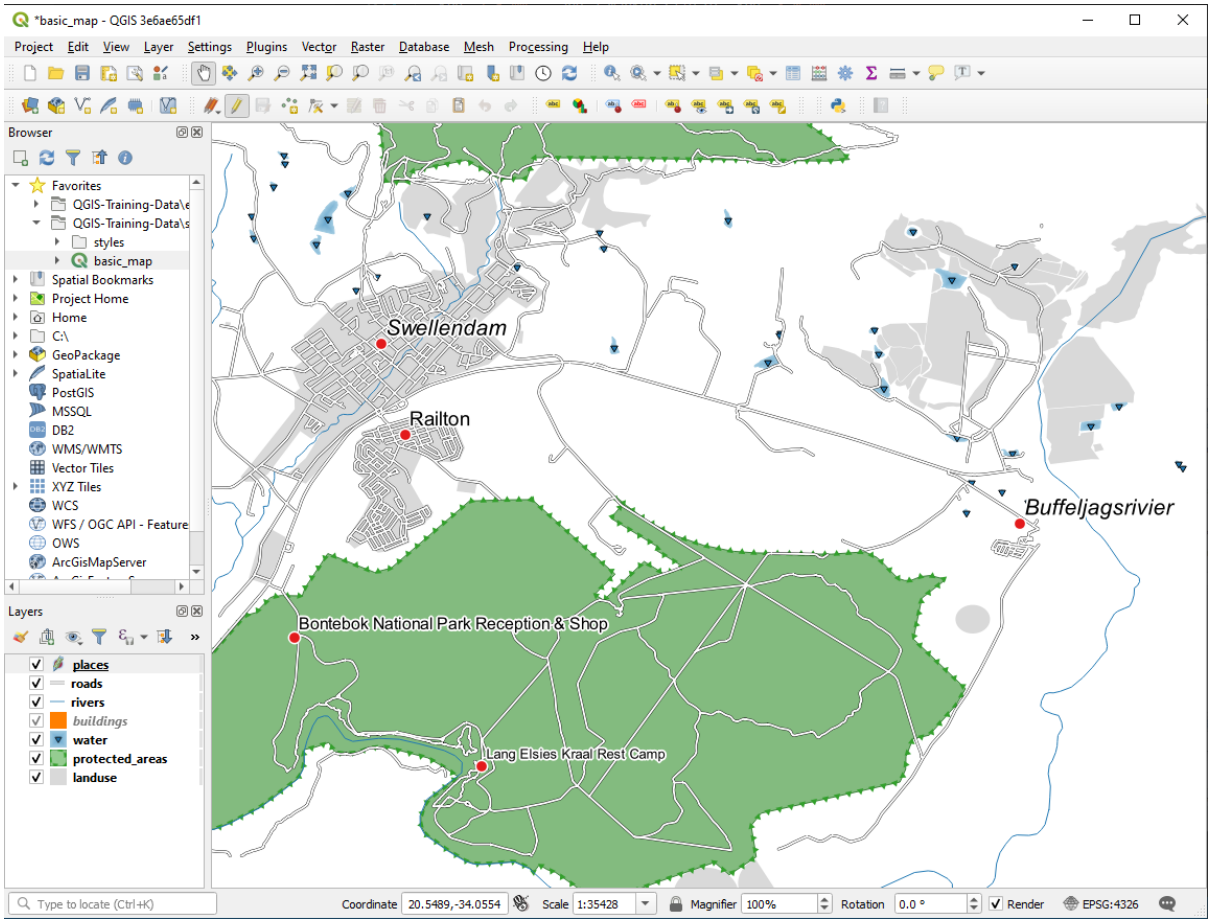
5. Use this to set custom font sizes for each different type of place (each key in the `place` field)

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 Further Possibilities With Labeling

We can't cover every option in this course, but be aware that the *Label* tab has many other useful functions. You can set scale-based rendering, alter the rendering priority for labels in a layer, and set every label option using layer attributes. You can even set the rotation, XY position, and other properties of a label (if you have attribute fields allocated for the purpose), then edit these properties using the tools adjacent to the main *Layer Labeling Options* button:

(These tools will be active if the required attribute fields exist and you are in edit mode.)

Feel free to explore more possibilities of the labeling system.

3.2.9 In Conclusion

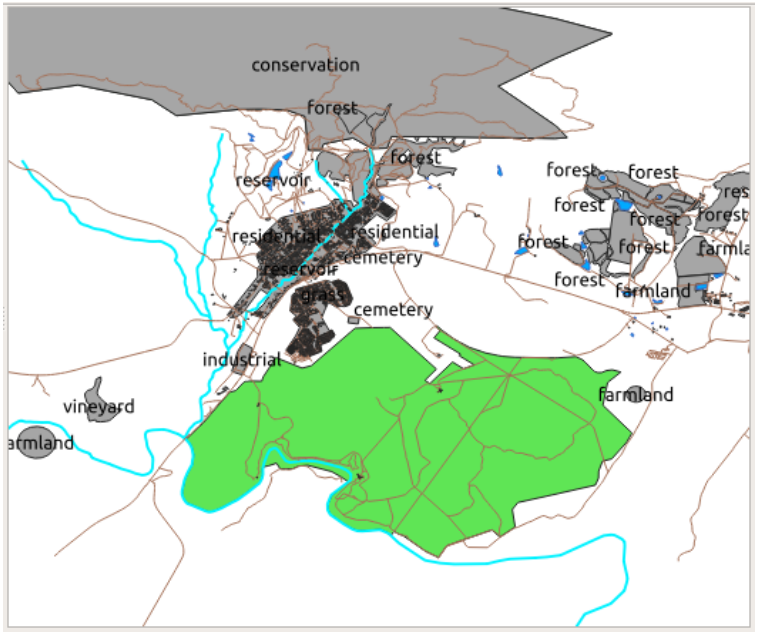
You've learned how to use layer attributes to create dynamic labels. This can make your map a lot more informative and stylish!

3.2.10 What's Next?

Now that you know how attributes can make a visual difference for your map, how about using them to change the symbology of objects themselves? That's the topic for the next lesson!

3.3 Lesson: Klassifikation

Labels are a good way to communicate information such as the names of individual places, but they can't be used for everything. For example, let us say that someone wants to know what each `landuse` area is used for. Using labels, you would get this:

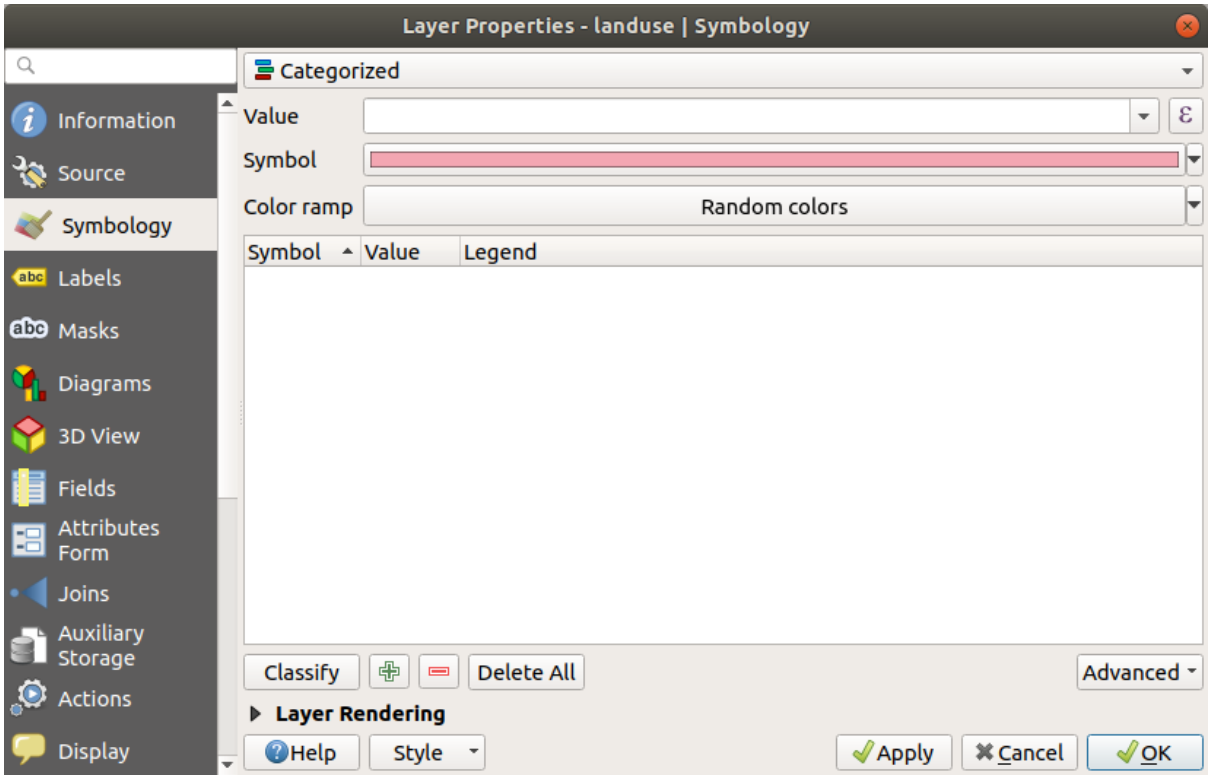


This makes the map's labeling difficult to read and even overwhelming if there are numerous different landuse areas on the map.

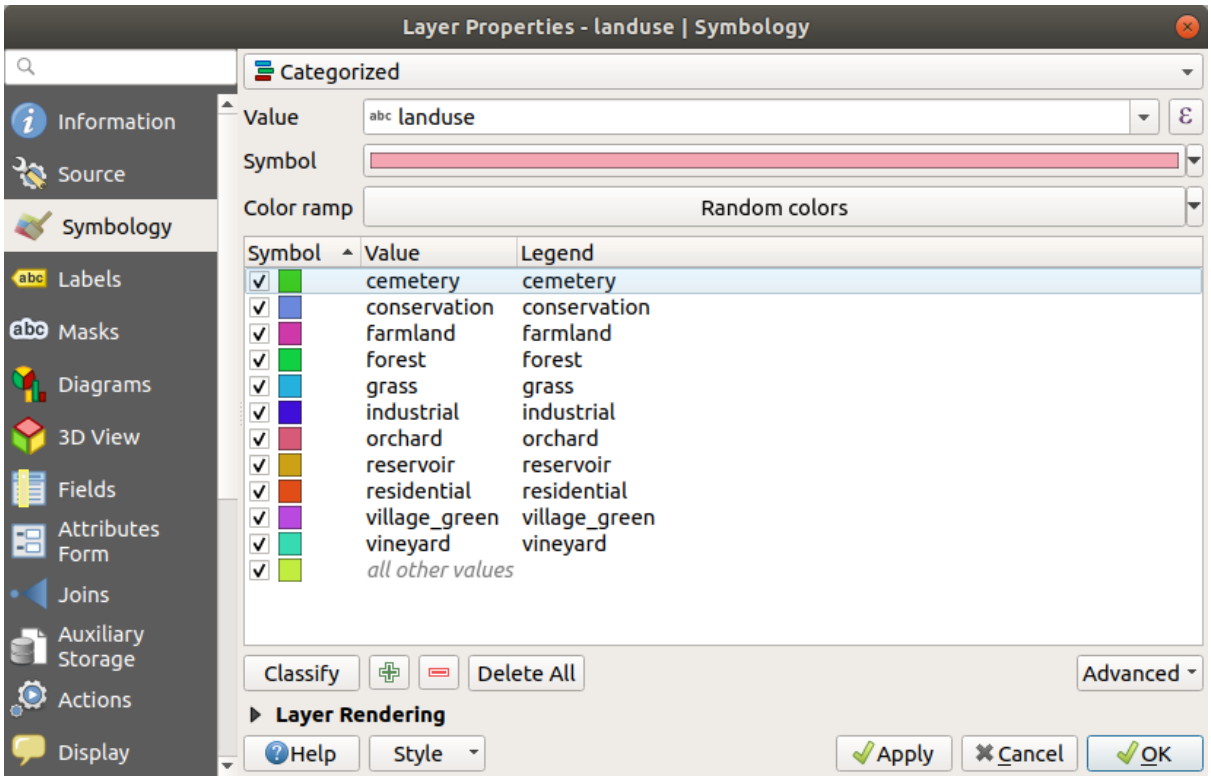
The goal for this lesson: To learn how to classify vector data effectively.

3.3.1 Follow Along: Classifying Nominal Data

1. Open the *Layer Properties* dialog for the `landuse` layer
2. Go to the *Symbology* tab
3. Click on the dropdown that says *Single Symbol* and change it to *Categorized*:

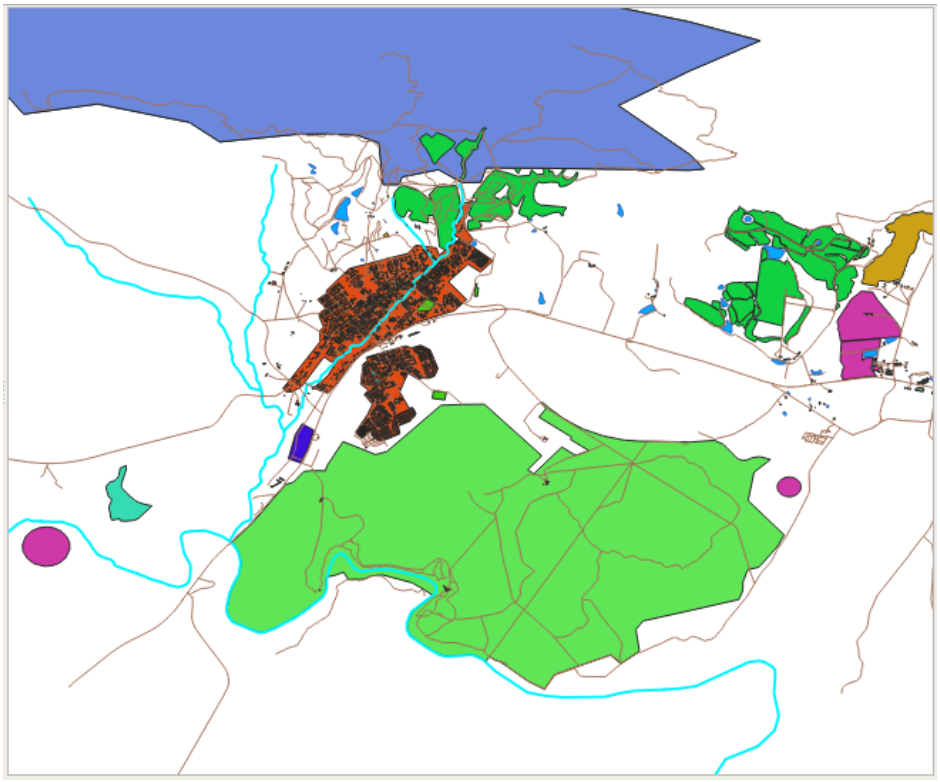


4. In the new panel, change the *Value* to `landuse` and the *Color ramp* to *Random colors*
5. Click the button labeled *Classify*

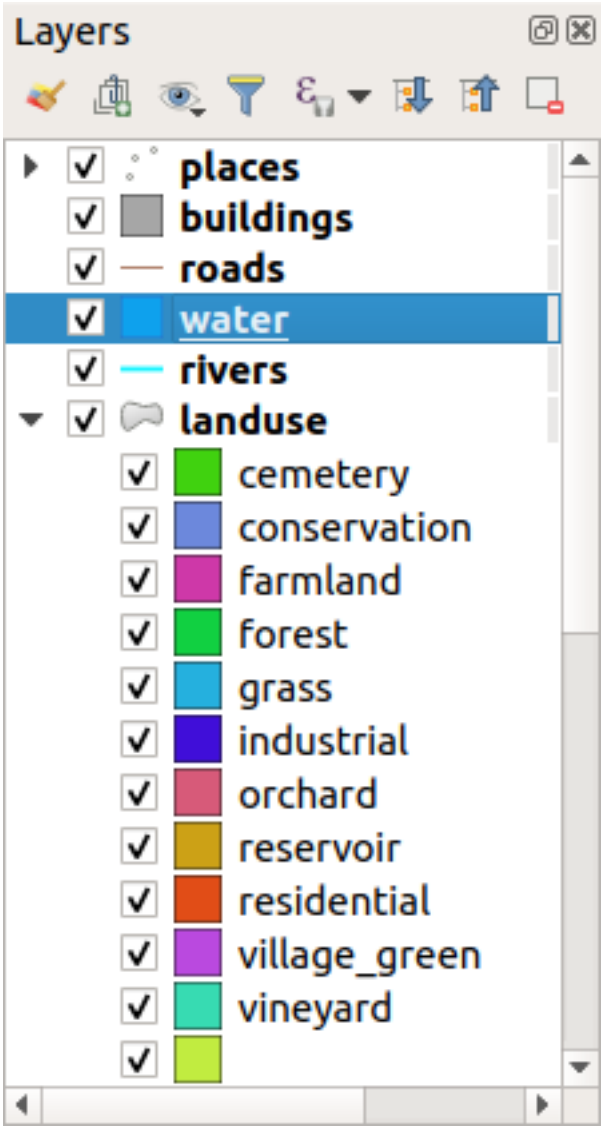


6. Klicken Sie auf *OK*

   You'll see something like this:

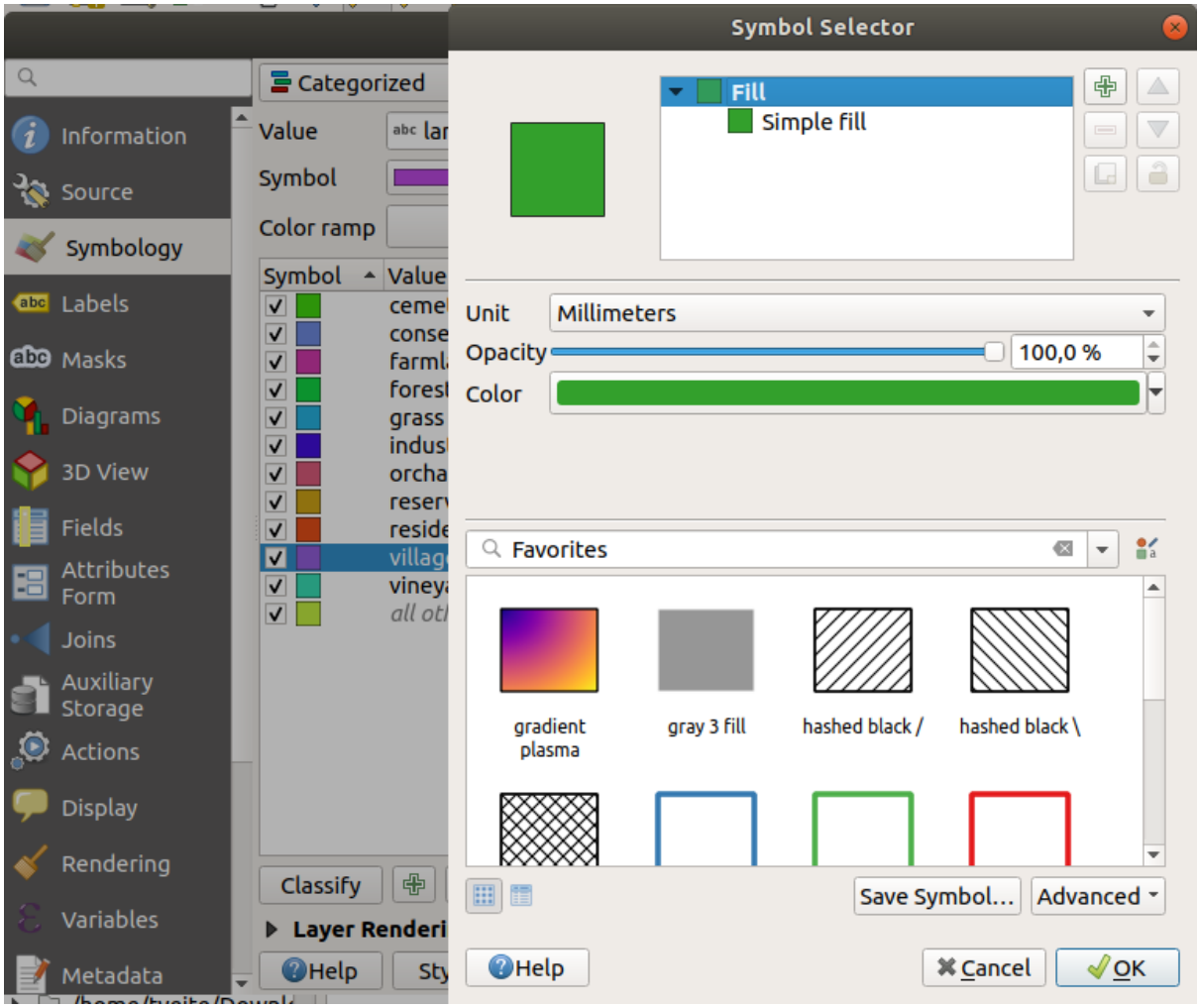


7. Click the arrow (or plus sign) next to `landuse` in the *Layers* panel, you'll see the categories explained:

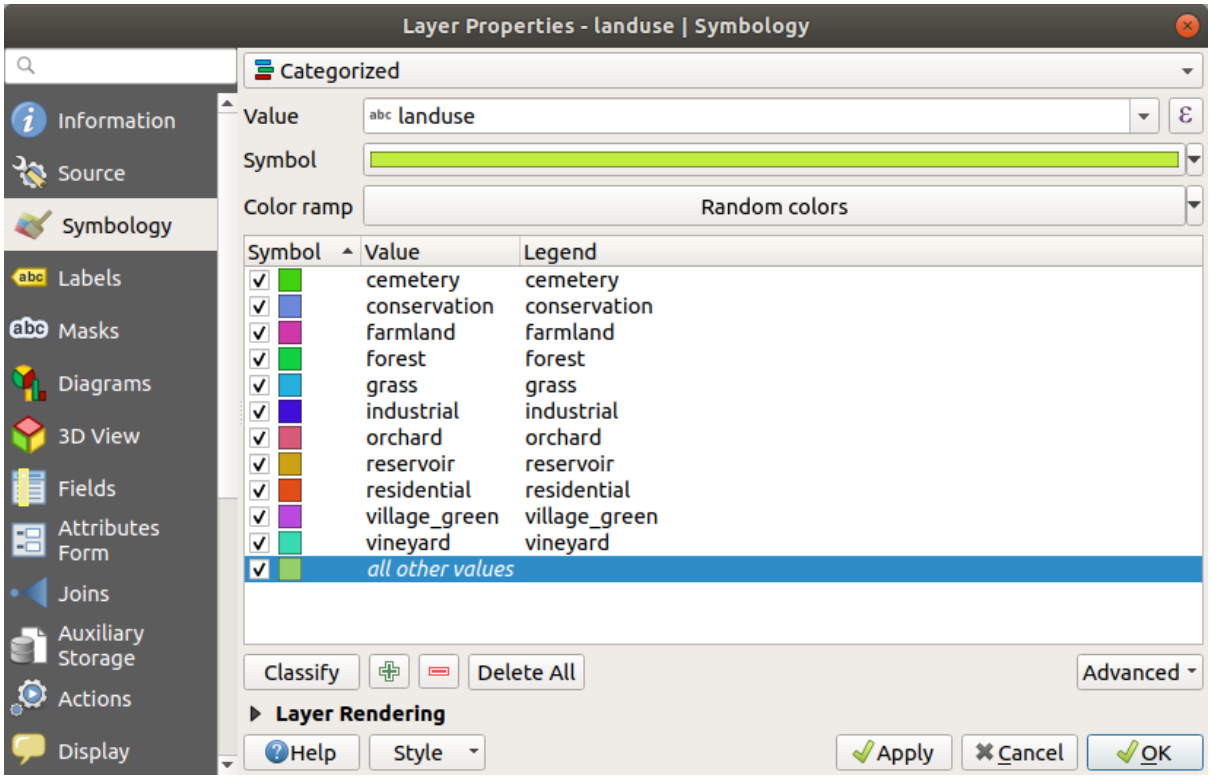


Now our landuse polygons are colored and are classified so that areas with the same land use are the same color.

8. If you wish to, you can change the symbol of each landuse area by double-clicking the relevant color block in the *Layers* panel or in the *Layer Properties* dialog:



Notice that there is one category that's empty:



This empty category is used to color any objects which do not have a landuse value defined or which have a *NULL* value. It can be useful to keep this empty category so that areas with a *NULL* value are still represented on the map. You may like to change the color to more obviously represent a blank or *NULL* value.

Remember to save your map now so that you don't lose all your hard-earned changes!

3.3.2 Try Yourself More Classification

Use the knowledge you gained above to classify the `buildings` layer. Set the categorisation against the `building` field and use the *Spectral* color ramp.

Bemerkung: Remember to zoom into an urban area to see the results.

3.3.3 Follow Along: Ratio Classification

There are four types of classification: *nominal*, *ordinal*, *interval* and *ratio*.

In **nominal** classification, the categories that objects are classified into are name-based; they have no order. For example: town names, district codes, etc. Symbols that are used for nominal data should not imply any order or magnitude.

- For points, we can use symbols of different shape.
- For polygons, we can use different types of hatching or different colours (avoid mixing light and dark colours).
- For lines, we can use different dash patterns, different colours (avoid mixing light and dark colours) and different symbols along the lines.

In **ordinal** classification, the categories are arranged in a certain order. For example, world cities are given a rank depending on their importance for world trade, travel, culture, etc. Symbols that are used for ordinal data should imply order, but not magnitude.

- For points, we can use symbols with light to dark colours.
- For polygons, we can use graduated colours (light to dark).
- For lines, we can use graduated colours (light to dark).

In **interval** classification, the numbers are on a scale with positive, negative and zero values. For example: height above/below sea level, temperature in degrees Celsius. Symbols that are used for interval data should imply order and magnitude.

- For points, we can use symbols with varying size (small to big).
- For polygons, we can use graduated colours (light to dark) or add diagrams of varying size.
- For lines, we can use thickness (thin to thick).

In **ratio** classification, the numbers are on a scale with only positive and zero values. For example: temperature above absolute zero (0 degrees Kelvin), distance from a point, the average amount of traffic on a given street per month, etc. Symbols that are used for ratio data should imply order and magnitude.

- For points, we can use symbols with varying size (small to big).
- For polygons, we can use graduated colours (light to dark) or add diagrams of varying size.
- For lines, we can use thickness (thin to thick).

In the example above, we used nominal classification to color each record in the `landuse` layer based on its `landuse` attribute. Now we will use ratio classification to classify the records by area.

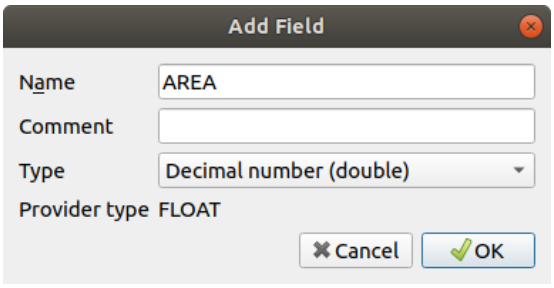
We are going to reclassify the layer, so existing classes will be lost if not saved. To store the current classification:

1. Open the layer's properties dialog
2. Click the *Save Style …* button in the *Style* drop-down menu.
3. Select *Rename Current…*, enter `land usage` and press *OK*.

   The categories and their symbols are now saved in the layer's properties.
4. Click now on the *Add…* entry of the *Style* drop-down menu and create a new style named `ratio`. This will store the new classification.
5. Close the *Layer Properties* dialog

We want to classify the landuse areas by size, but there is a problem: they don't have a size field, so we'll have to make one.

1. Open the Attributes Table for the `landuse` layer.
2. Enter edit mode by clicking the Toggle editing button
3. Add a new column of decimal type, called `AREA`, using the New field button:



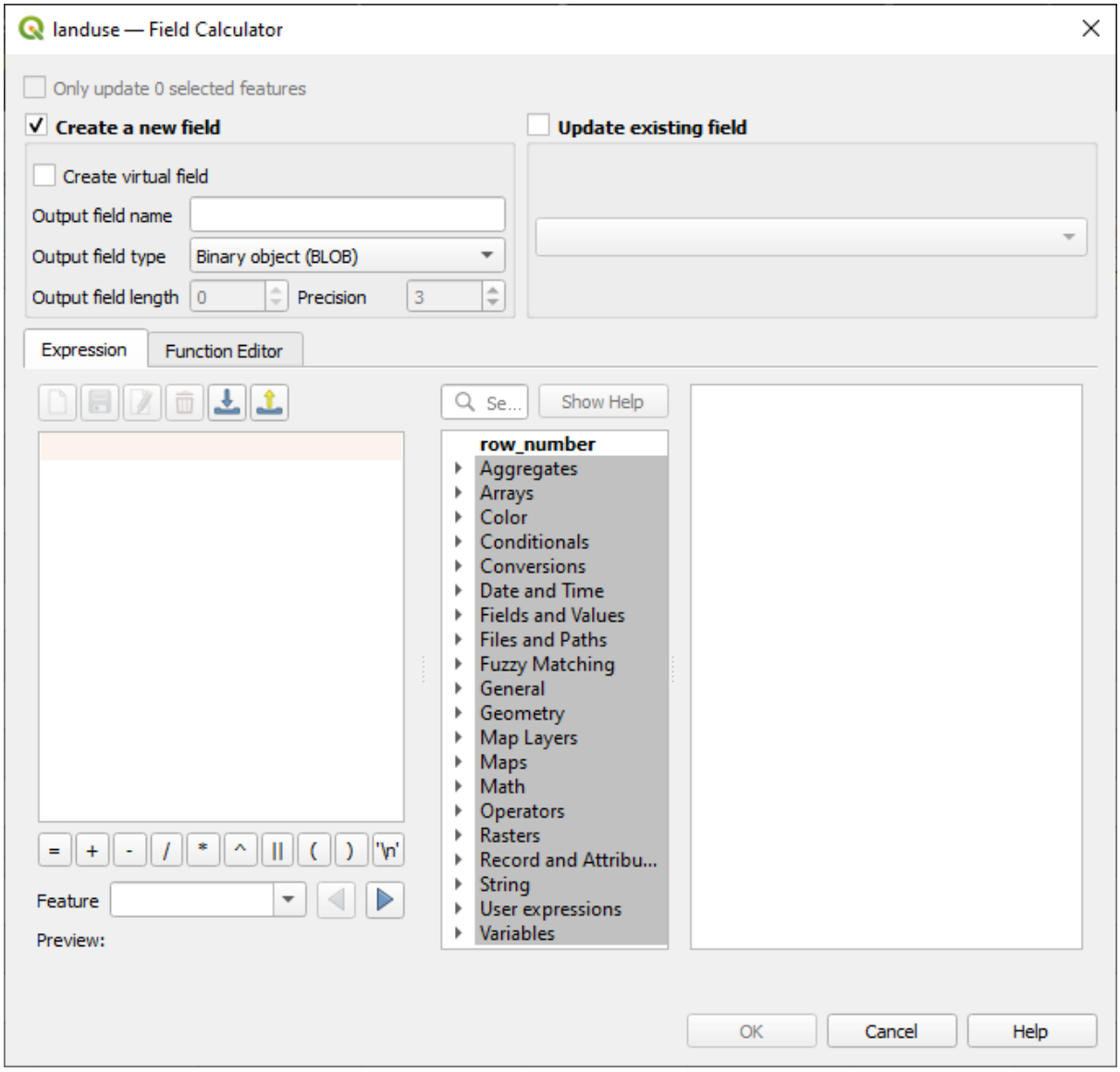
4. Klicken Sie auf *OK*

   The new field will be added (at the far right of the table; you may need to scroll horizontally to see it). However, at the moment it is not populated, it just has a lot of *NULL* values.

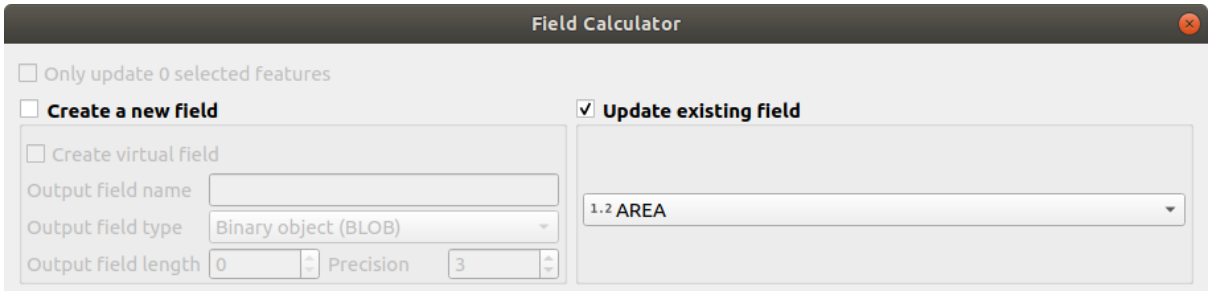
   To solve this problem, we will need to calculate the areas.

   1. Open the field calculator with the button.

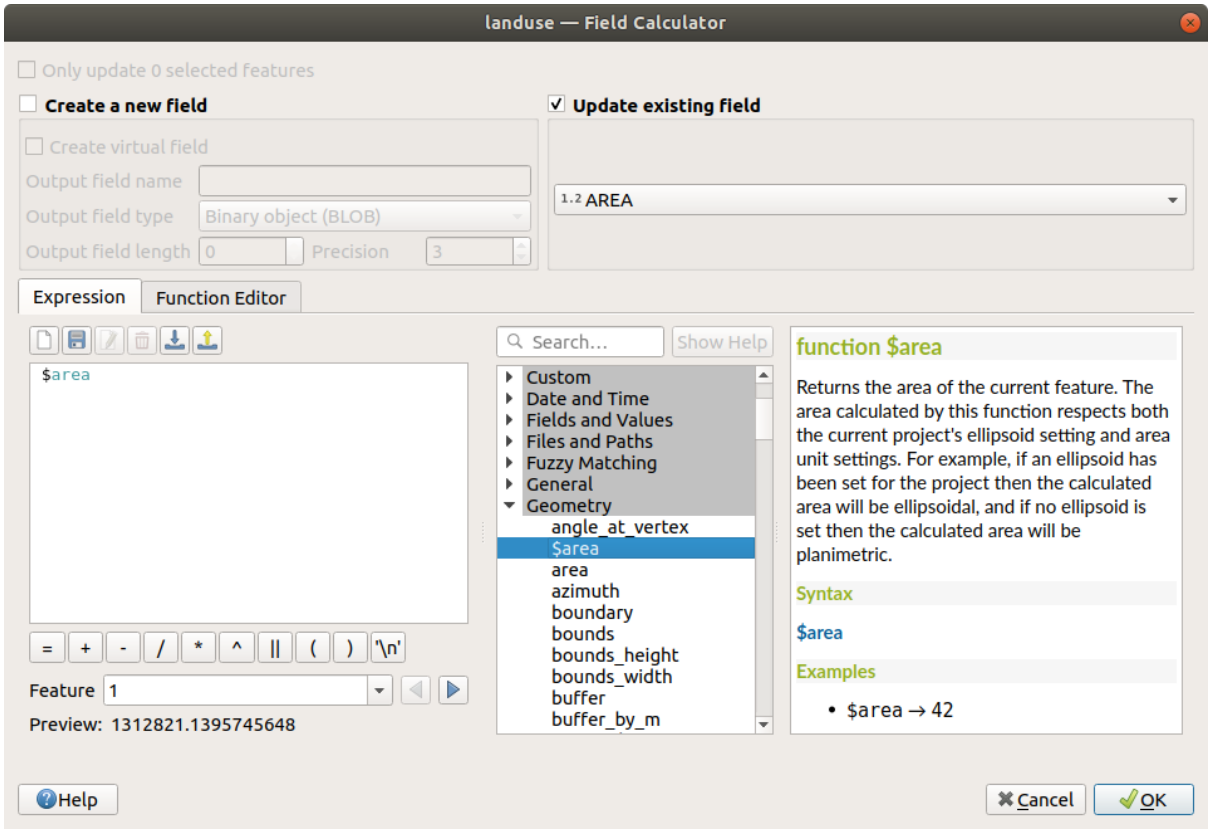
      You will get this dialog:



   2. Check the *Update existing fields*
   3. Select *AREA* in the fields drop-down menu



4. Under the *Expression* tab, expand the *Geometry* functions group in the list and find *$area*
5. Double-click on it so that it appears in the *Expression* field



6. Klicken Sie auf *OK*
7. Scroll to the `AREA` field in the attribute table and you will notice that it is populated with values (you may need to click the column header to refresh the data).

Bemerkung: These areas respect the project's area unit settings, so they may be in square meters or square degrees.

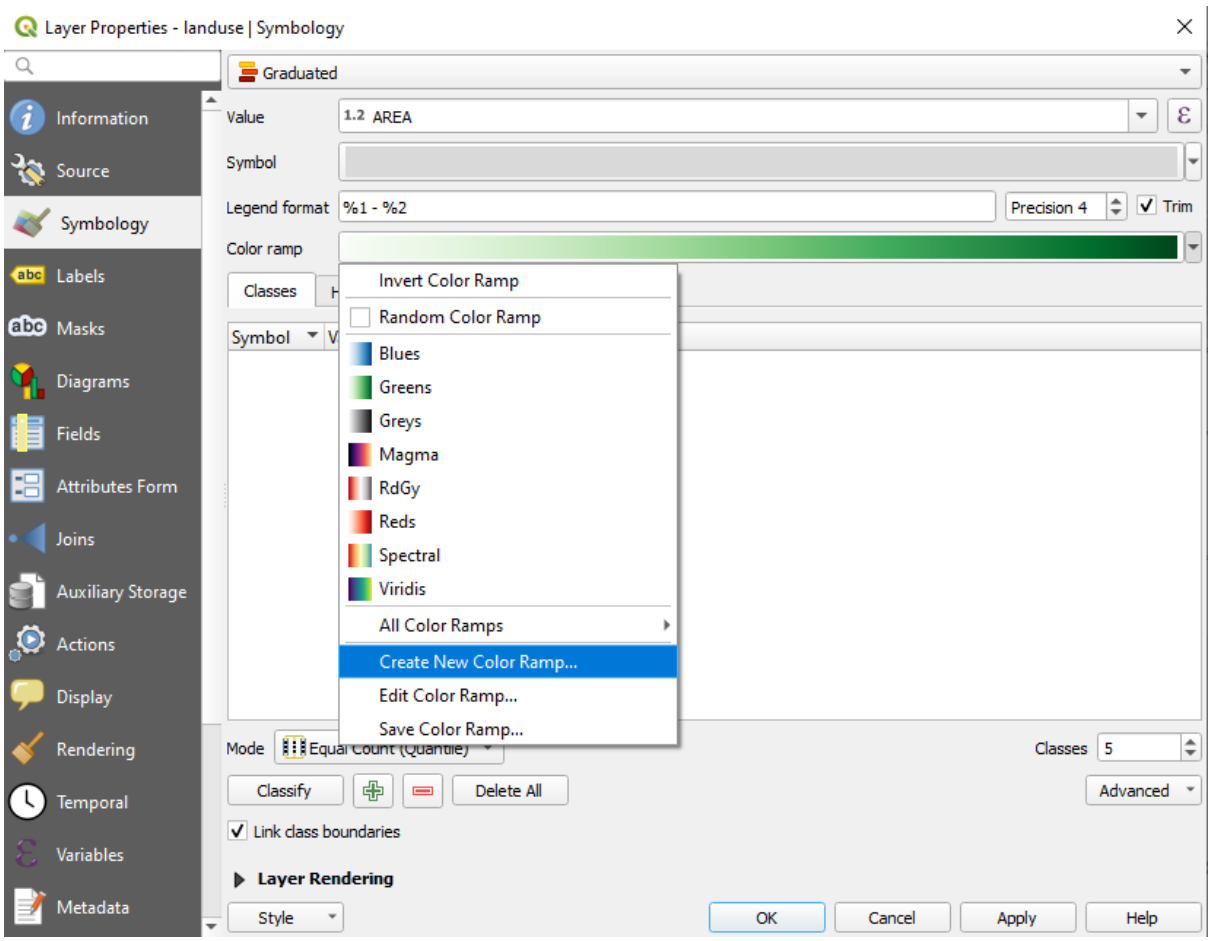
---

5. Press to save the edits and exit the edit mode with Toggle editing
6. Close the attribute table

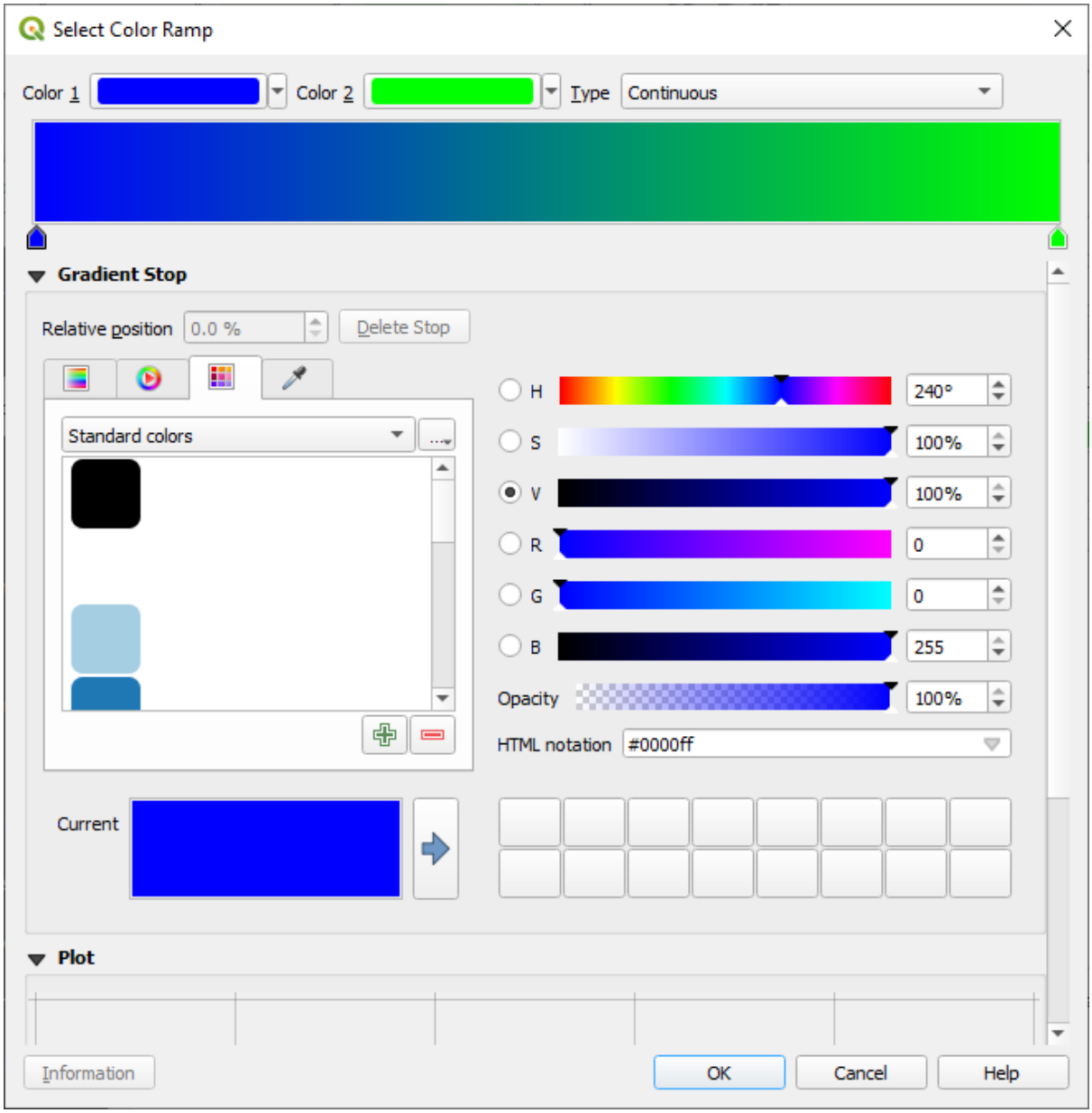
Now that we have the data, let's use them to render the `landuse` layer.

1. Open the *Layer properties* dialog's *Symbology* tab for the `landuse` layer
2. Change the classification style from *Categorized* to *Graduated*

3. Change the *Value* to `AREA`
4. Under *Color ramp*, choose the option *Create New Color Ramp…*:



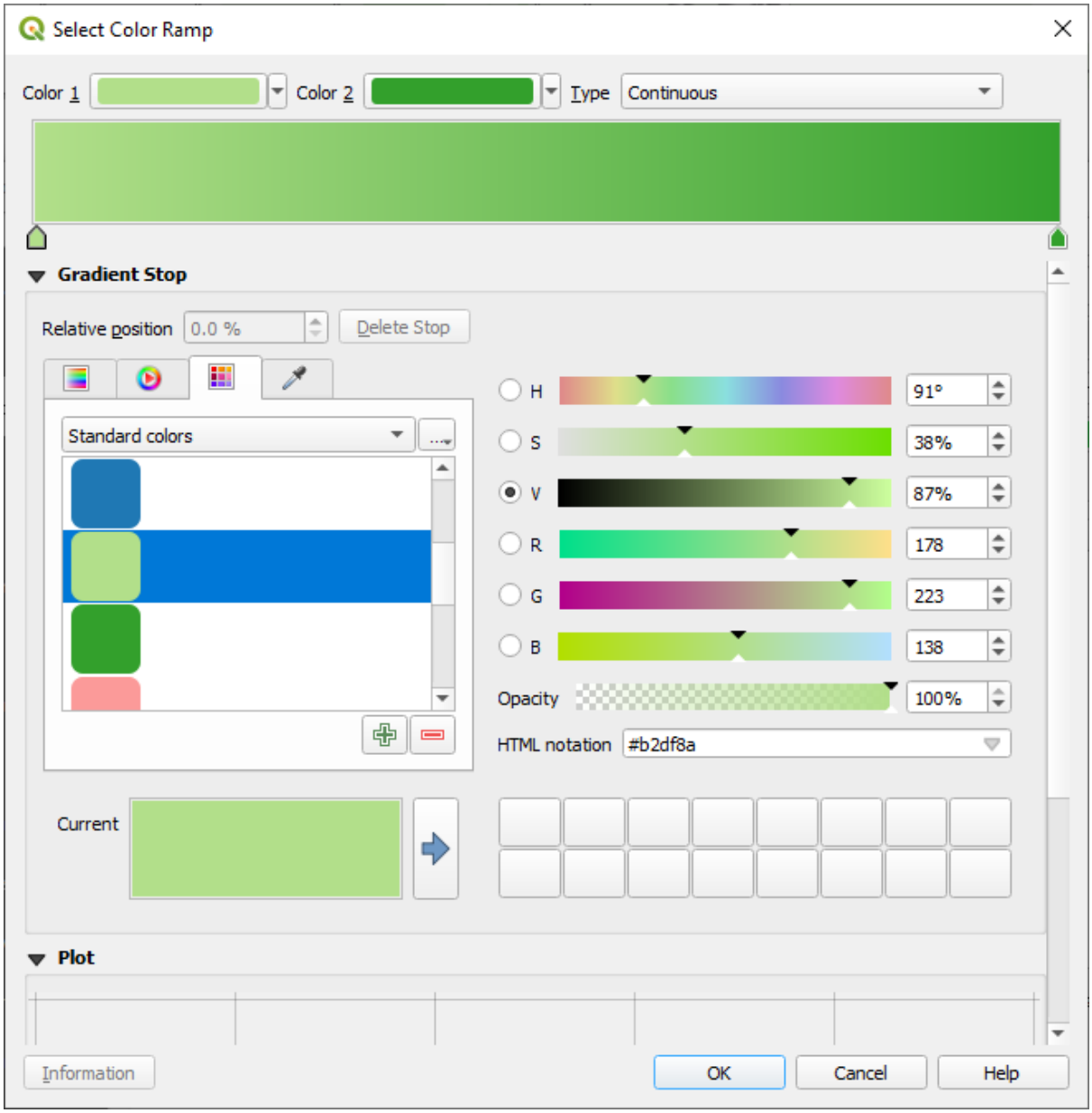
5. Choose *Gradient* (if it's not selected already) and click *OK*. You will see this:



You'll be using this to denote area, with small areas as *Color 1* and large areas as *Color 2*.

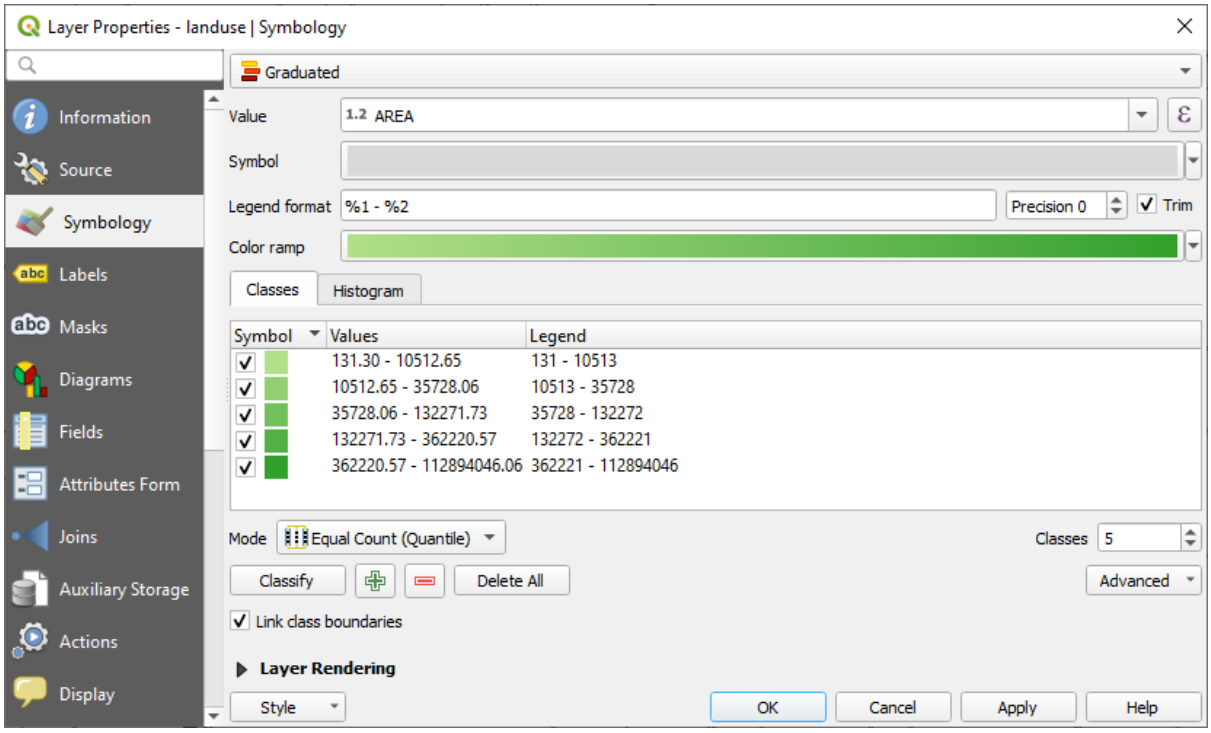
6. Choose appropriate colors

   In the example, the result looks like this:



7. Klicken Sie auf *OK*
8. You can save the colour ramp by selecting *Save Color Ramp…* under the *Color ramp* tab. Choose an appropriate name for the colour ramp and click *Save*. You will now be able to select the same colour ramp easily under *All Color Ramps*.
9. Under *Mode* choose Equal Count (Quantile) .
10. Click *Classify*

    Now you will have something like this:



Leave everything else as-is.

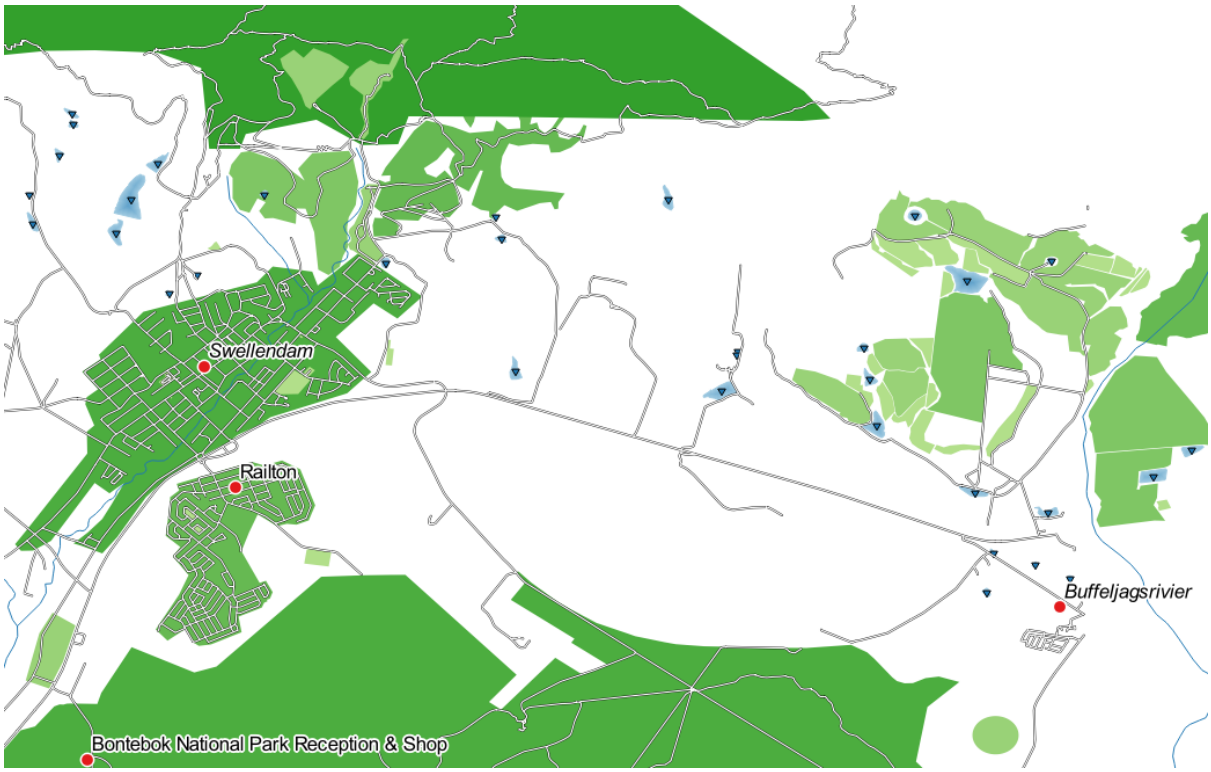
11. Click *OK*:

3.3.4 Try Yourself Refine the Classification

- Change the values of *Mode* and *Classes* until you get a classification that makes sense.

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:



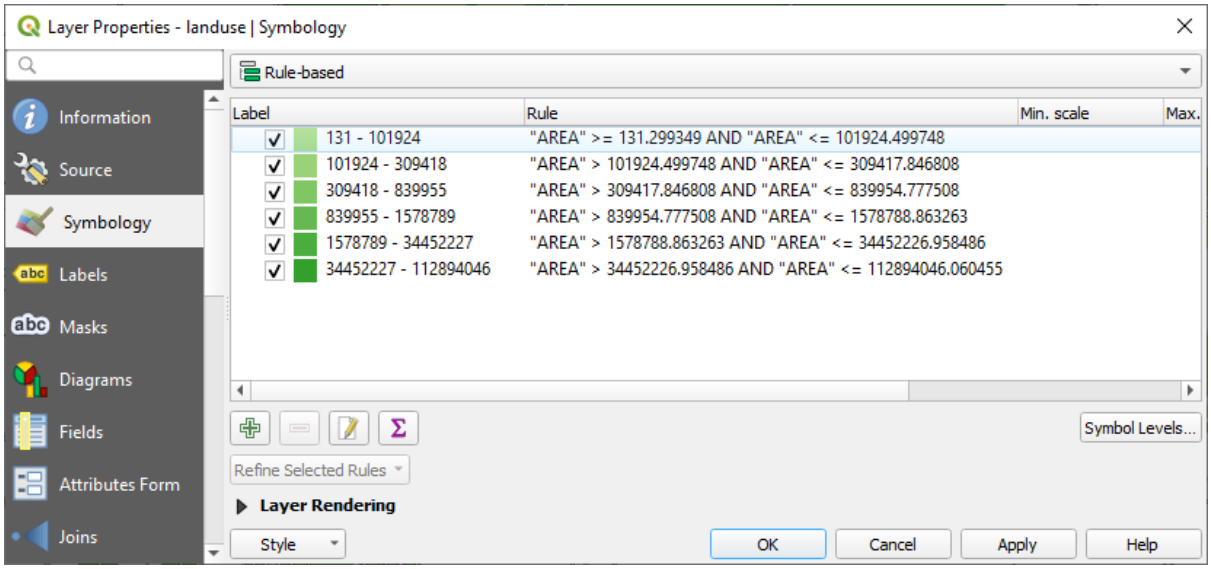
3.3.5 Follow Along: Rule-based Classification

It's often useful to combine multiple criteria for a classification, but unfortunately normal classification only takes one attribute into account. That's where rule-based classification comes in handy.

In this lesson, we will represent the `landuse` layer in a way to easily identify Swellendam city from the other residential area, and from the other types of landuse (based on their area).

1. Open the *Layer Properties* dialog for the `landuse` layer
2. Switch to the *Symbology* tab
3. Switch the classification style to *Rule-based*

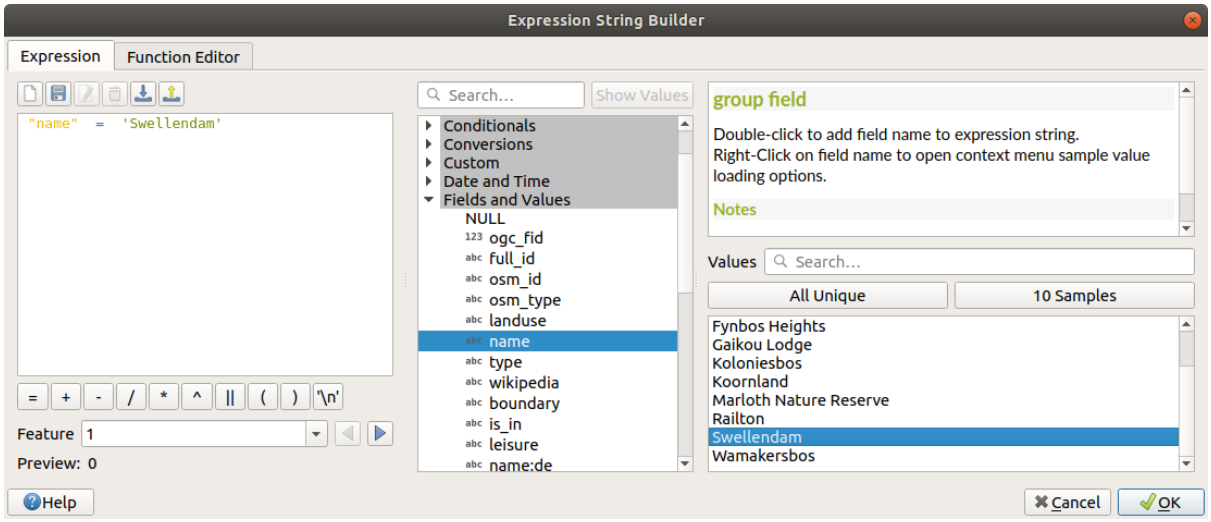
   QGIS will automatically show the rules that represent the current classification implemented for this layer. For example, after completing the exercise above, you may see something like this:



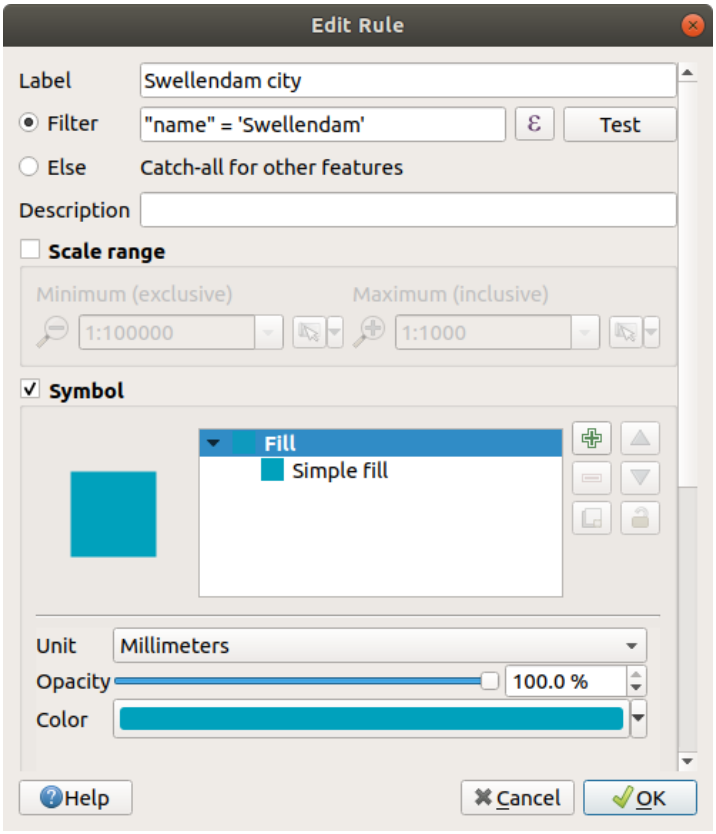
4. Click and drag to select all the rules
5. Use the Remove selected rules button to remove all of the existing rules

Let's now add our custom rules.

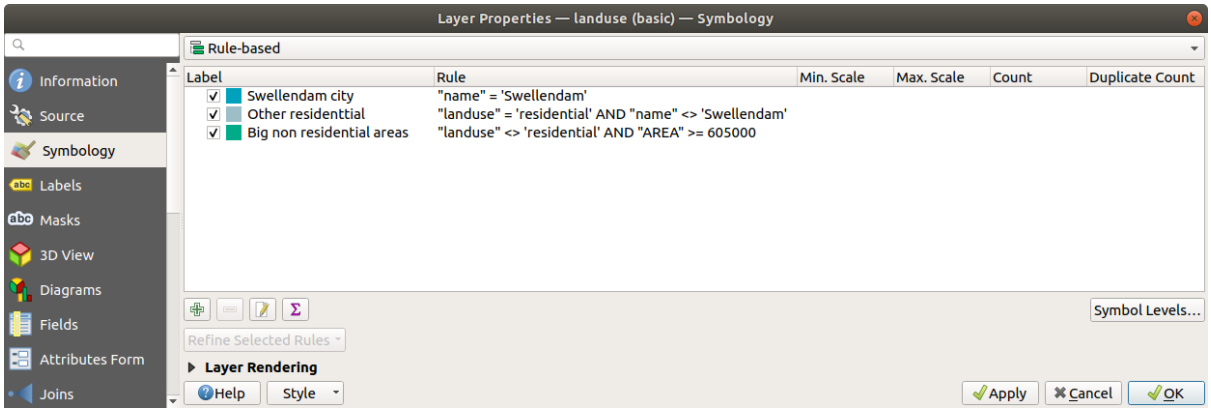
1. Click the Add rule button
2. The *Edit rule* dialog then appears
3. Enter `Swellendam city` as *Label*
4. Click the button next to the *Filter* text area to open the *Expression String Builder*
5. Enter the criterion `"name" = 'Swellendam'` and validate



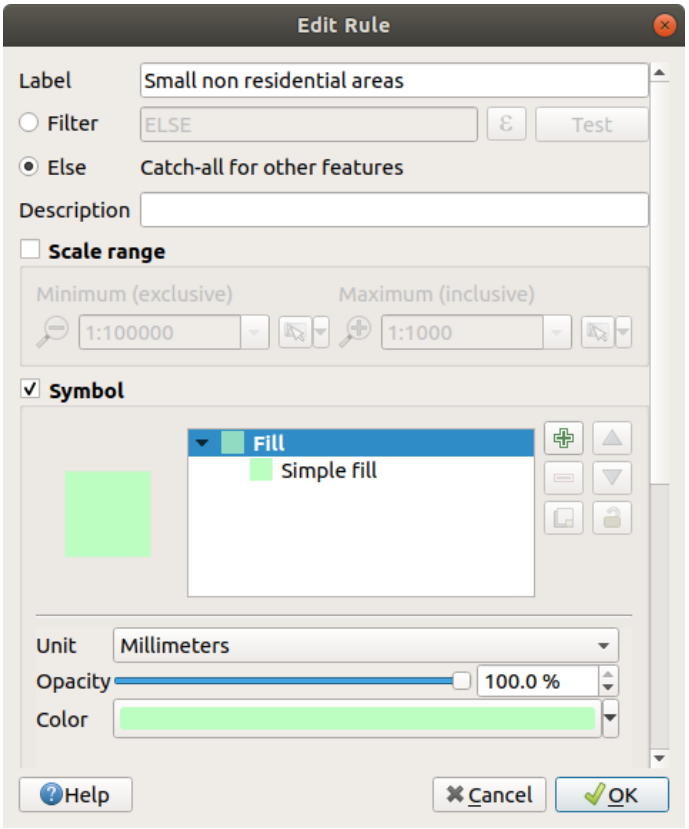
6. Back to the *Edit rule* dialog, assign it a darker grey-blue color in order to indicate the town's importance in the region and remove the border



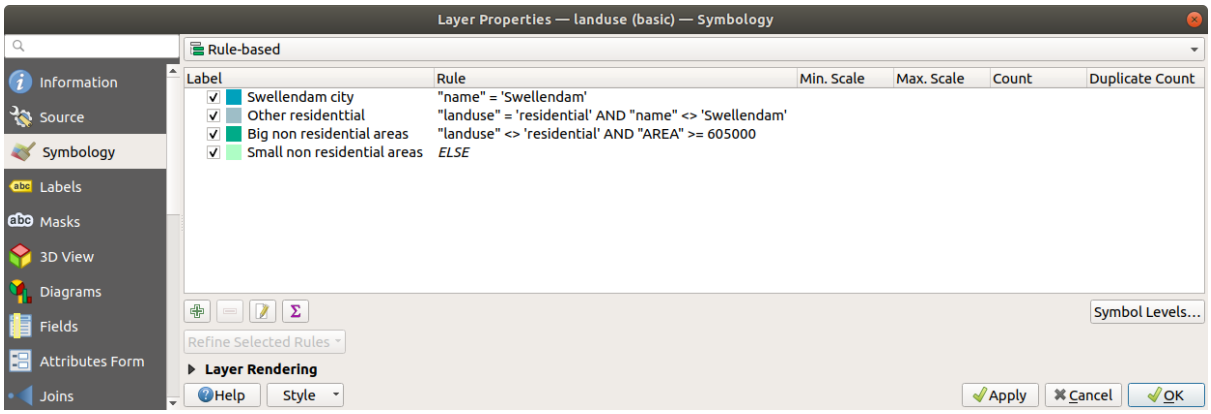
7. Klicken Sie auf *OK*
8. Repeat the steps above to add the following rules:
    1. **Other residential** label with the criterion `"landuse" = 'residential' AND "name" <> 'Swellendam'`. Choose a pale blue-grey *Fill color*
    2. **Big non residential areas** label with the criterion `"landuse" <> 'residential' AND "AREA" >= 605000`. Choose a mid-green color.



    These filters are exclusive, in that they exclude areas on the map (non-residential areas which are smaller than 605000 (square meters) are not included in any of the rules).
    3. We will catch the remaining features using a new rule labeled **Small non residential areas**. Instead of a filter expression, Check the *Else*. Give this category a suitable pale green color.



Your rules should now look like this:



9. Apply this symbology

Your map will look something like this:

Now you have a map with Swellendam the most prominent residential area and other non-residential areas colored according to their size.

3.3.6 In Conclusion

Symbology allows us to represent the attributes of a layer in an easy-to-read way. It allows us as well as the map reader to understand the significance of features, using any relevant attributes that we choose. Depending on the problems you face, you'll apply different classification techniques to solve them.

3.3.7 What's Next?

Now we have a nice-looking map, but how are we going to get it out of QGIS and into a format we can print out, or make into an image or PDF? That's the topic of the next lesson!

KAPITEL 4

Module: Layout der Karten

In diesem Modul lernen Sie, wie Sie das QGIS-Drucklayout verwenden, um qualitativ hochwertige Karten mit allen erforderlichen Kartenkomponenten zu erstellen.

4.1 Lesson: Using Print Layout

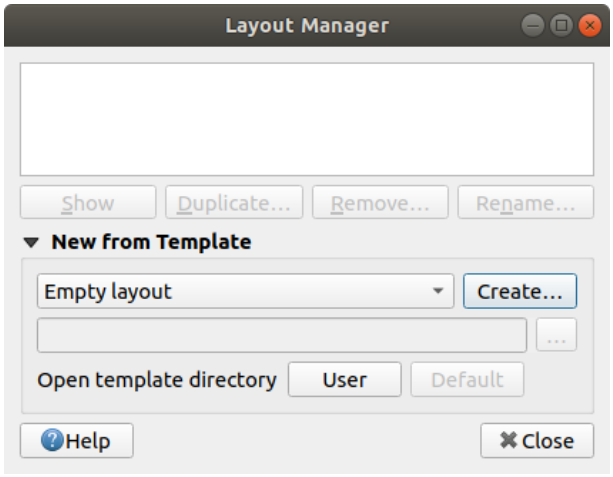
Now that you've got a map, you need to be able to print it or to export it to a document. The reason is, a GIS map file is not an image. Rather, it saves the state of the GIS program, with references to all the layers, their labels,colors, etc. So for someone who doesn't have the data or the same GIS program (such as QGIS), the map file will be useless. Luckily, QGIS can export its map file to a format that anyone's computer can read, as well as printing out the map if you have a printer connected. Both exporting and printing is handled via the *Print Layout*.

The goal for this lesson: To use the QGIS *Print Layout* to create a basic map with all the required settings.

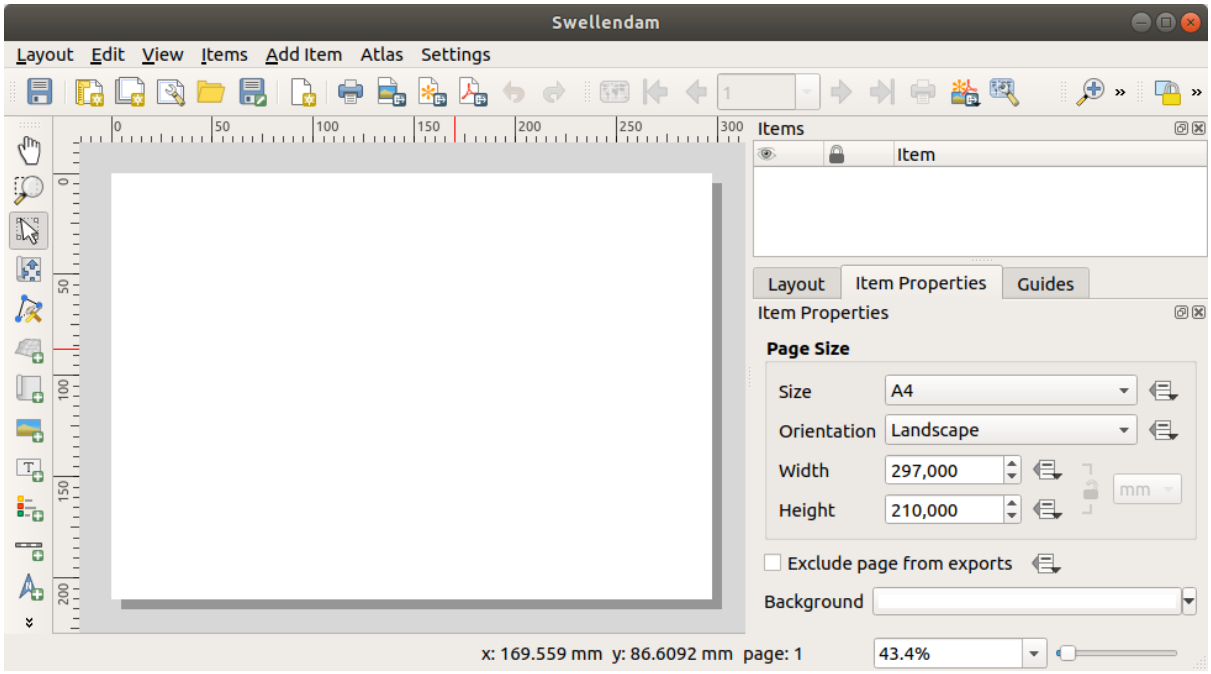
4.1.1 Follow Along: The Layout Manager

QGIS allows you to create multiple maps using the same map file. For this reason, it has a tool called the *Layout Manager*.

1. Click on the *Project* ► *Layout Manager…* menu entry to open this tool. You'll see a blank *Layout manager* dialog appear.

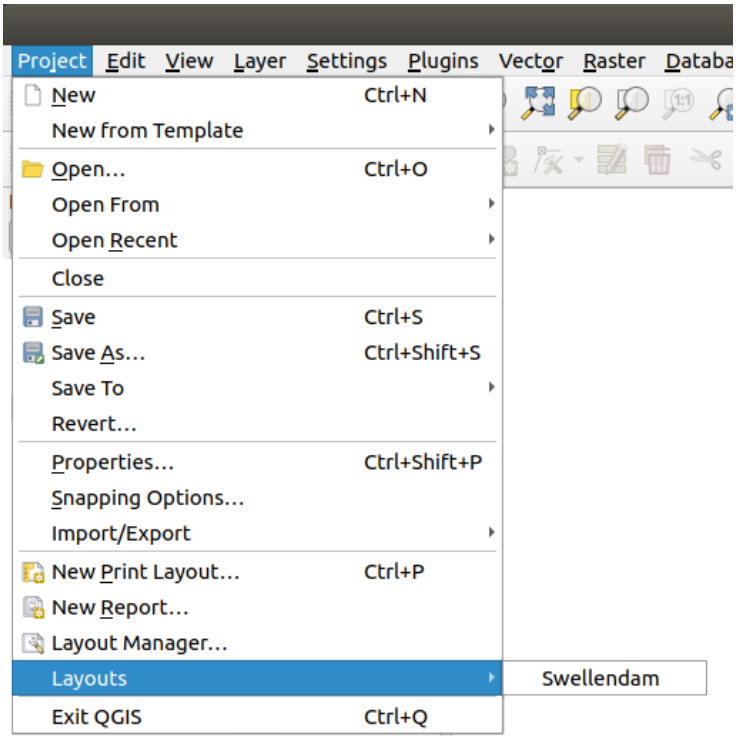


2. Under *New from Template*, select *Empty layout* and press the *Create…* button.
3. Give the new layout the name of Swellendam and click *OK*.
4. You will now see the *Print Layout* window:



You could also create this new layout via the *Project* ► *New Print Layout…* menu.

Whichever route you take, the new print layout is now accessible from the *Project* ► *Layouts* ► menu, as in the image below.



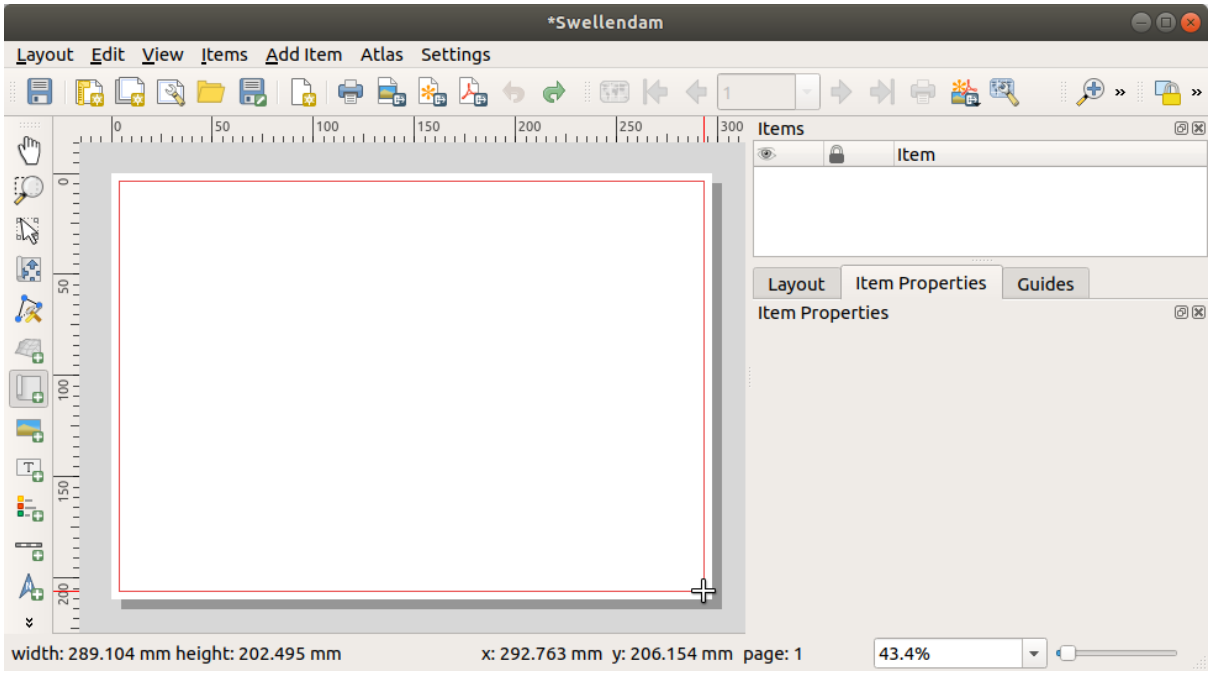
4.1.2 Follow Along: Basic Map Composition

In this example, the composition was already the way we wanted it. Ensure that yours is as well.

1. Right-click on the sheet in the central part of the layout window and choose *Page properties...* in the context menu.
2. Check that the values in the *Item Properties* tab are set to the following:
   - *Size*: `A4`
   - *Orientation*: `Landscape`

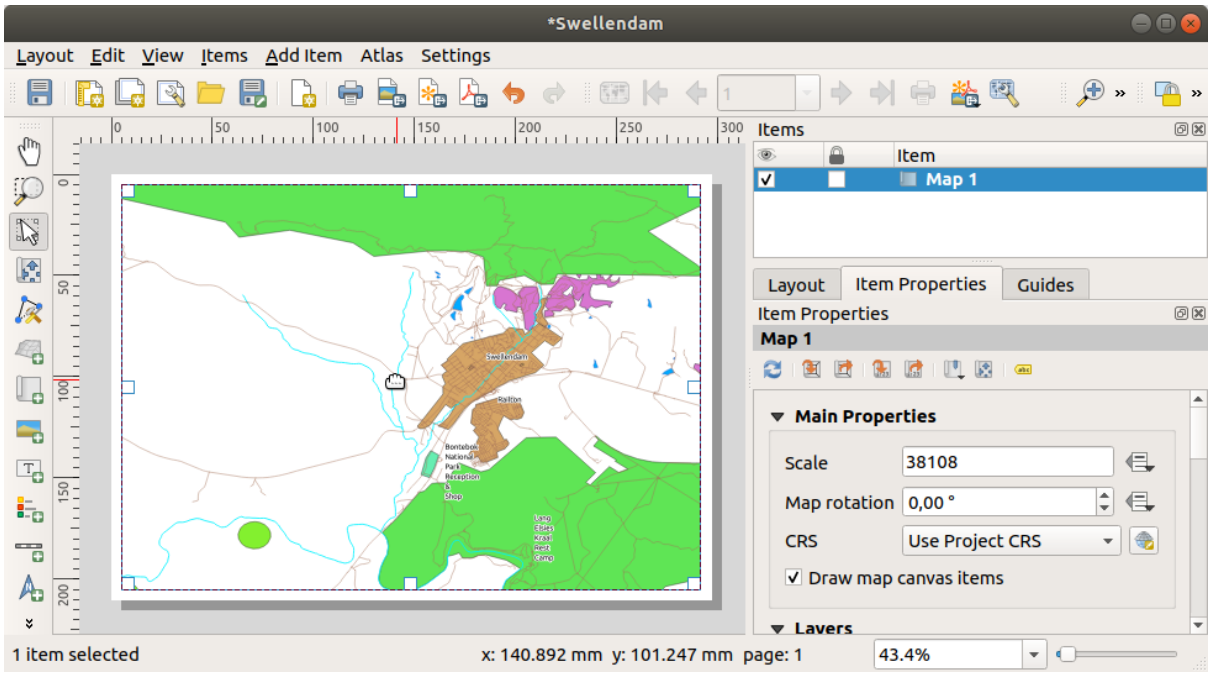
   Now you've got the page layout the way you wanted it, but this page is still blank. It clearly lacks a map. Let's fix that!
3. Click on the Add Map button.

   With this tool activated, you will be able to place a map on the page.
4. Click and drag a box on the blank page:

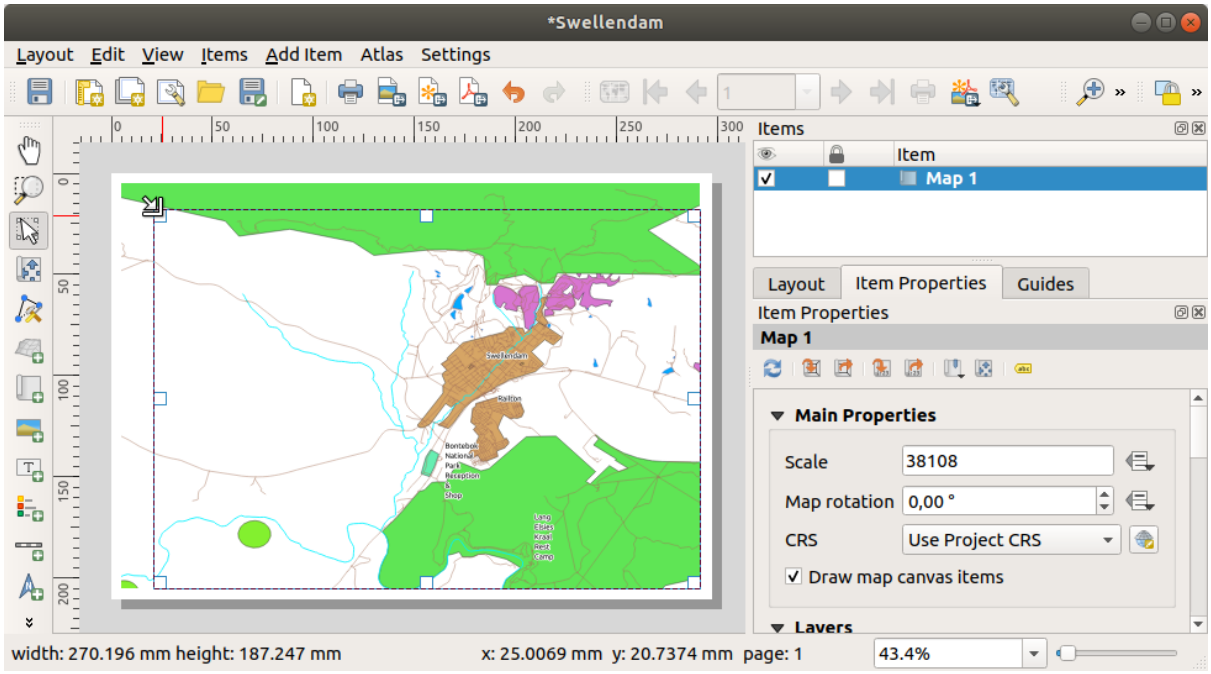


The map will appear on the page.

5. Move the map by clicking and dragging it around:



6. Resize it by clicking and dragging the boxes on the edges:



Bemerkung: Your map may look a lot different, of course! This depends on how your own project is set up. But not to worry! These instructions are general, so they will work the same regardless of what the map itself looks like.

7. Be sure to leave margins along the edges, and a space along the top for the title.
8. Zoom in and out on the page (but not the map!) by using these buttons:

9. Zoom and pan the map in the main QGIS window. You can also pan the map using the Move item content tool.

   The map view updates as you zoom in or zoom out.

10. If, for any reason, the map view does not refresh correctly, you can force the map to refresh by clicking the Refresh view button.

    Remember that the size and position you've given the map doesn't need to be final. You can always come back and change it later if you're not satisfied. For now, you need to ensure that you've saved your work on this map. Because a *Print Layout* in QGIS is part of the main map file, you must save your project.

11. Go to the *Layout* ► *Save Project*. This is a convenient shortcut to the one in the main dialog.

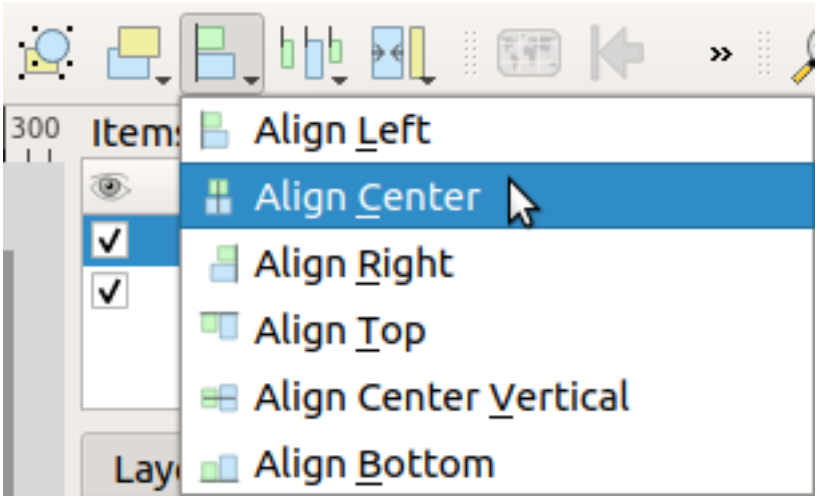
4.1.3 Follow Along: Adding a Title

Now your map is looking good on the page, but your readers/users are not being told what's going on yet. They need some context, which is what you'll provide for them by adding map elements. First, let us add a title.

1. Click on the Add Label button
2. Click on the page, above the map, accept the suggested values in the *New Item Properties* dialog, and a label will appear at the top of the map.
3. Resize it and place it in the top center of the page. It can be resized and moved in the same way that you resized and moved the map.

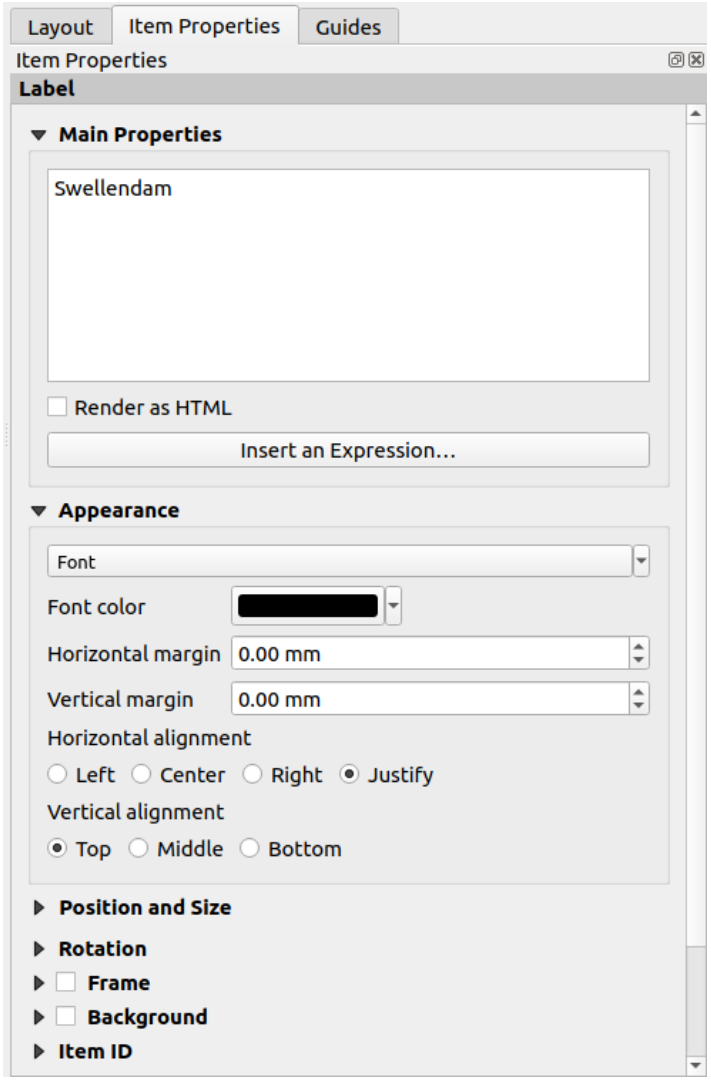
As you move the title, you'll notice that guidelines appear to help you position the title in the center of the page. However, there is also a tool in the Actions Toolbar to help position the title relative to the map (not the page):

4. Click the map to select it
5. Hold in `Shift` on your keyboard and click on the label so that both the map and the label are selected.
6. Look for the Align selected items left button and click on the dropdown arrow next to it to reveal the positioning options and click *Align center*:



Now the label frame is centered on the map, but not the contents. To center the contents of the label:

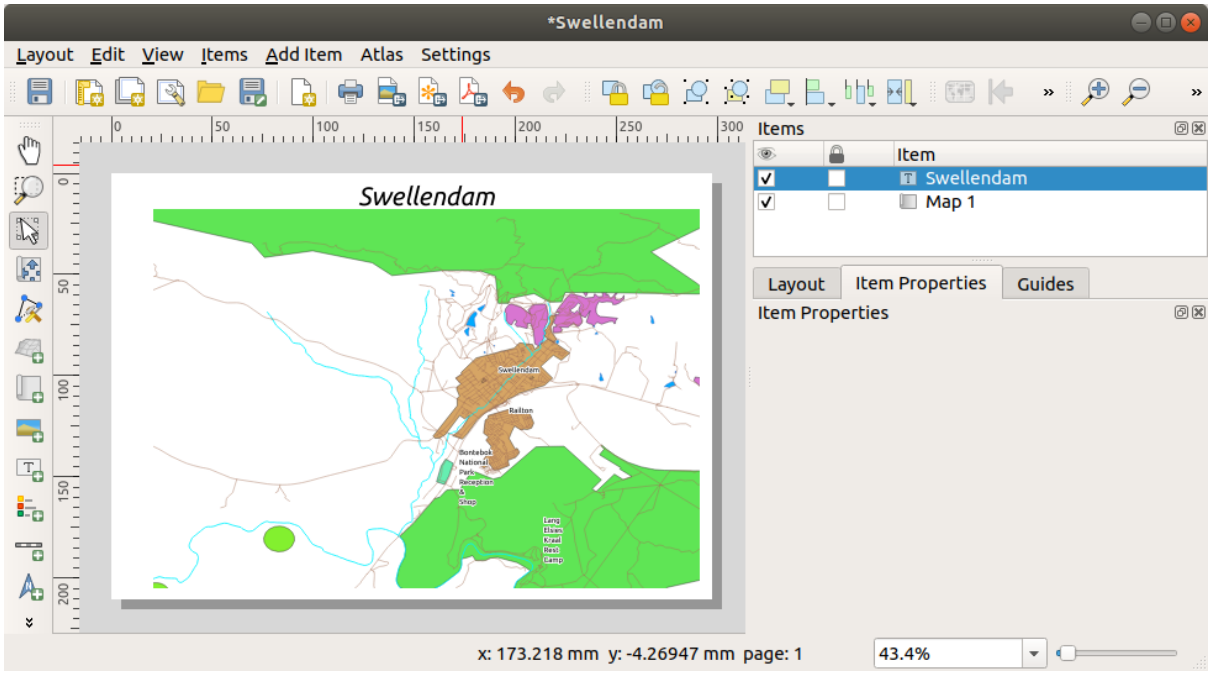
1. Select the label by clicking on it.
2. Click on the *Item Properties* tab in the side panel of the layout window.
3. Change the text of the label to „Swellendam“:



4. Use this interface to set the font and alignment options under the *Appearance* section:
    1. Choose a large but sensible font (the example will use the default font with a size of `36`)
    2. Set the *Horizontal Alignment* to *Center*.

    You can also change the font color, but it's probably best to keep it black as per the default.
5. The default setting is not to add a frame to the title's text box. However, if you wish to add a frame, you can do so:
    1. In the *Item Properties* tab, scroll down until you see the *Frame* option.
    2. Click the *Frame* checkbox to enable the frame. You can also change the frame's color and width.

In this example, we won't enable the frame, so here is our page so far:



To make sure that you don't accidentally move these elements around now that you've aligned them, you can lock items into place:

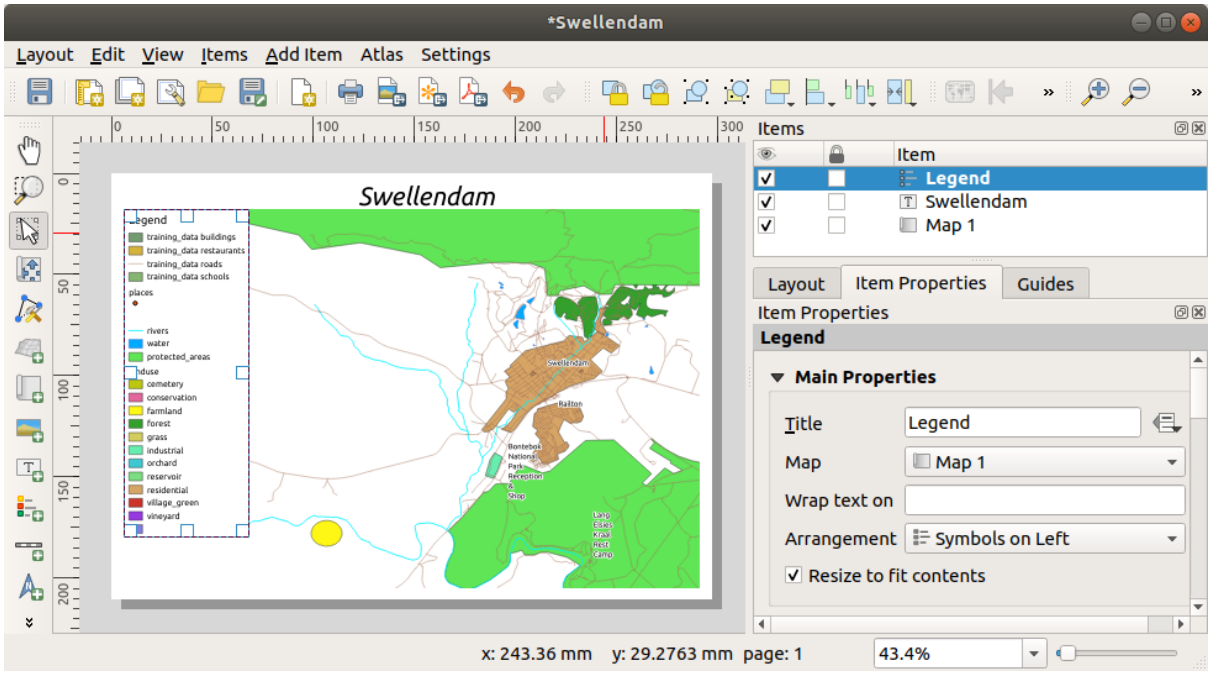
1. Select both the label and the map items
2. Click the Lock Selected Items button in the *Actions* Toolbar.

Bemerkung: Click the Unlock All Items button in the *Actions* Toolbar to be able to edit the items again.

4.1.4 Follow Along: Adding a Legend

The map reader also needs to be able to see what various things on the map actually mean. In some cases, like the place names, this is quite obvious. In other cases, it's more difficult to guess, like the colors of the forests. Let's add a new legend.

1. Click on the Add Legend button
2. Click on the page to place the legend, accept the suggested values in the *New Item Properties* dialog,
3. A legend is added to the layout page, showing layers symbology as set in the main dialog.
4. As usual, you can click and move the item to where you want it:



4.1.5 Follow Along: Customizing Legend Items

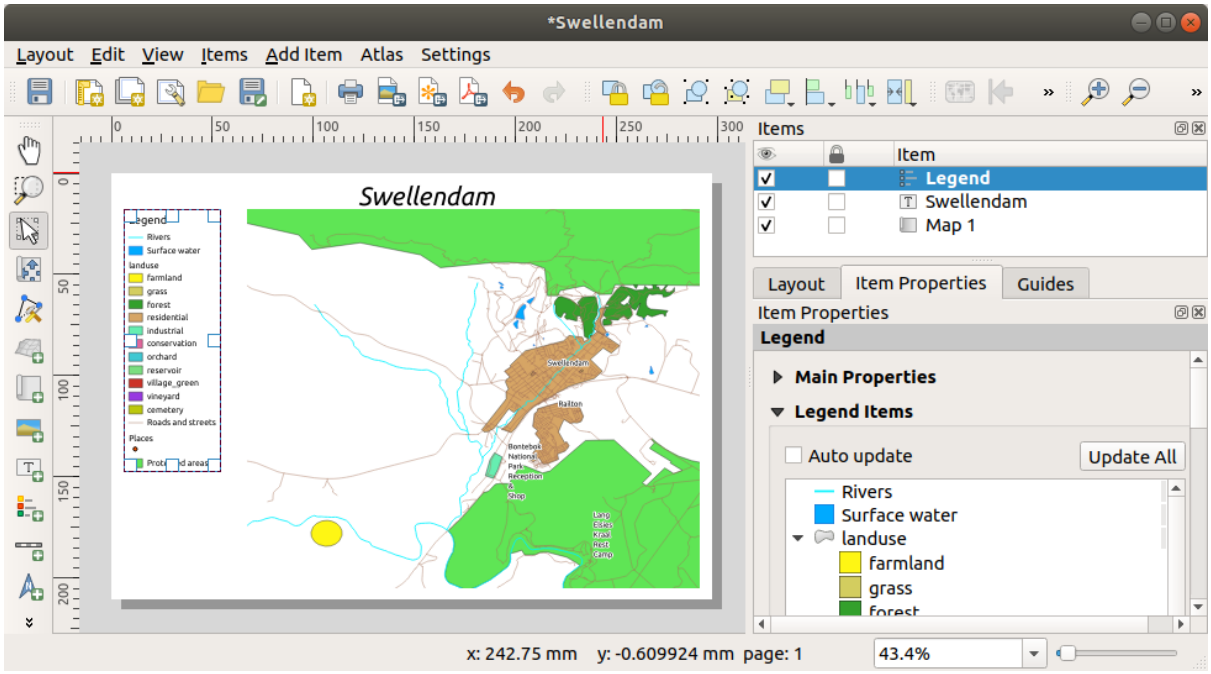
Not everything on the legend is necessary, so let's remove some unwanted items.

1. In the *Item Properties* tab, you'll find the *Legend items* group.
2. Uncheck the *Auto update* box, allowing you to directly modify the legend items
3. Select the entry with *buildings*
4. Delete it from the legend by clicking the button

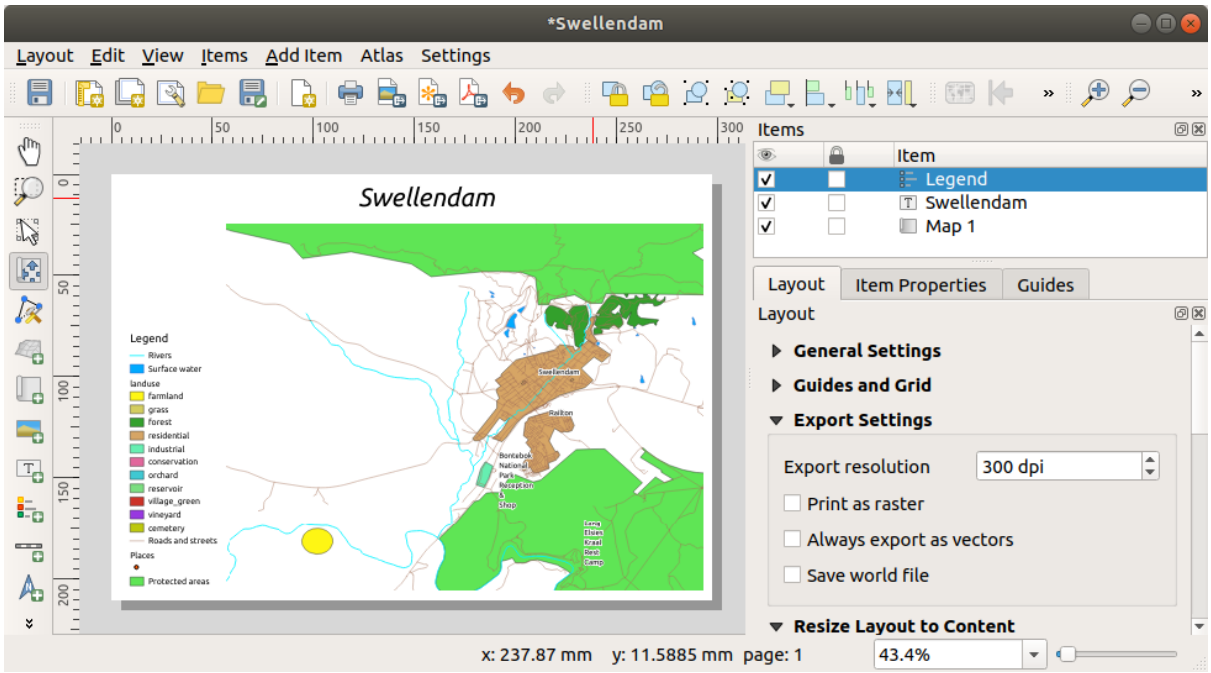
You can also rename items.

1. Select a layer from the same list.
2. Click the Edit selected item properties button.
3. Rename the layers to `Places`, `Roads and Streets`, `Surface Water`, and `Rivers`.

You can also reorder the items:



As the legend will likely be widened by the new layer names, you may wish to move and resize the legend and or map. This is the result:



4.1.6 Follow Along: Exporting Your Map

Bemerkung: Did you remember to save your work often?

Finally the map is ready for export! You'll see the export buttons near the top left corner of the layout window:

- Print Layout: interfaces with a printer. Since the printer options will differ depending on the model of printer that you're working with, it's probably better to consult the printer manual or a general guide to printing for more information on this topic.

  The other buttons allow you to export the map page to a file.

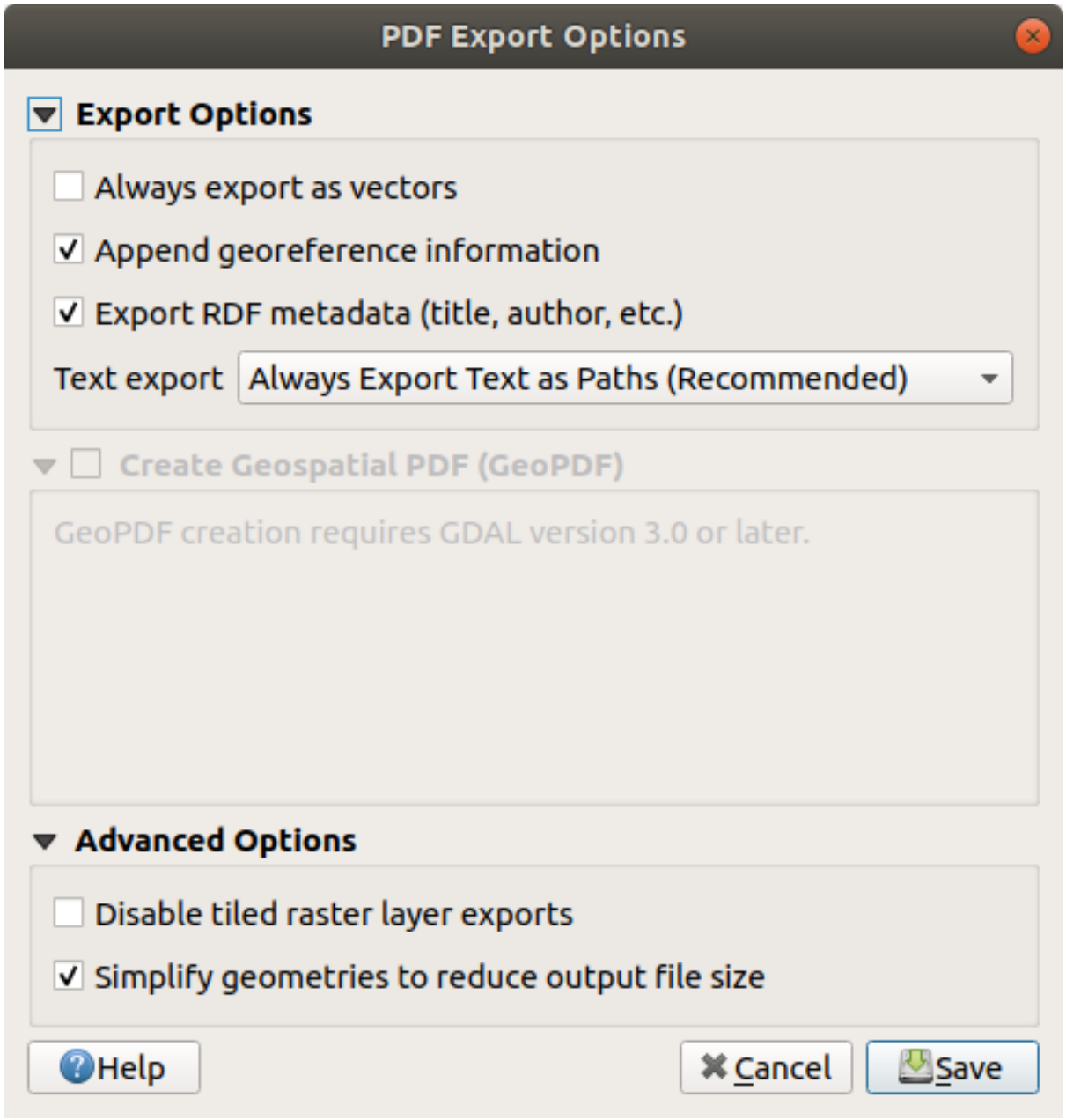
- Export as Image: gives you a selection of various common image formats to choose from. This is probably the simplest option, but the image it creates is „dead" and difficult to edit.

- Export as SVG: If you're sending the map to a cartographer (who may want to edit the map for publication), it's best to export as an SVG. SVG stands for „Scalable Vector Graphic", and can be imported to programs like Inkscape or other vector image editing software.

- Export as PDF: If you need to send the map to a client, it's most common to use a PDF, because it's easier to set up printing options for a PDF. Some cartographers may prefer PDF as well, if they have a program that allows them to import and edit this format.

For our purposes, we're going to use PDF.

1. Click the Export as PDF button
2. Choose a save location and a file name as usual. The following dialog will show up.



3. You can safely use the default values now and click *Save*.

   QGIS will proceed to the map export and push a message on top of the print layout dialog as soon as it finishes.

4. Click the hyperlink in the message to open the folder in which the PDF has been saved in your system's file manager
5. Open it and see how your layout looks.

   Everything is OK? Congratulations on your first completed QGIS map project!

6. Anything unsatisfying? Go back to the QGIS window, do the appropriate modifications and export again.
7. Remember to save your project file.

4.1.7 In Conclusion

Now you know how to create a basic static map layout. We can go a step further and create a map layout that adapts dynamically, with more layout items.

4.2 Lesson: Creating a Dynamic Print Layout

Now that you have learned to create a basic map layout we go a step further and create a map layout that adapts dynamically to our map extent and to the page properties, e.g. when you change the size of the page. Also, the date of creation will adapt dynamically.

4.2.1 Follow Along: Creating the dynamic map canvas

1. Load the ESRI Shapefile format datasets `protected_areas.shp`, `places.shp`, `rivers.shp` and `water.shp` into the map canvas and adapt its properties to suit your own convenience.
2. After everything is rendered and symbolized to your liking, click the New Print Layout icon in the *Project* toolbar or choose *Project* ► *New Print Layout*. You will be prompted to choose a title for the new print layout.
3. We want to create a map layout consisting of a header and a map of the region near Swellendam, South Africa. The layout should have a margin of 7.5 mm and the header should be 36mm high.
4. Create a map item called `main map` on the canvas and go to the *Layout* panel. Scroll down to the *Variables* section and find the *Layout* part. Here we set some variables you can use all over the dynamic print layout. Go to the *Layout* panel and scroll down to the *Variables* section. The first variable will define the margin. Press the button and type in the name `sw_layout_margin`. Set the value to `7.5`. Press the button again and type in the name `sw_layout_height_header`. Set the value to `36`.
5. Now you are ready to create the position and the size of the map canvas automatically by means of the variables. Make sure that your map item is selected, go to the *Item Properties* panel, scroll down to and open the *Position and Size* section. Click the Data defined override for *X* and from the *Variables* entry, choose `@sw_layout_margin`.
6. Click the Data defined override for *Y*, choose *Edit…* and type in the formula:

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

7. You can create the size of the map item by using the variables for *Width* and *Height*. Click the Data defined override for *Width* and choose *Edit …* again. Fill in the formula:

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

Click the Data defined override for *Height* and choose *Edit ….* Here fill in the formula:

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

8. We will also create a grid containing the coordinates of the main canvas map extent. Go to *Item Properties* again and choose the *Grids* section. Insert a grid by clicking the button. Click on *Modify grid …* and set the *Interval* for *X*, *Y* and *Offset* according to the map scale you chose in the QGIS main canvas. The *Grid type Cross* is very well suited for our purposes.

4.2.2 Follow Along: Creating the dynamic header

1. Insert a rectangle which will contain the header with the Add Shape button. In the *Items* panel enter the name `header`.
2. Again, go to the *Item Properties* and open the *Position and Size* section. Using Data defined override, choose the `sw_layout_margin` variable for *X* as well as for *Y*. *Width* shall be defined by the expression:

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

   and *Height* by the `sw_layout_height_header` variable.
3. We will insert a horizontal line and two vertical lines to divide the header into different sections using the Add Node Item. Create a horizontal line and two vertical lines and name them `Horizontal line`, `Vertical line 1` `Vertical line 2`.
   1. For the horizontal line:
      1. Set *X* to the variable `sw_layout_margin`
      2. Set the expression for *Y* to:

         ```
         @sw_layout_margin + 8
         ```

      3. Set the expression for *Width* to:

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

   2. For the first vertical line:
      1. Set the expression for *X* to:

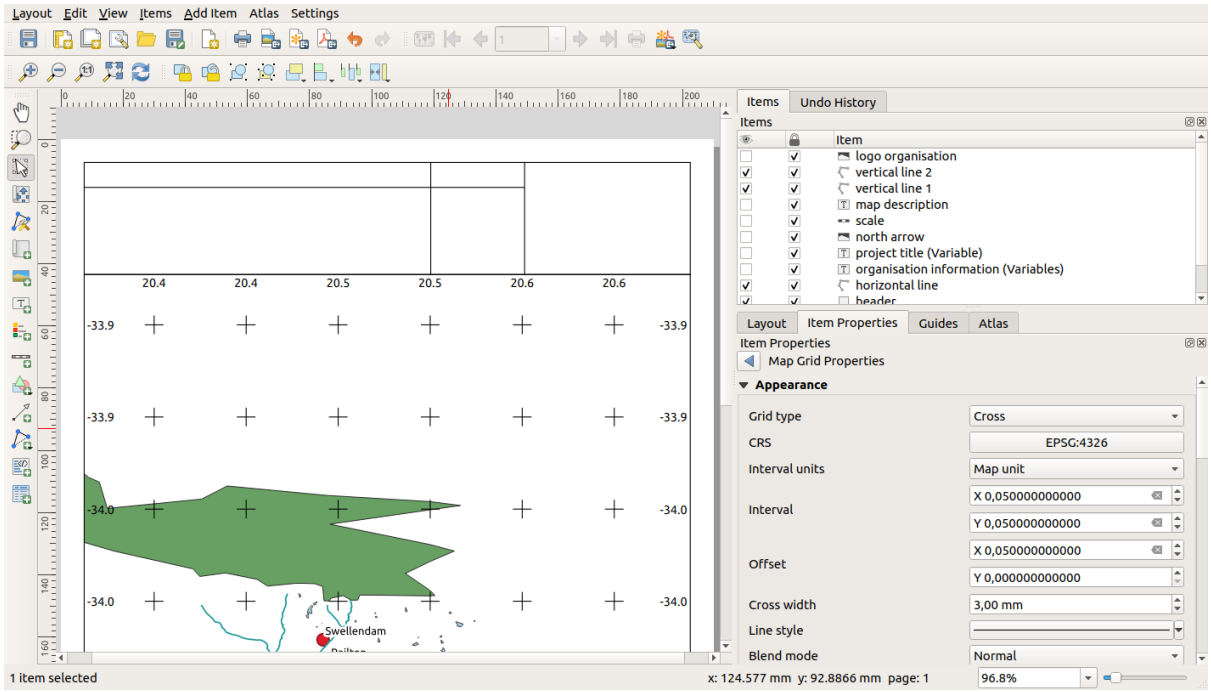
         ```
         @layout_pagewidth - @sw_layout_margin * 2 - 53.5
         ```

      2. Set *Y* to the variable `sw_layout_margin`
      3. The height must be the same as the header we created, so set *Height* to the variable `sw_layout_height_header`.
   3. The second vertical line is placed to the left of the first one.
      1. Set the expression for *X* to:

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

      2. Set *Y* to the variable `sw_layout_margin`
      3. The height shall be the same as the other vertical line, so set *Height* to the variable `sw_layout_height_header`.

   The figure below shows the structure of our dynamic layout. We will fill the areas created by the lines with some elements.



4.2.3 Follow Along: Creating labels for the dynamic header

1. The title of your QGIS project can be included automatically. The title is set in the *Project Properties*. Insert a label with the Add Label button and enter the name `project title (variable)`. In the *Main Properties* of the *Items Properties* Panel enter the expression:

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

   Set the position of the label.

   1. For *X*, use the expression:

```
@sw_layout_margin + 3
```

   2. For *Y*, use the expression:

```
@sw_layout_margin + 0.25
```

   3. For *Width*, use the expression:

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

   4. Enter `11.25` for *Height*

   Under *Appearance* set the Font size to 16 pt.

2. The second label will include a description of the map you created. Again, insert a label and name it `map description`. In the *Main Properties* enter the text `map description`. In the *Main Properties* we will also include:

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

   Here we used two `Date and Time` functions (`now` and `format_date`).

   Set the position of the label.

   1. For *X*, use the expression:

```
@sw_layout_margin + 3
```

2. For *Y*, use the expression:

```
@sw_layout_margin + 11.5
```

3. The third label will include information about your organisation. First we will create some variables in the *Variables* menu of the *Item Properties*. Go to the *Layout* menu, click the button each time and enter the names `o_department`, `o_name` , `o_adress` and `o_postcode`. In the second row enter the information about your organisation. We will use these variables in the *Main Properties* section.

   In *Main Properties* enter:

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

   Set the position of the label.

   1. For *X*, use the expression:

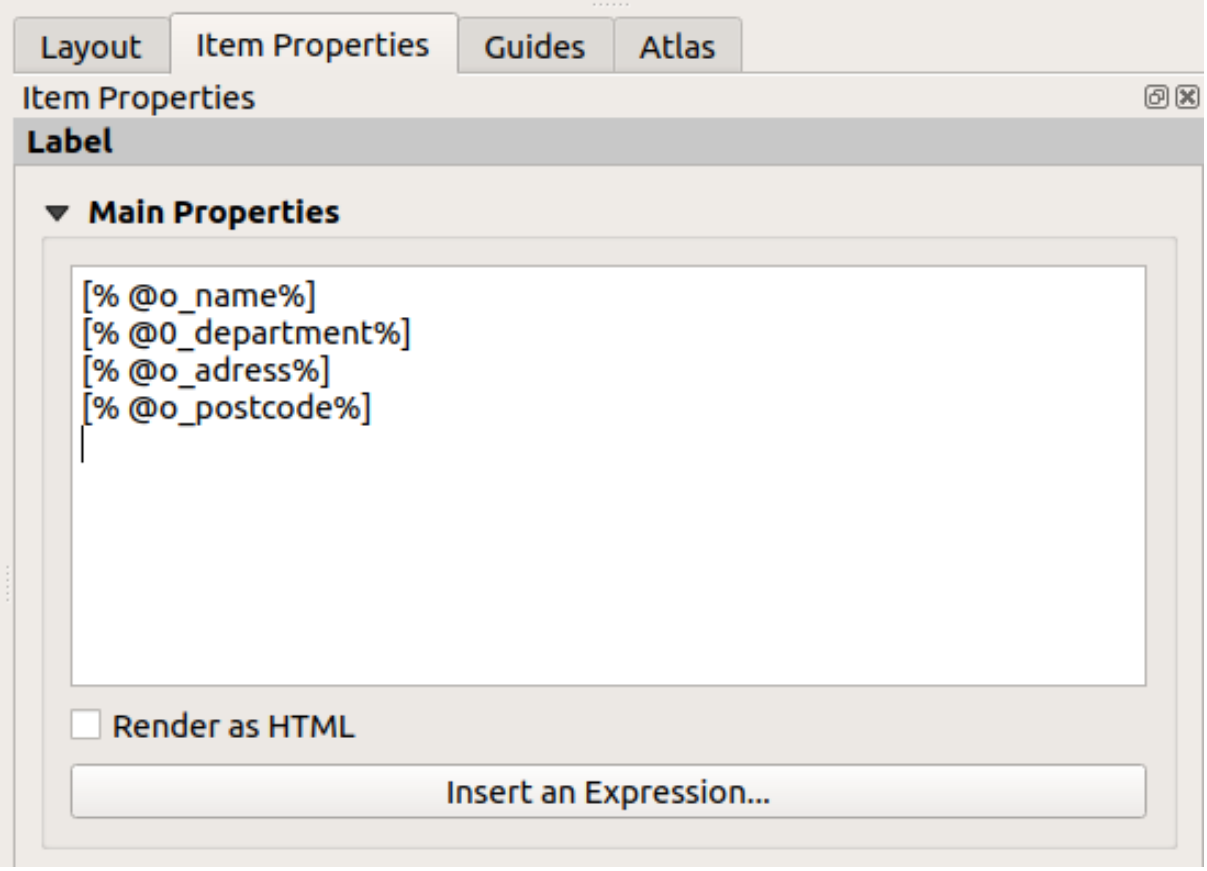
```
@layout_pagewidth - @sw_layout_margin - 49.5
```

   2. For *Y*, use the expression:

```
@sw_layout_margin + 15.5
```

   3. For *Width*, use `49.00`
   4. For *Height*, use the expression:

```
@sw_layout_height_header - 15.5
```



4.2.4 Follow Along: Adding pictures to the dynamic header

1. Use the Add Picture button to place a picture above your label `organisation information`. After entering the name `organisation logo` define the position and size of the logo:
   1. For *X*, use the expression:

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

   2. For *Y*, use the expression:

      ```
      @sw_layout_margin + 3.5
      ```

   3. For *Width*, use `39.292`
   4. For *Height*, use `9.583`

   To include a logo of your organisation you have to save your logo under your home directory and enter the path under *Main Properties* ► *Image Source*.

2. Our layout still needs a north arrow. This will also be inserted by using Add North Arrow. We will use the default north arrow. Define the position:
   1. For *X*, use the expression:

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

   2. For *Y*, use the expression:

```
@sw_layout_margin + 9
```

3. For *Width*, use `21.027`
4. For *Height*, use `21.157`

4.2.5 Follow Along: Creating the scalebar of the dynamic header

1. To insert a scalebar in the header click on Add Scale Bar and place it in the rectangle above the north arrow. In *Map* under the *Main Properties* choose your `main map(Map 1)`. This means that the scale changes automatically according to the extent you choose in the QGIS main canvas. Choose the *Style Numeric*. This means that we insert a simple scale without a scalebar. The scale still needs a position and size.
   1. For *X*, use the expression:

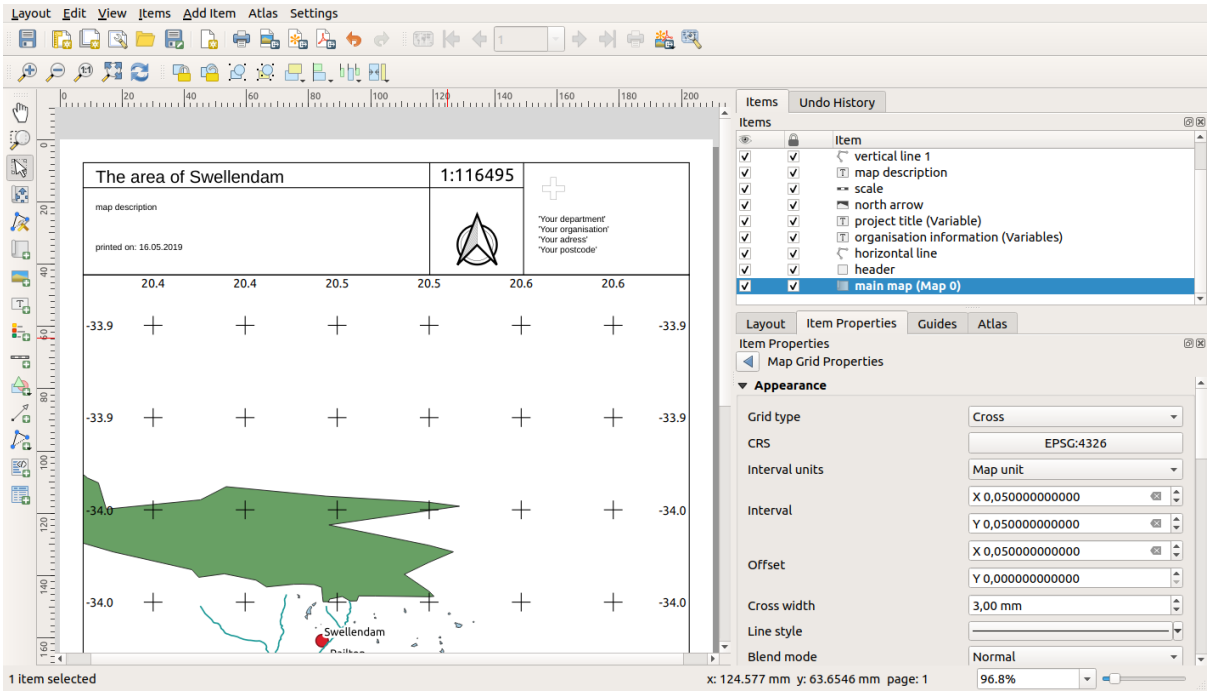
      ```
      @layout_pagewidth - @sw_layout_margin * 2 - 78
      ```

   2. For *Y*, use the expression:

      ```
      @sw_layout_margin + 1
      ```

   3. For *Width*, use `25`
   4. For *Height*, use `8`
   5. Place the `Reference point` in the center.

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



4.2.6 What's Next?

On the next page, you will be given an assignment to complete. This will allow you to practice the techniques you have learned so far.

4.3 Zuordnung 1

Open your existing map project and revise it thoroughly. If you have noticed small errors or things you'd have liked to fix earlier, do so now.

While customizing your map, keep asking yourself questions. Is this map easy to read and understand for someone who's unfamiliar with the data? If I saw this map on the Internet, or on a poster, or in a magazine, would it capture my attention? Would I want to read this map if it wasn't mine?

If you're doing this course at a Basic or Intermediate level, read up on techniques from the more advanced sections. If you see something you'd like to do in your map, why not try to implement it?

If this course is being presented to you, the course presenter may require you to submit a final version of your map, exported to PDF, for evaluation. If you're doing this course by yourself, it's recommended that you evaluate your own map using the same criteria. Your map will be evaluated on the overall appearance and symbology of the map itself, as well as the appearance and layout of the map page and elements. Remember that the emphasis for evaluating the appearance of maps will always be *ease of use*. The nicer the map is to look at and the easier it is to understand at a glance, the better.

Happy customizing!

4.3.1 In Conclusion

The first four modules have taught you how to create and style a vector map. In the next four modules, you'll learn how to use QGIS for a complete GIS analysis. This will include creating and editing vector data; analyzing vector data; using and analyzing raster data; and using GIS to solve a problem from start to finish, using both raster and vector data sources.

KAPITEL 5

Module: Vektordaten erzeugen

Die Erstellung von Karten unter Verwendung vorhandener Daten ist nur der Anfang. In diesem Modul lernen Sie, wie Sie bestehende Vektordaten modifizieren und vollständig neue Datensätze erstellen können.

5.1 Lesson: Creating a New Vector Dataset

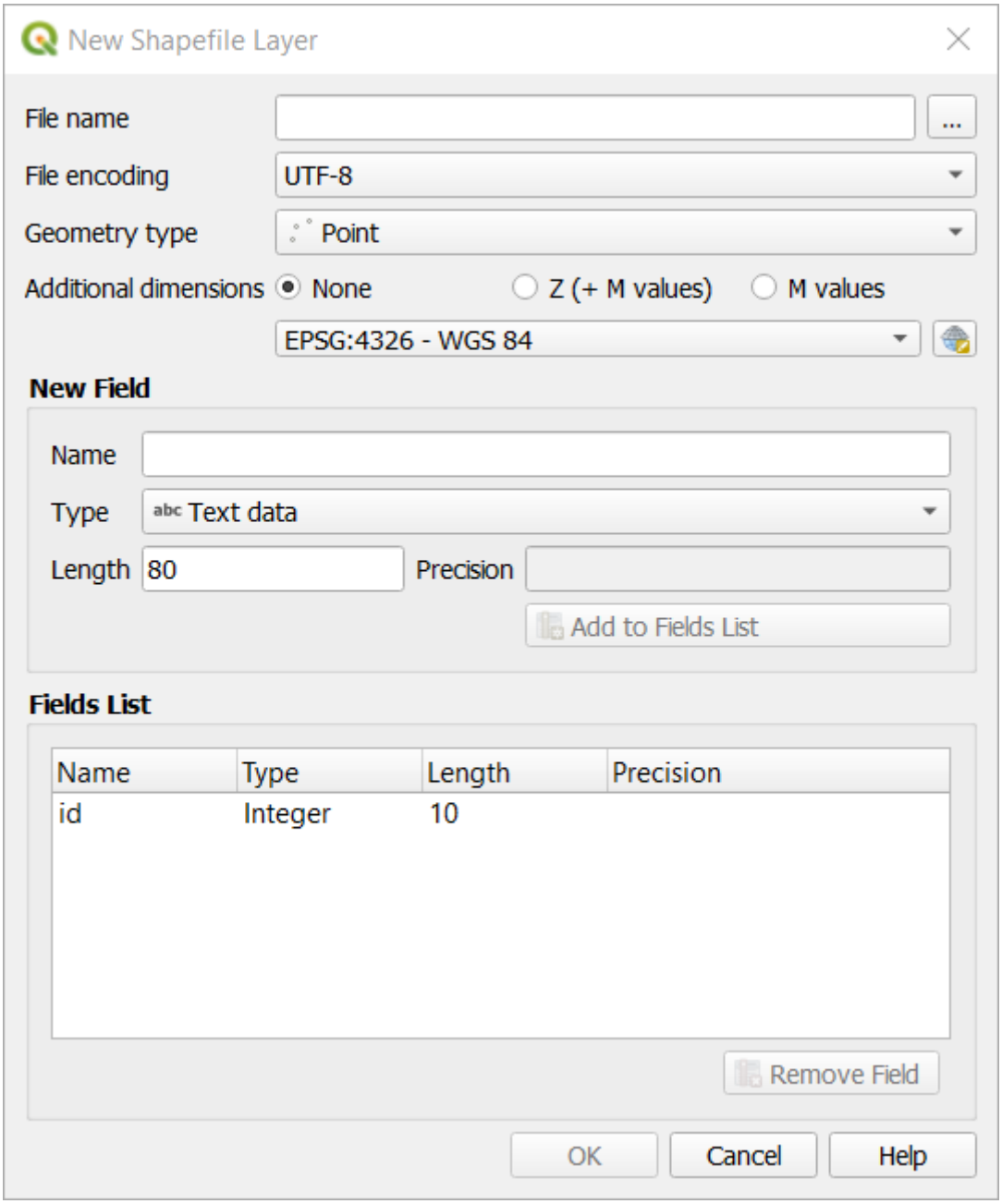
The data that you use has to come from somewhere. For most common applications, the data exists already; but the more particular and specialized the project, the less likely it is that the data will already be available. In such cases, you'll need to create your own new data.

The goal for this lesson: To create a new vector dataset.

5.1.1 Follow Along: The Layer Creation Dialog

Before you can add new vector data, you need a vector dataset to add it to. In our case, you'll begin by creating new data entirely, rather than editing an existing dataset. Therefore, you'll need to define your own new dataset first.

1. Open QGIS and create a new blank project.
2. Navigate to and click on the menu entry *Layer* ► *Create Layer* ► *New Shapefile Layer*. You'll be presented with the *New Shapefile Layer* dialog, which will allow you to define a new layer.

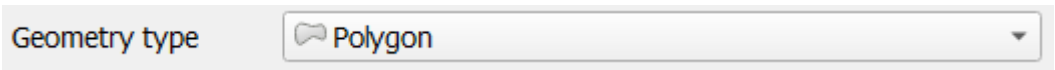


3. Click … for the *File name* field. A save dialog will appear.
4. Navigate to the `exercise_data` directory.
5. Save your new layer as `school_property.shp`.

   It's important to decide which kind of dataset you want at this stage. Each different vector layer type is „built differently" in the background, so once you've created the layer, you can't change its type.

   For the next exercise, we're going to create new features which describe areas. For such features, you'll need to create a polygon dataset.

6. For *Geometry Type*, select *Polygon* from the drop down menu:

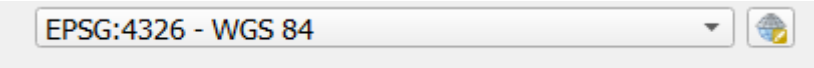


   This has no impact on the rest of the dialog, but it will cause the correct type of geometry to be used when the

vector dataset is created.

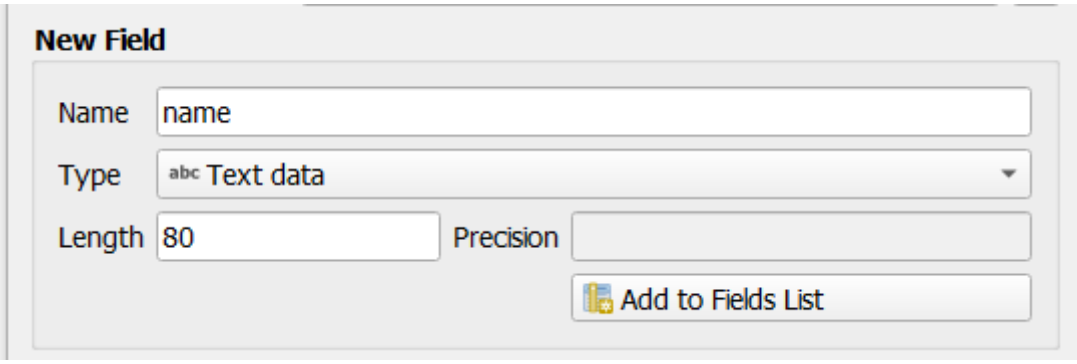
The next field allows you to specify the Coordinate Reference System, or CRS. CRS is a method of associating numerical coordinates with a position on the surface of the Earth. See the User Manual on Working with Projections to learn more.

For this example we will use the default CRS associated with this project, which is WGS84.

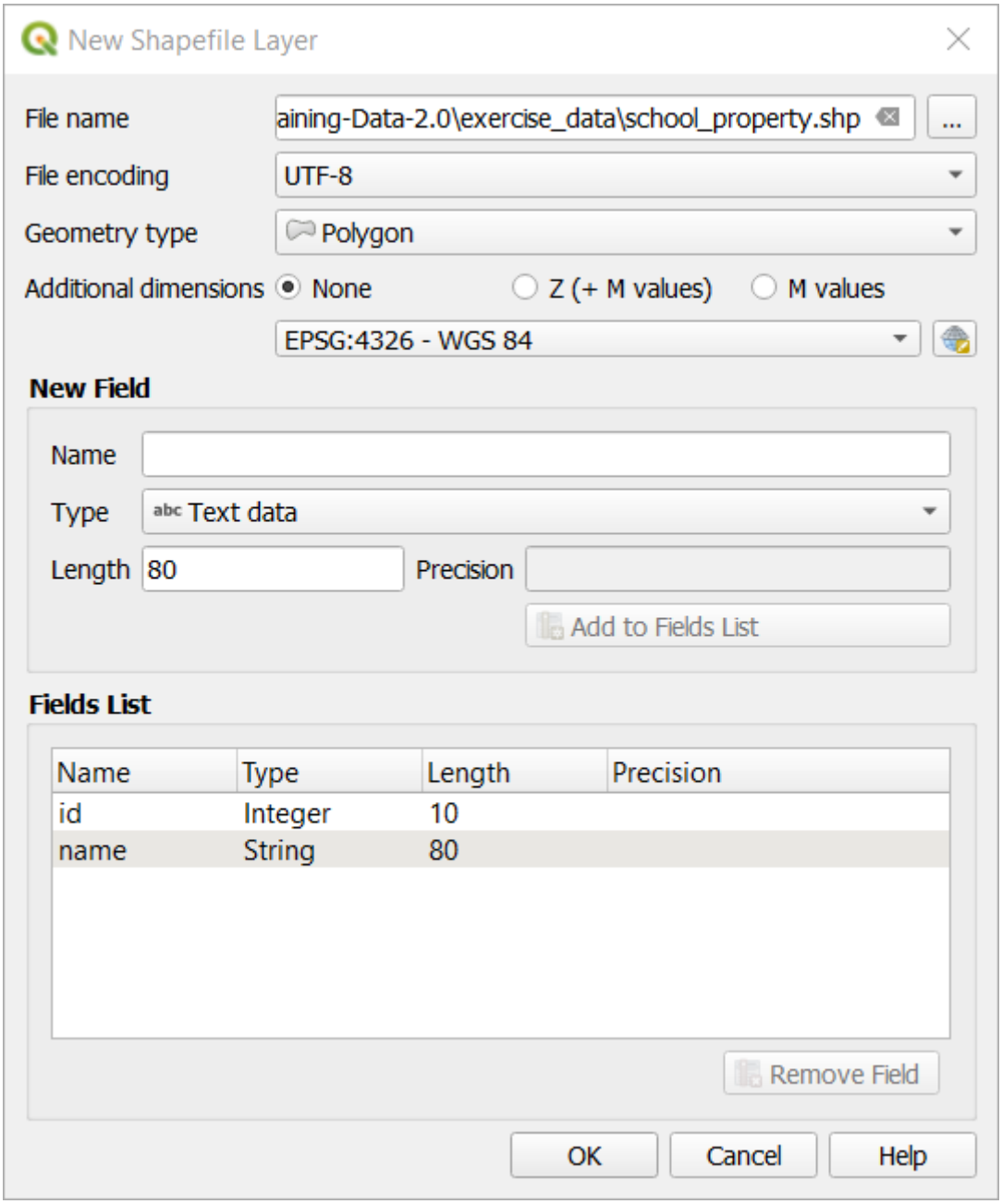


Next there is a collection of fields grouped under *New Field*. By default, a new layer has only one attribute, the `id` field (which you should see in the *Fields list*) below. However, in order for the data you create to be useful, you actually need to say something about the features you'll be creating in this new layer. For our current purposes, it will be enough to add one field called `name` that will hold `Text data` and will be limited to text length of `80` characters.

7. Replicate the setup below, then click the *Add to Fields List* button:



8. Check that your dialog now looks like this:



9. Klicken Sie auf *OK*

The new layer should appear in your *Layers* panel.

5.1.2 Follow Along: Datenquellen

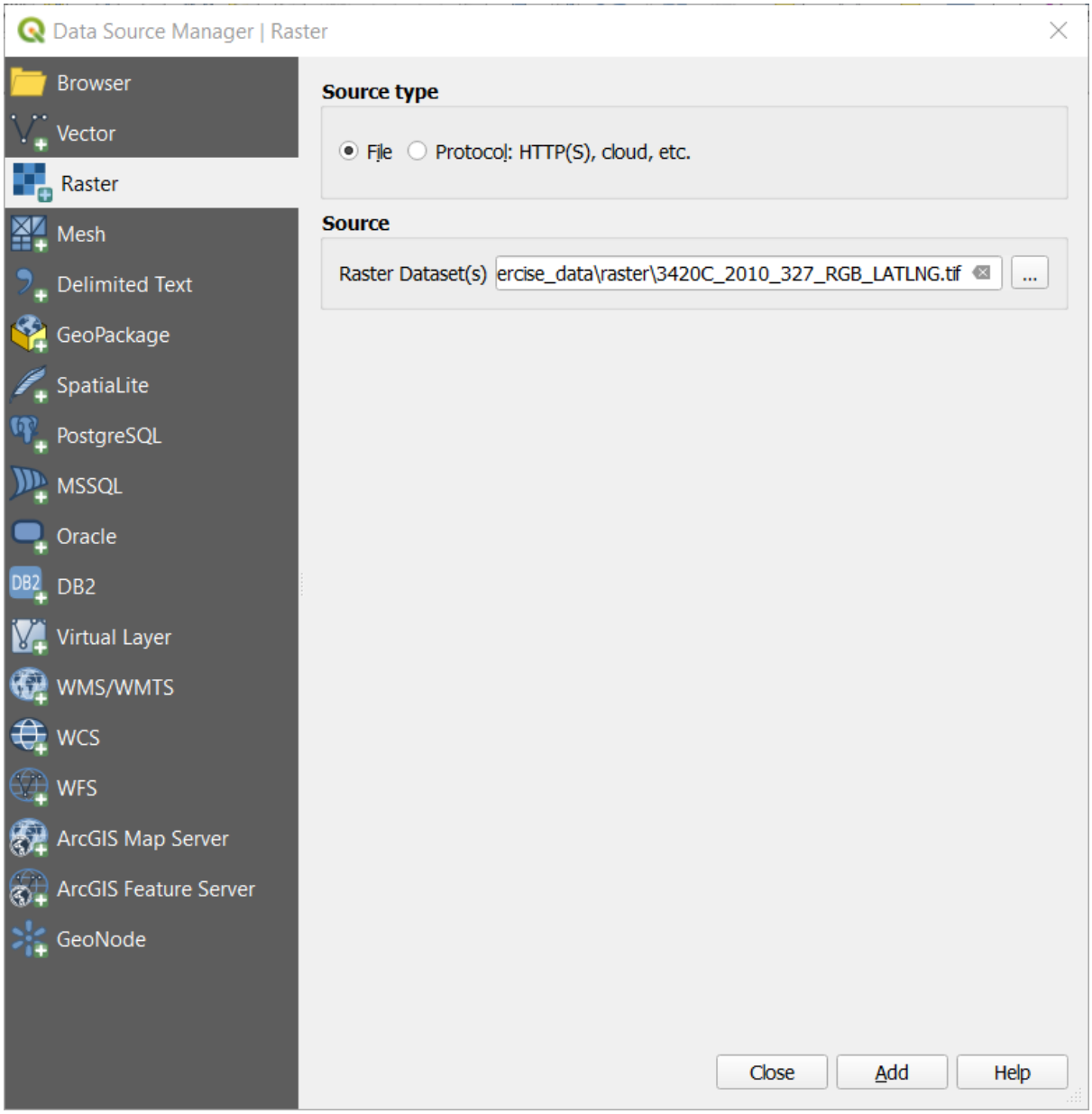
When you create new data, it obviously has to be about objects that really exist on the ground. Therefore, you'll need to get your information from somewhere.

There are many different ways to obtain data about objects. For example, you could use a GPS to capture points in the real world, then import the data into QGIS afterwards. Or you could survey points using a theodolite, and enter the coordinates manually to create new features. Or you could use the digitizing process to trace objects from remote sensing data, such as satellite imagery or aerial photography.

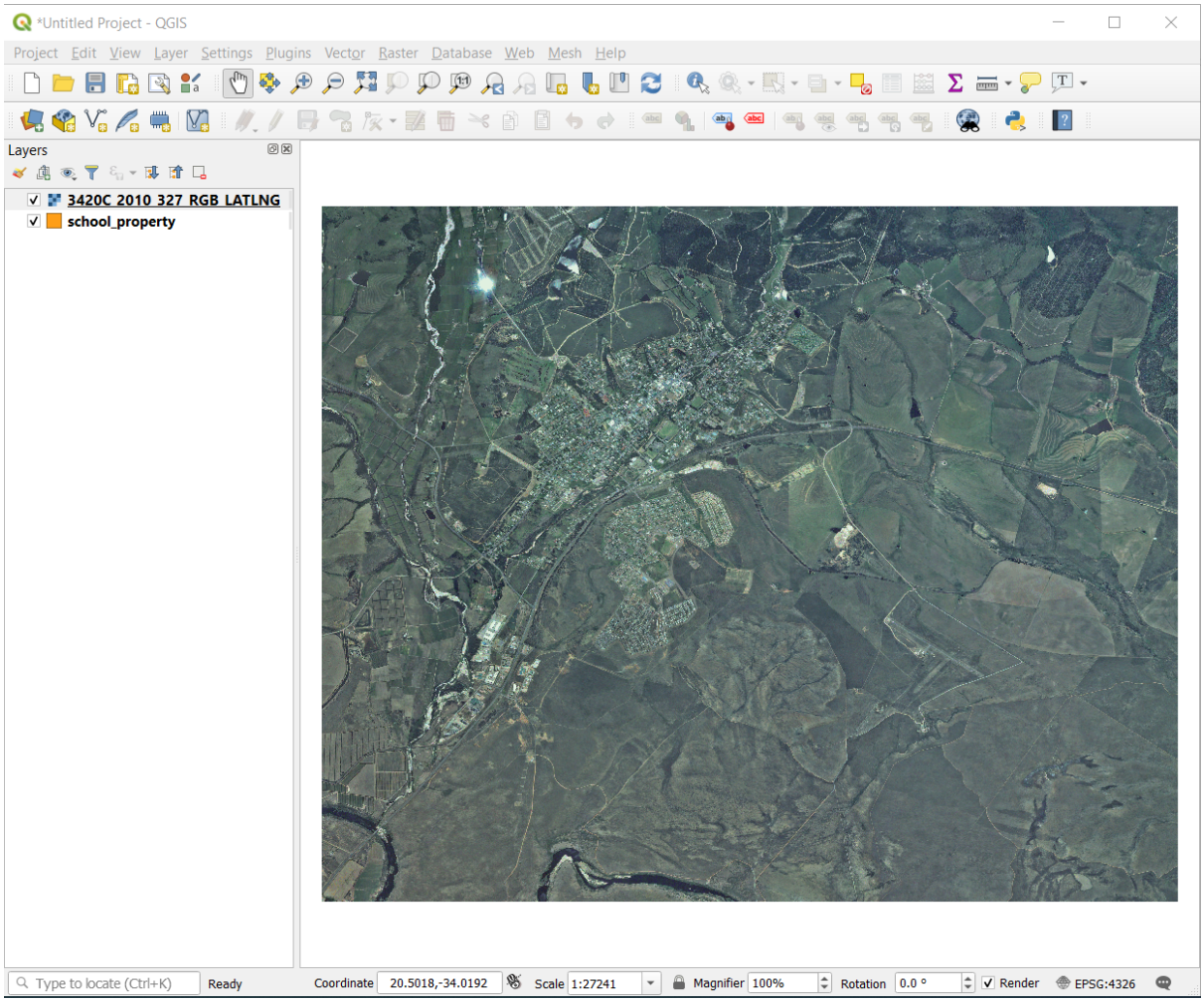
For our example, you'll be using the digitizing approach. Sample raster datasets are provided, so you'll need to import

them as necessary.

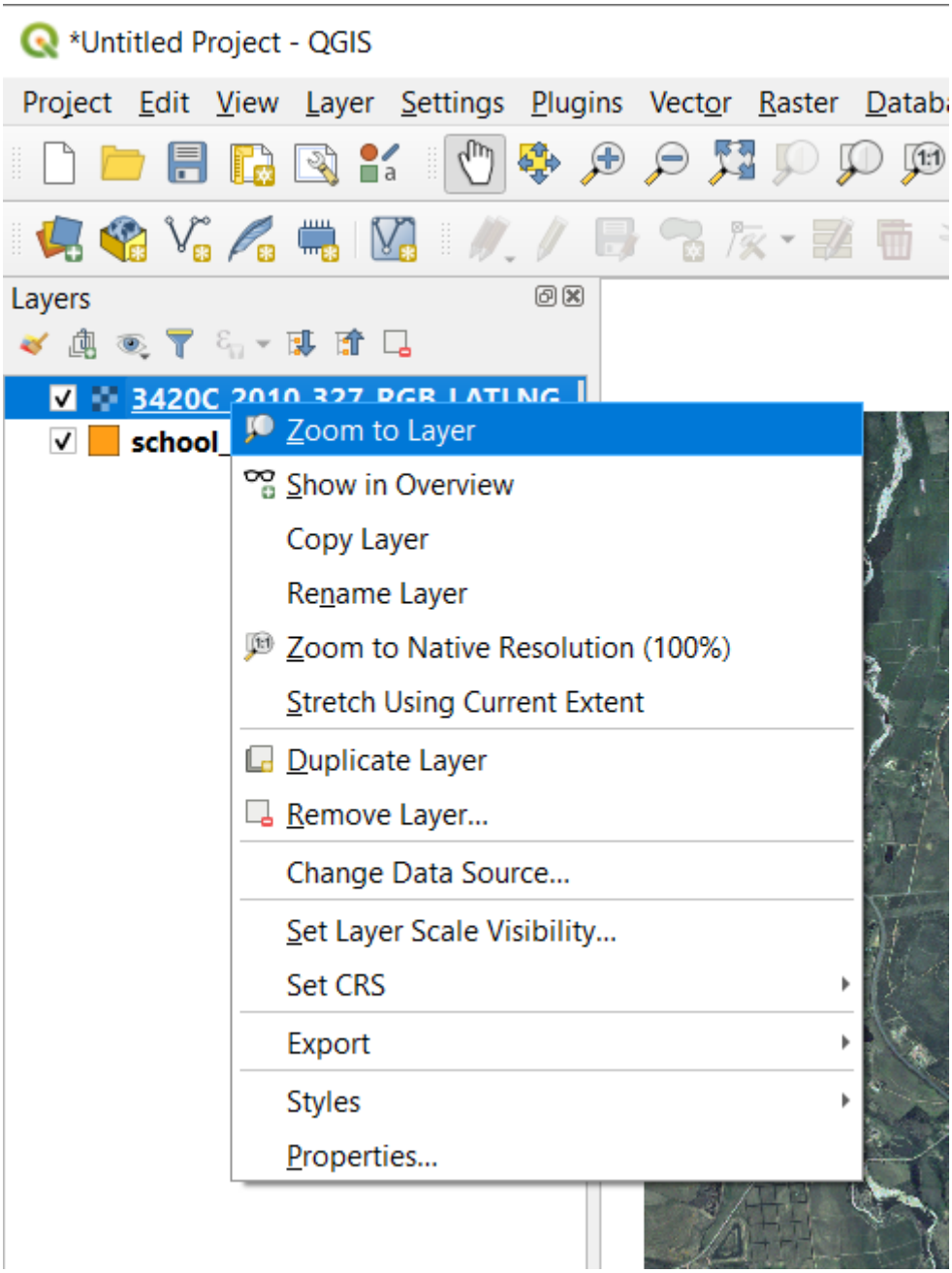
1. Click on Data Source Manager button.
2. Select *Raster* on the left side.
3. In the *Source* panel, click on the … button:
4. Navigate to `exercise_data/raster/`.
5. Select the file `3420C_2010_327_RGB_LATLNG.tif`.
6. Click *Open* to close the dialogue window.



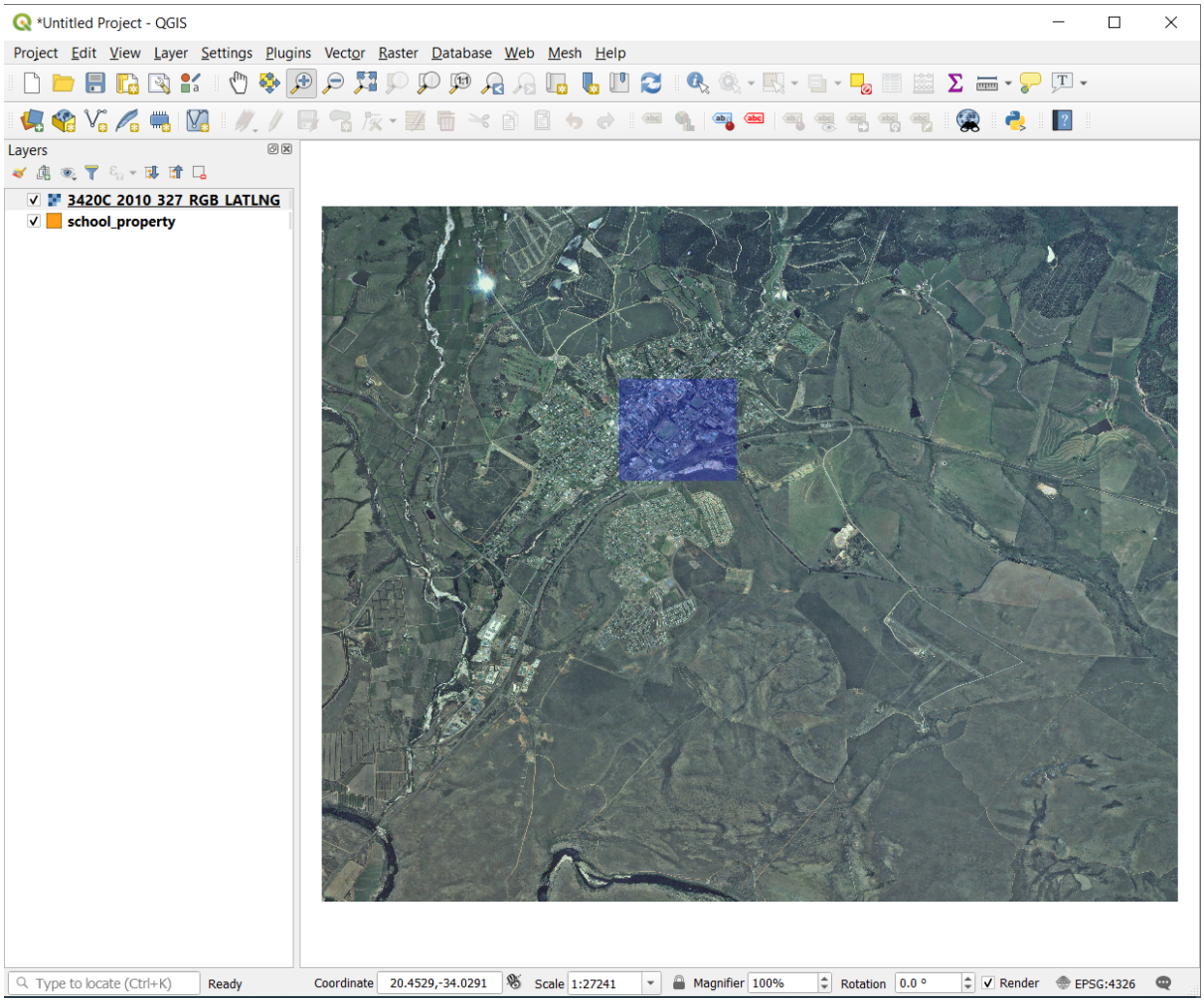
7. Click *Add* and *Close*. An image will load into your map.



8. If you don't see an aerial image appear, select the new layer, right click, and choose *Zoom to Layer* in the context menu.



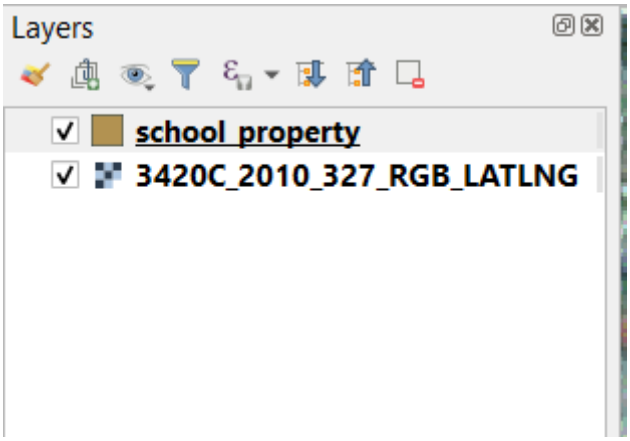
9. Click on the Zoom In button, and zoom to the area highlighted in blue below:



Now you are ready to digitize these three fields:

Before starting to digitize, let's move the `school_property` layer above the aerial image.

1. Select `school_property` layer in the *Layers* pane and drag it to the top.



In order to begin digitizing, you'll need to enter **edit mode**. GIS software commonly requires this to prevent you from

accidentally editing or deleting important data. Edit mode is switched on or off individually for each layer.

To enter edit mode for the `school_property` layer:

1. Click on the `school_property` layer in the *Layers* panel to select it.
2. Click on the Toggle Editing button.

   If you can't find this button, check that the *Digitizing* toolbar is enabled. There should be a check mark next to the *View* ► *Toolbars* ► *Digitizing* menu entry.

   As soon as you are in edit mode, you'll see that some digitizing tools have become active:

   - Capture Polygon
   - Vertex Tool

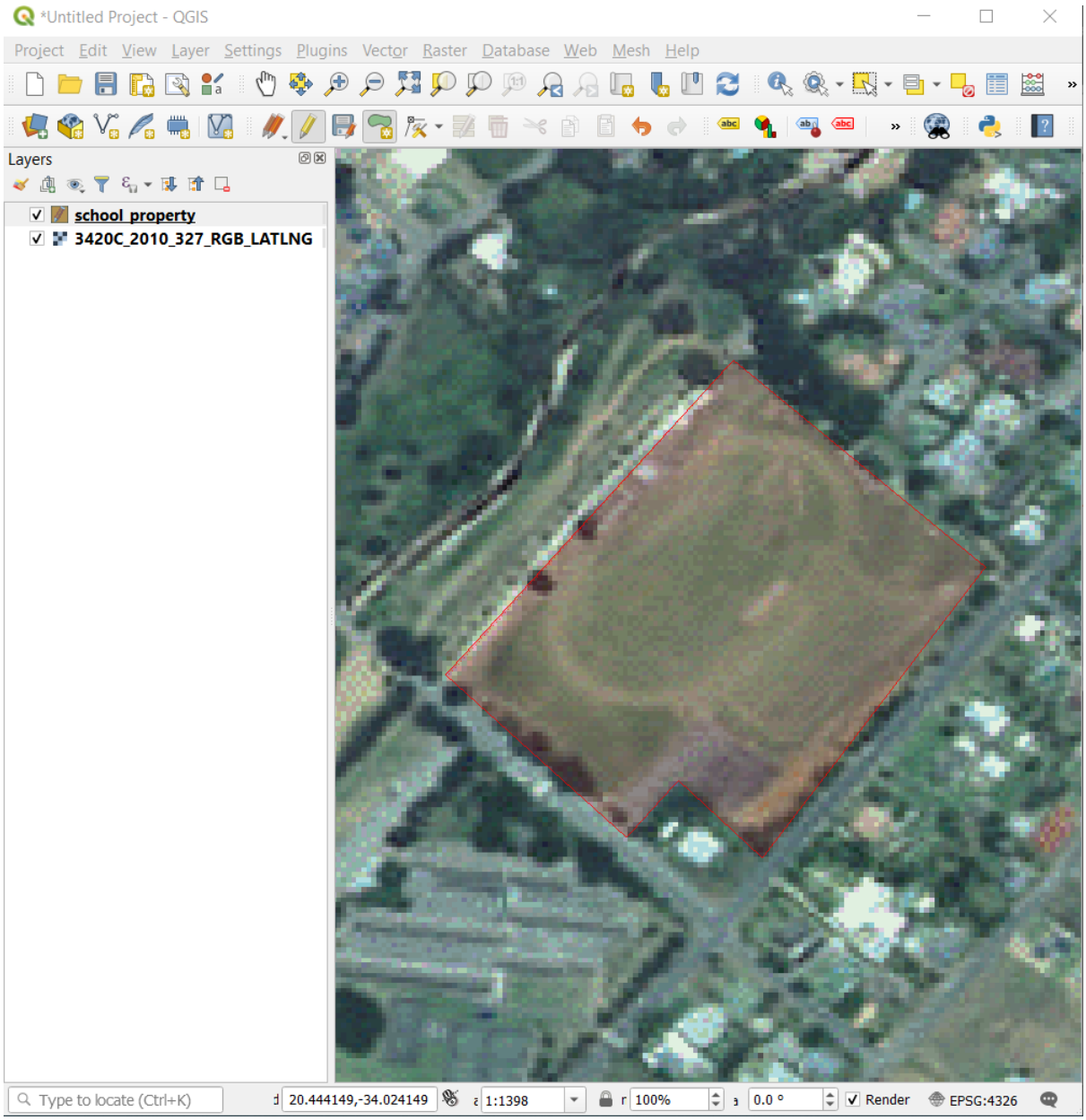
   Other relevant buttons are still inactive, but will become active when we start interacting with our new data.

   Notice that the layer `school_property` in the *Layers* panel now has the pencil icon, indicating that it is in edit mode.
3. Click on the Capture Polygon button to begin digitizing our school fields.

   You'll notice that your mouse cursor has become a crosshair. This allows you to more accurately place the points you'll be digitizing. Remember that even when you're using the digitizing tool, you can zoom in and out on your map by rolling the mouse wheel, and you can pan around by holding down the mouse wheel and dragging around in the map.

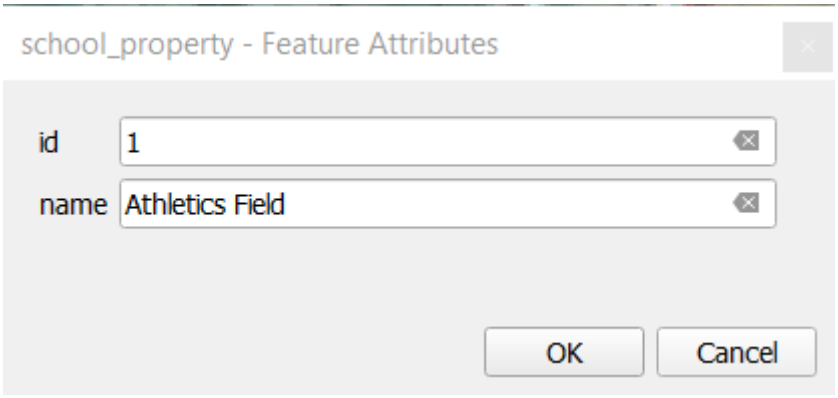
   The first feature you'll be digitizing is the athletics field:

4. Start digitizing by clicking on a point somewhere along the edge of the field.
5. Place more points by clicking further along the edge, until the shape you're drawing completely covers the field.

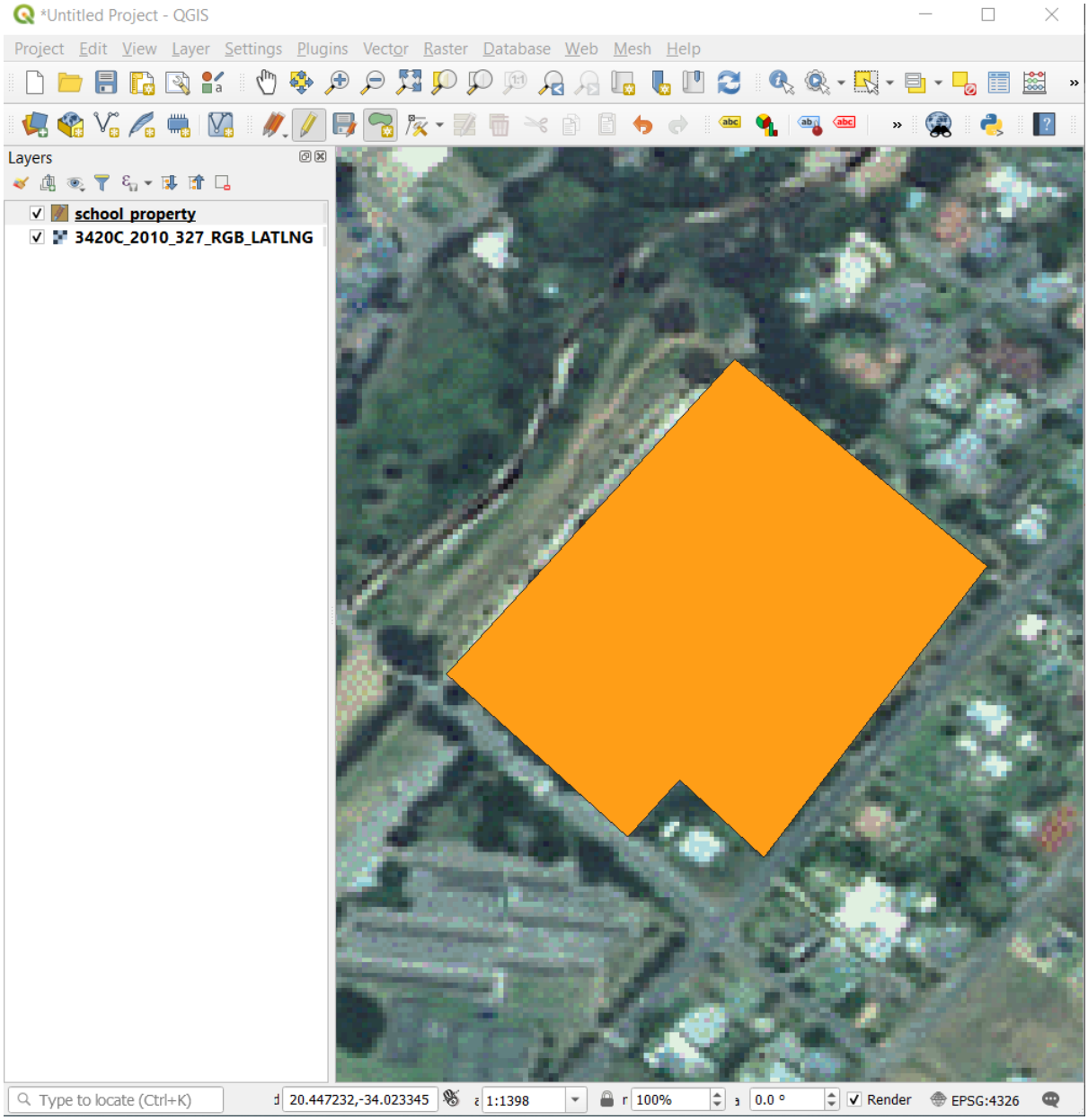


6. After placing your last point, right click to finish drawing the polygon. This will finalize the feature and show you the *Attributes* dialog.

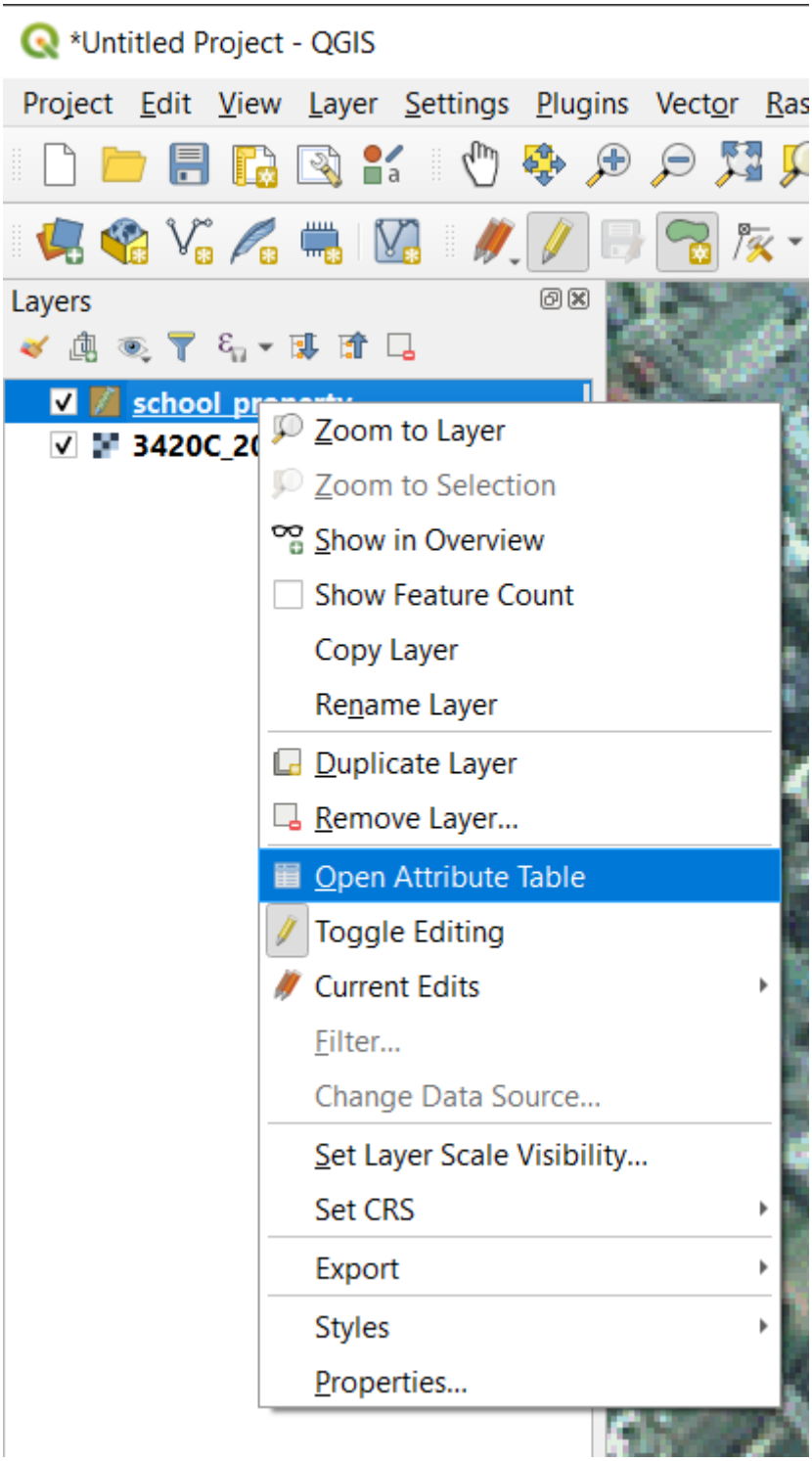
7. Fill in the values as below:



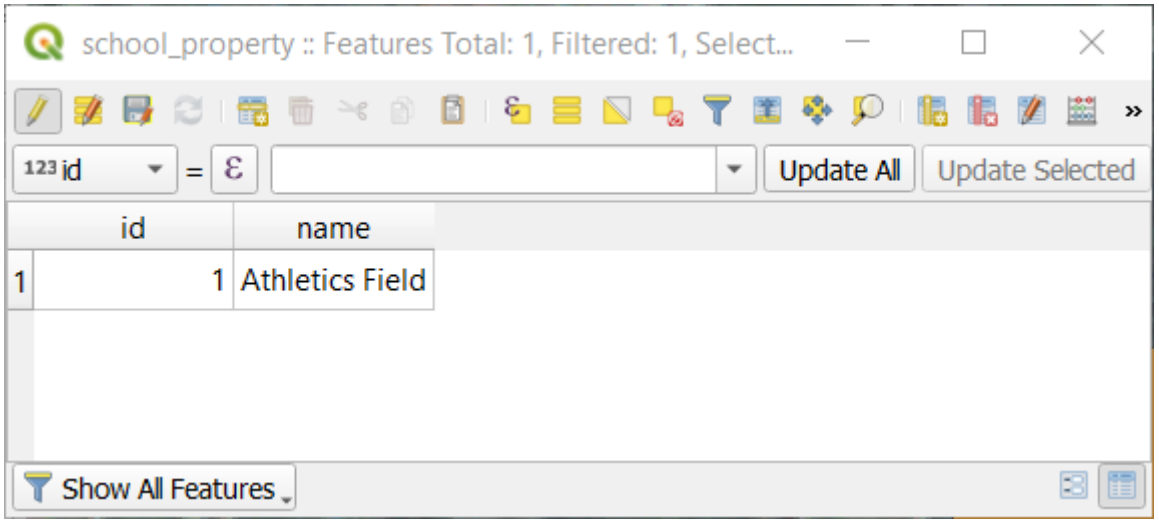
8. Click *OK*, and you have created a new feature!



9. In the *Layers* panel select the `school_property` layer.
10. Right click and choose *Open Attribute Table* in the context menu.



In the table you will see the feature you just added. While in edit mode you can update the attributes data by double click on the cell you want to update.



11. Close the attribute table.
12. To save the new feature we just created, click on Save Edits button.

Remember, if you've made a mistake while digitizing a feature, you can always edit it after you're done creating it. If you've made a mistake, continue digitizing until you're done creating the feature as above. Then:

1. Click on Vertex Tool button.
2. Hover the mouse over a vertex you want to move and left click on the vertex.
3. Move the mouse to the correct location of the vertex, and left click. This will move the vertex to the new location.

The same procedure can be used to move a line segment, but you will need to hover over the midpoint of the line segment.

If you want to undo a change, you can press the Undo button or `Ctrl+Z`.

4. Remember to save your changes by clicking the Save Edits button.
5. When done editing, click the Toggle Editing button to get out of edit mode.

5.1.3 Try Yourself Digitizing Polygons

Digitize the school itself and the upper field. Use this image to assist you:

Remember that each new feature needs to have a unique `id` value!

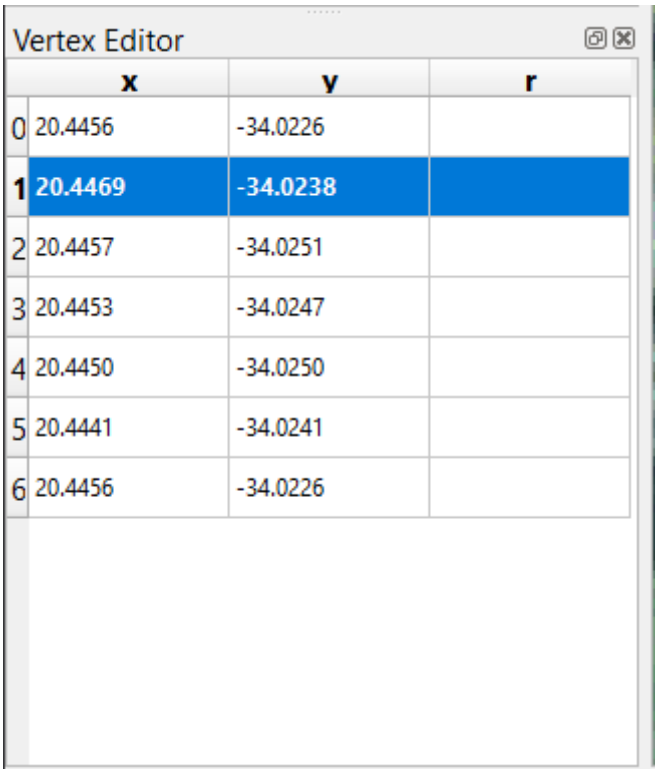
Bemerkung: When you're done adding features to a layer, remember to save your edits and then exit edit mode.

Bemerkung: You can style the fill, outline and label placement and formatting of the `school_property` using techniques learnt in earlier lessons.

5.1.4 Follow Along: Using Vertex Editor Table

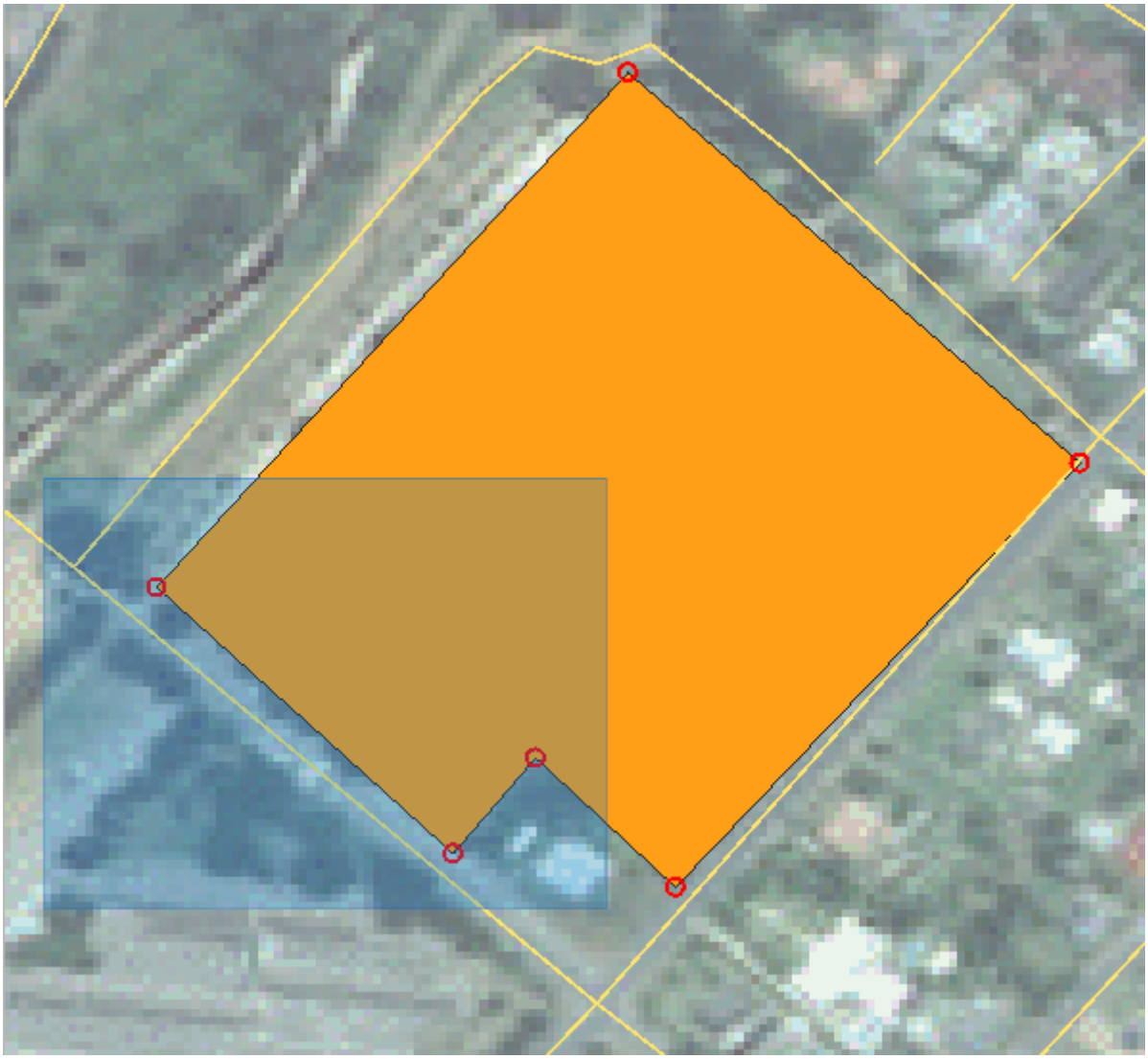
Another way to edit a feature is to manually enter the actual coordinate values for each vertex using the *Vertex Editor* table.

1. Make sure you are in edit mode on layer `school_property`.
2. If not already activated, click on Vertex Tool button.
3. Move the mouse over one of the polygon features you created in the `school_property` layer and right click on it. This will select the feature and a *Vertex Editor* pane will appear.

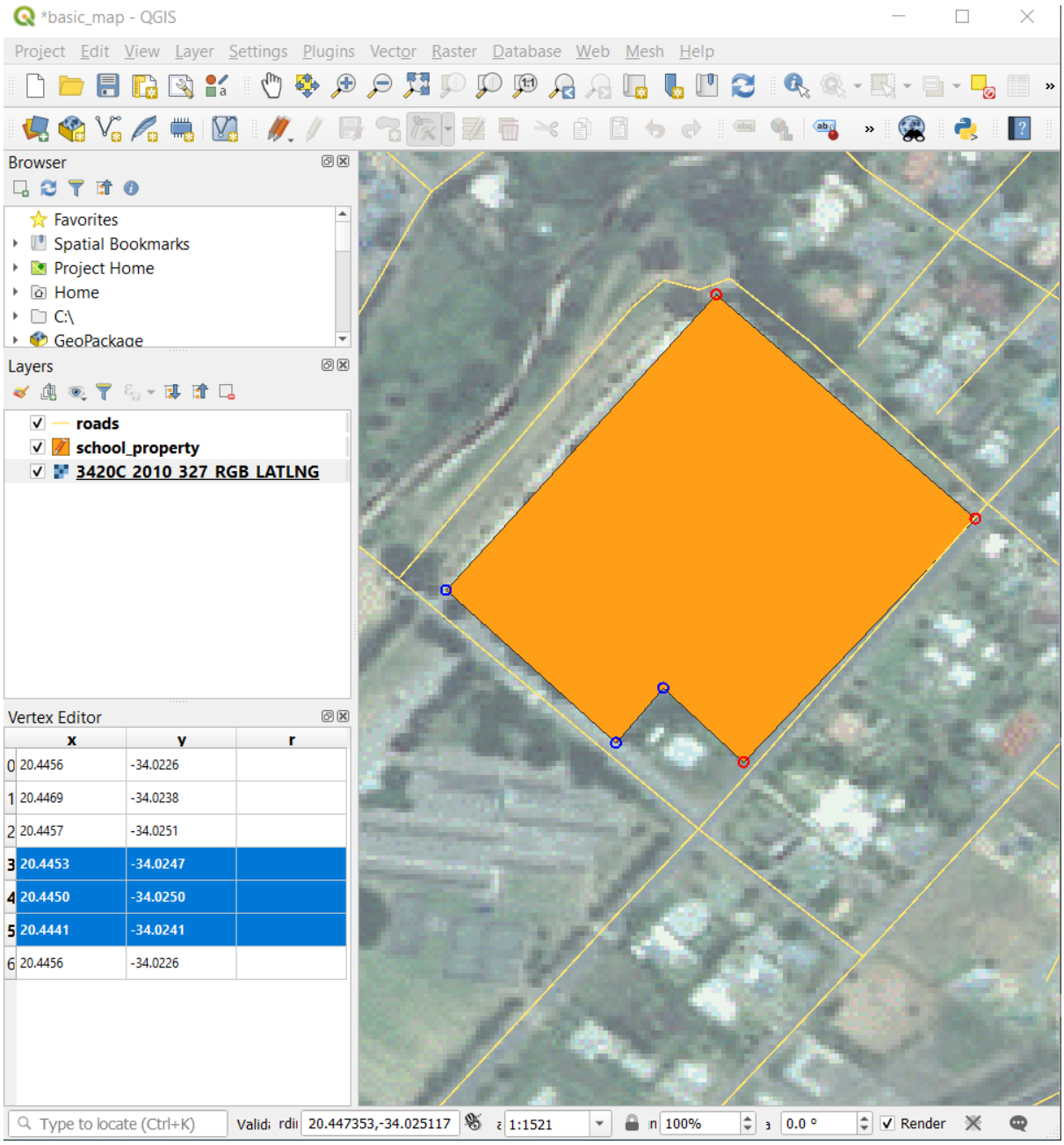


Bemerkung: This table contains the coordinates for the vertices of the feature. Notice there are seven vertices for this feature, but only six are visually identified in the map area. Upon closer inspection, one will notice that row 0 and 6 have identical coordinates. These are the start and end vertices of the feature geometry, and are required in order to create a closed polygon feature.

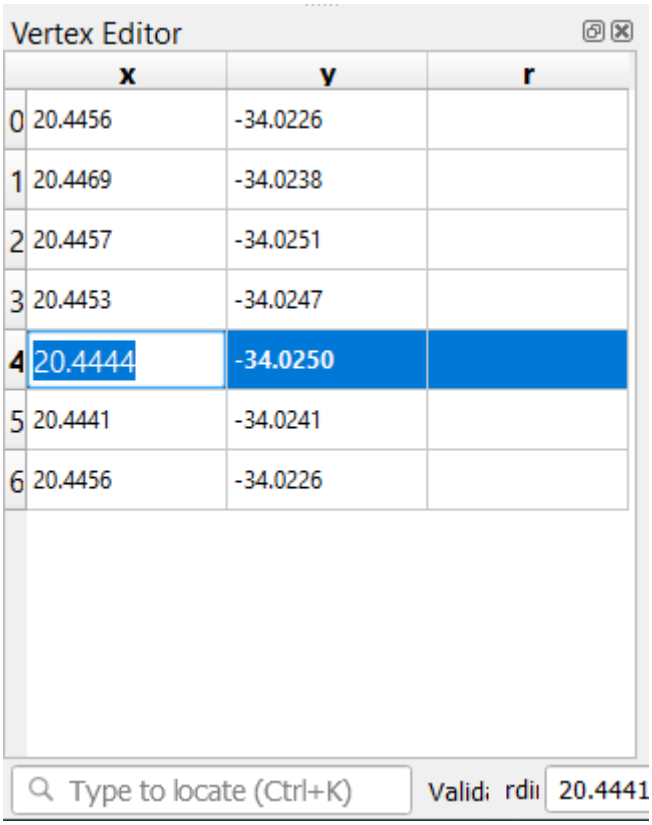
4. Click and drag a box over a vertex, or multiple vertices, of the selected feature.



The selected vertices will change to a color blue and the *Vertex Editor* table will have the corresponding rows highlighted, which contain the coordinates of the vertices.



5. To update a coordinate, double left click on the cell in the table that you want to edit and enter the updated value. In this example, the x coordinate of row `4` is updated from `20.4450` to `20.4444`.



6. After entering the updated value, hit the enter key to apply the change. You will see the vertex move to the new location in the map window.

7. When done editing, click the Toggle Editing button to get out of edit mode, and save your edits.

5.1.5 Try Yourself Digitizing Lines

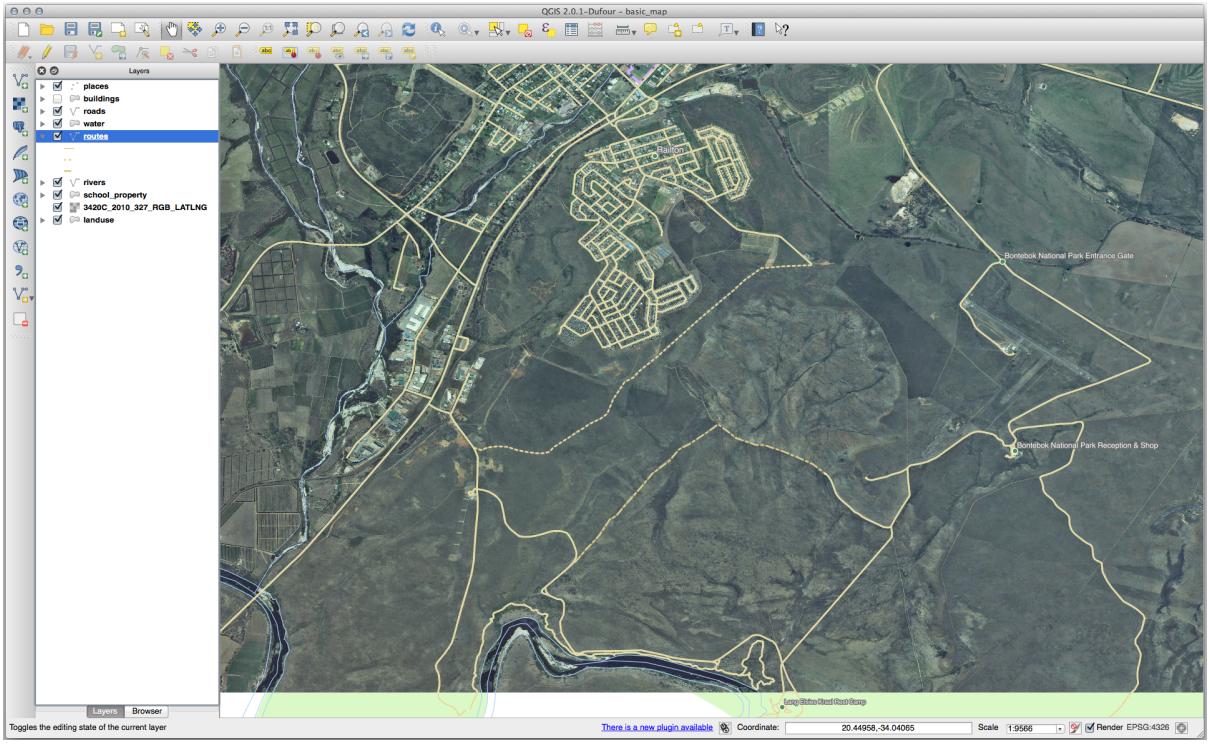
We are going to digitize two routes which are not already marked on the roads layer; one is a path, the other is a track. Our path runs along the southern edge of the suburb of Railton, starting and ending at marked roads:

Our track is a little further to the south:

1. If the *roads* layer is not yet in your map, then add the `roads` layer from the GeoPackage file `training-data.gpkg` included in the `exercise_data` folder of the training data you downloaded. You can read *Follow Along: Loading vector data from a GeoPackage Database* for a how-to.
2. Create a new ESRI Shapefile line dataset called `routes.shp` in the `exercise_data` directory, with attributes `id` and `type` (use the approach above to guide you.)
3. Activate edit mode on the *routes* layer.
4. Since you are working with a line feature, click on the Add Line button to initiate line digitizing mode.
5. One at a time, digitize the path and the track on the `routes` layer. Try to follow the routes as accurately as possible, adding additional points along corners or turns.
6. Set the `type` attribute value to `path` or `track`.
7. Use the *Layer Properties* dialog to add styling to your routes. Feel free to use different styles for paths and tracks.
8. Save your edits and toggle off editing mode by pressing the Toggle Editing button.

Answer

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



5.1.6 In Conclusion

Now you know how to create features! This course doesn't cover adding point features, because that's not really necessary once you've worked with more complicated features (lines and polygons). It works exactly the same, except that you only click once where you want the point to be, give it attributes as usual, and then the feature is created.

Knowing how to digitize is important because it's a very common activity in GIS programs.

5.1.7 What's Next?

Features in a GIS layer aren't just pictures, but objects in space. For example, adjacent polygons know where they are in relation to one another. This is called **topology**. In the next lesson you'll see an example of why this can be useful.

5.2 Lesson: Funktionstopologie

Topology is a useful aspect of vector data layers, because it minimizes errors such as overlap or gaps.

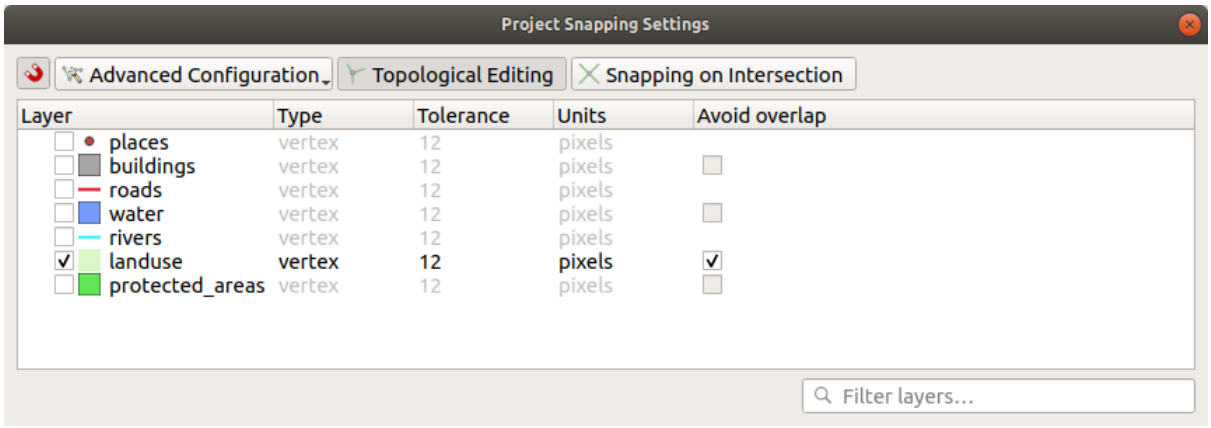
For example: if two features share a border, and you edit the border using topology, then you won't need to edit first one feature, then another, and carefully line up the borders so that they match. Instead, you can edit their shared border and both features will change at the same time.

The goal for this lesson: To understand topology using examples.

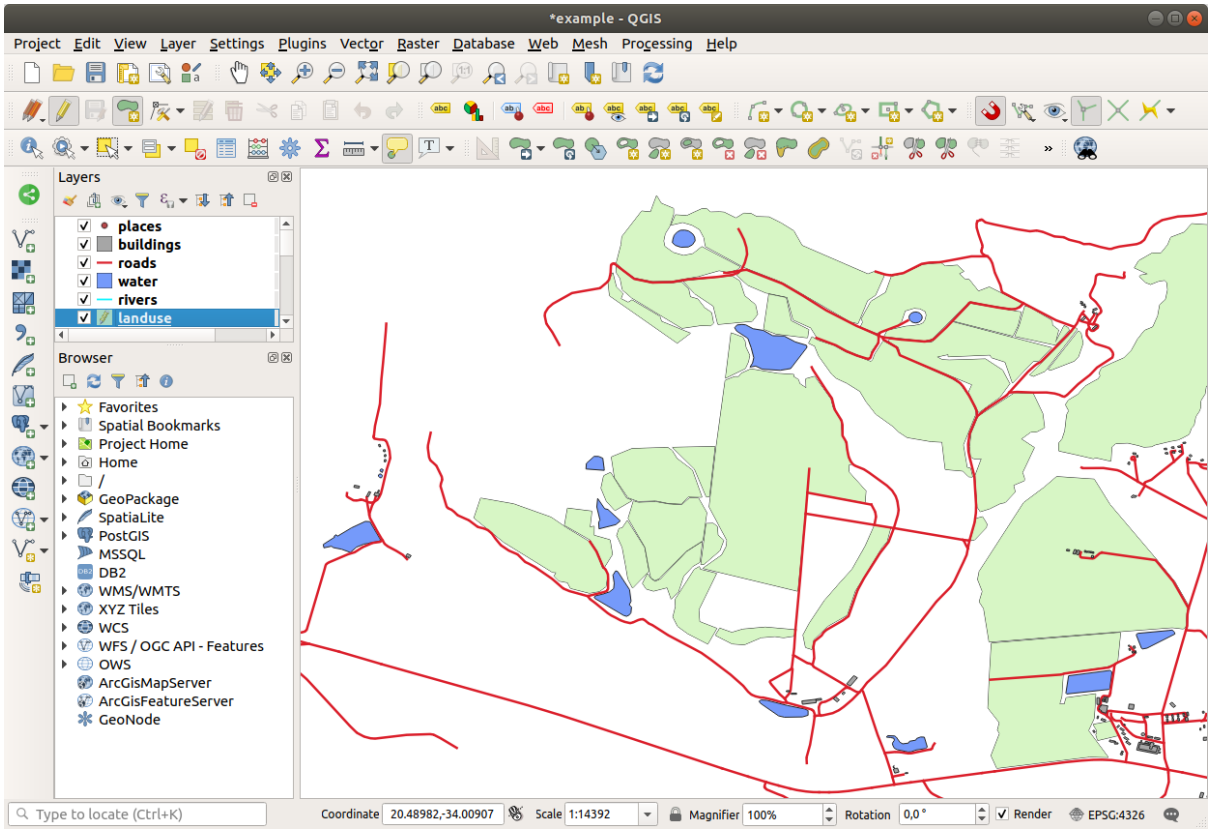
5.2.1 Follow Along: Snapping

Snapping makes topological editing easier. This will allow your mouse cursor to snap to other objects while you digitize. To set snapping options:

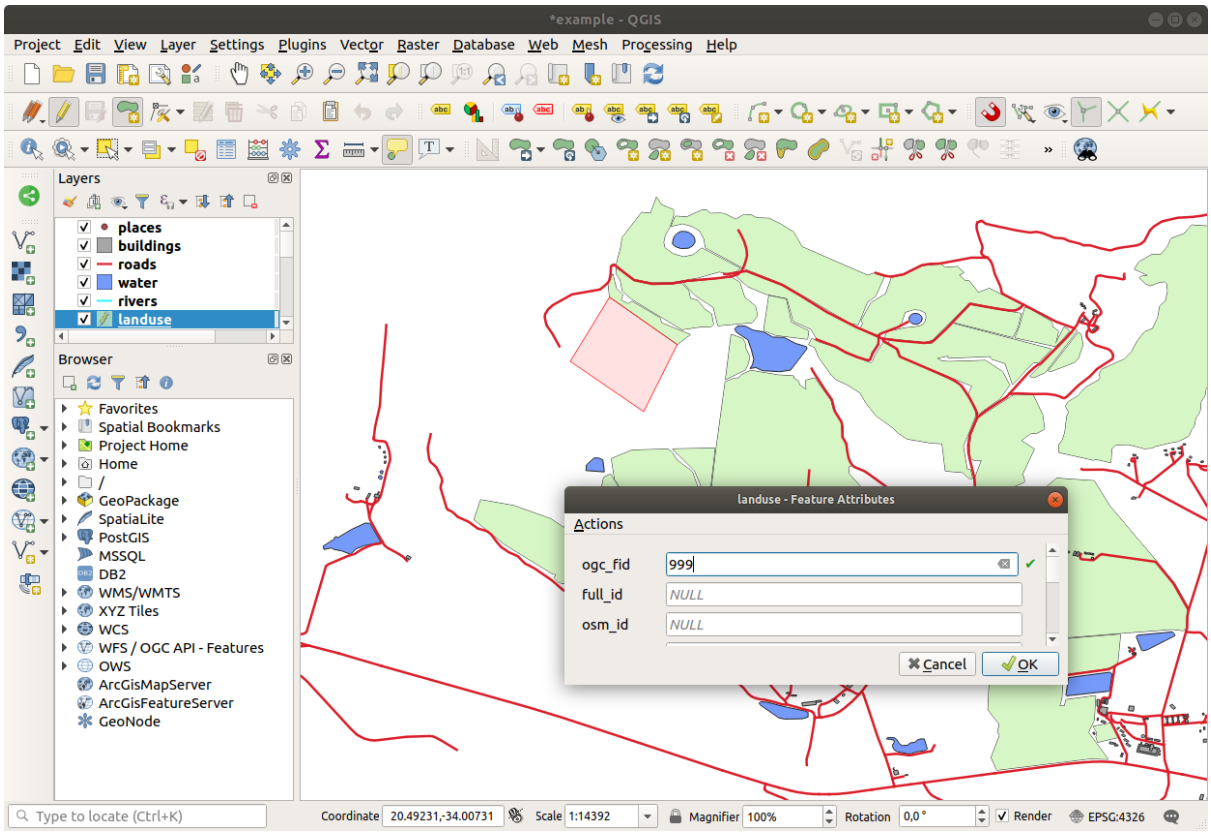
1. Navigate to the menu entry *Project* ► *Snapping Options…*.
2. Set up your *Snapping options* dialog to activate the `landuse` layer with *Type vertex* and tolerance `12` pixels:



3. Make sure that the box in the *Avoid overlap* column is checked.
4. Leave the dialog.
5. Select the *landuse* layer and enter edit mode ( )
6. Check (under *View* ► *Toolbars*) that the *Advanced Digitizing* toolbar is enabled.
7. Zoom to this area (enable layers and labels if necessary):



8. Digitize this new (fictional) area shown in red:



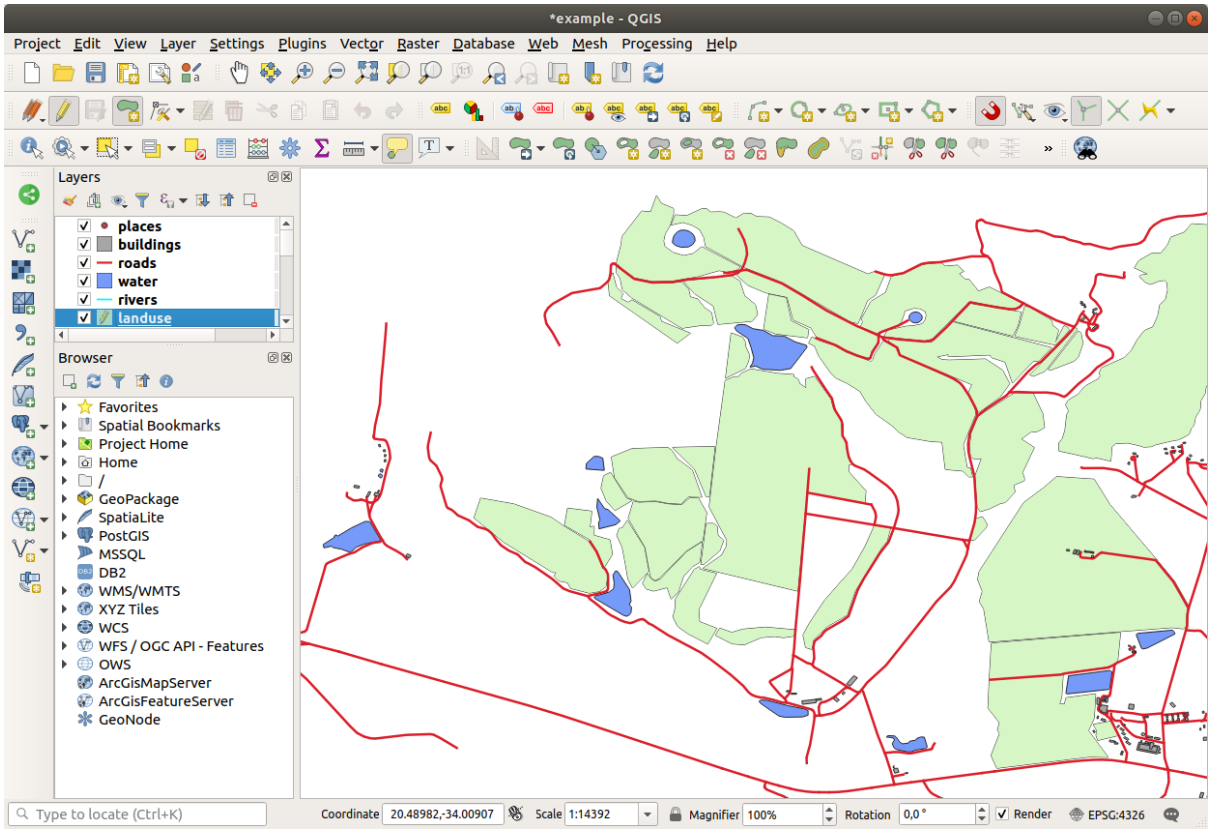
9. When prompted, give it an *OGC_FID* of `999`, but feel free to leave the other values unchanged.

   If you are careful while digitizing, and allow the cursor to snap to the vertices of adjoining areas, you'll notice that there won't be any gaps between your new area and the existing adjacent areas.

10. Note the undo and redo tools in the *Advanced Digitizing* toolbar.

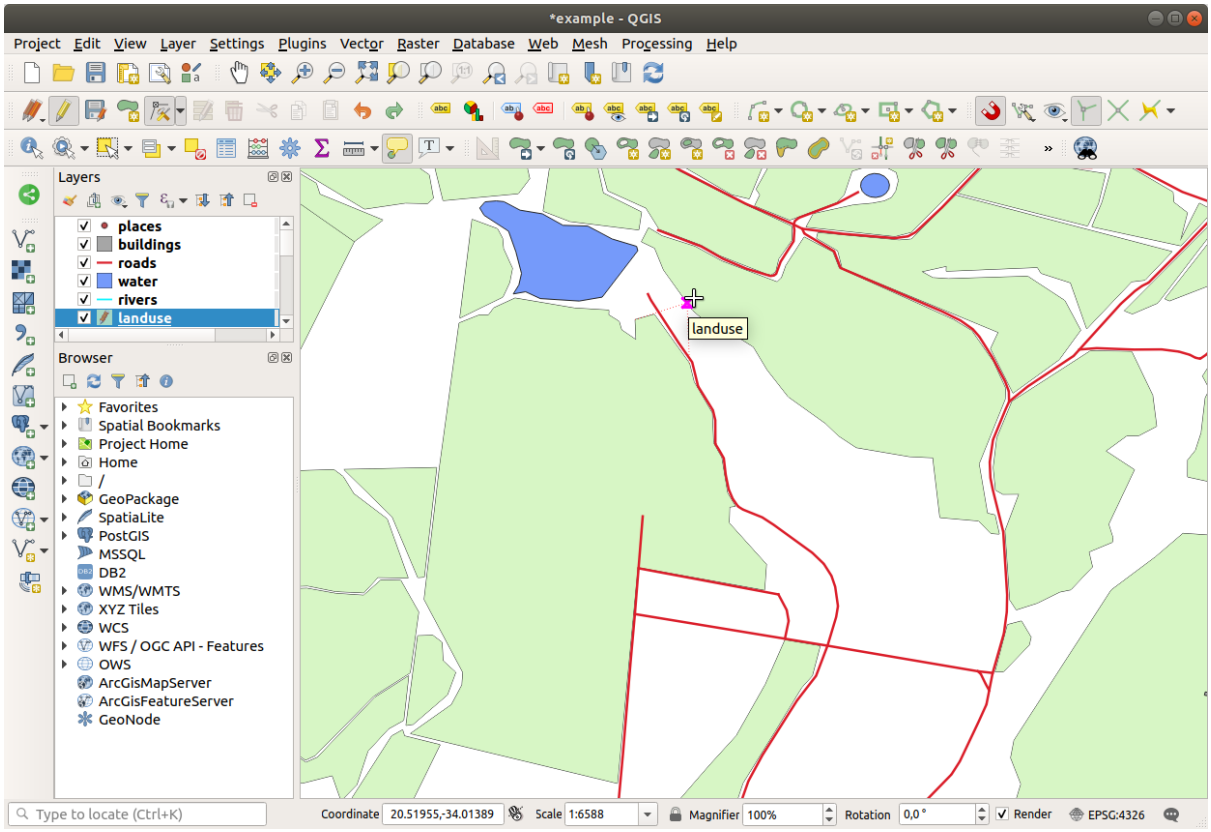
5.2.2 Follow Along: Correct Topological Features

Topology features can sometimes need to be updated. In our study area, an area has been turned into forest, so the `landuse` layer need an update. We will therefore expand and join some forest features in this area:



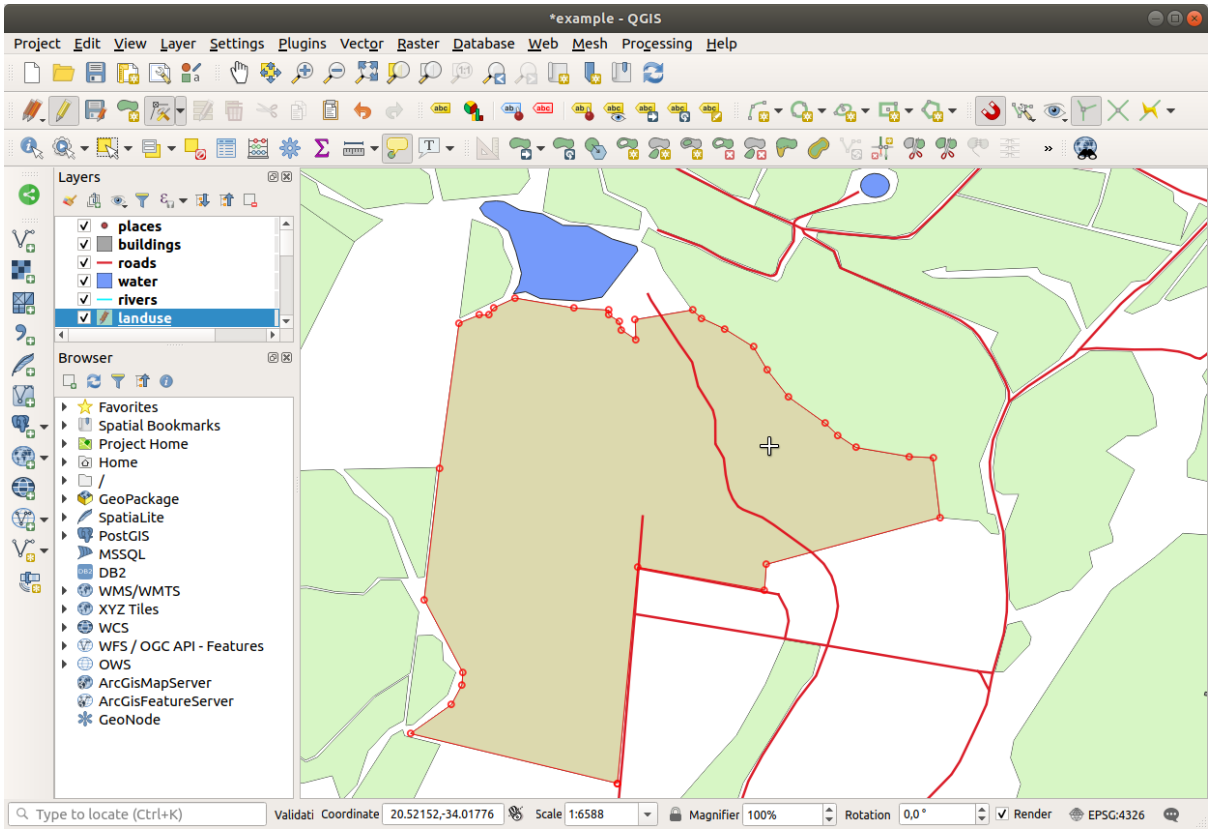
Instead of creating new polygons to join the forest areas, we are going to use the *Vertex Tool* to edit and join existing polygons.

1. Enter edit mode (if it is not active already)
2. Select the Vertex Tool tool.
3. Choose an area of forest, select a vertex, and move it to an adjoining vertex so that the two forest features meet:



4. Click on the other vertices and snap them into place.

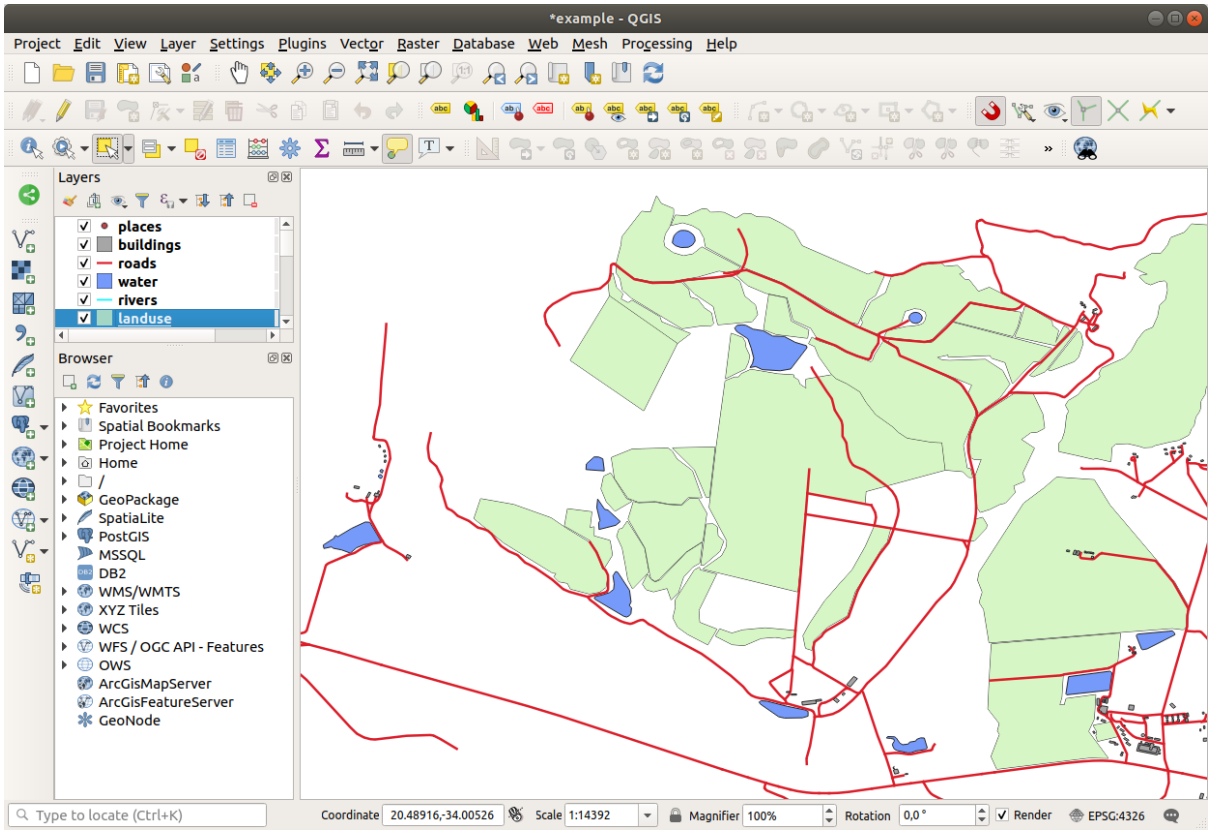
   The topologically correct border looks like this:



Go ahead and join a few more areas using the *Vertex Tool*.

You can also use the Add Polygon Feature tool to fill the gap between the two forest polygons. If you have enabled *Avoid overlap*, you don't have to add every single vertex - they will be added automatically if your new polygon overlaps the existing ones.

If you are using our example data, you should have a forest area looking something like this:

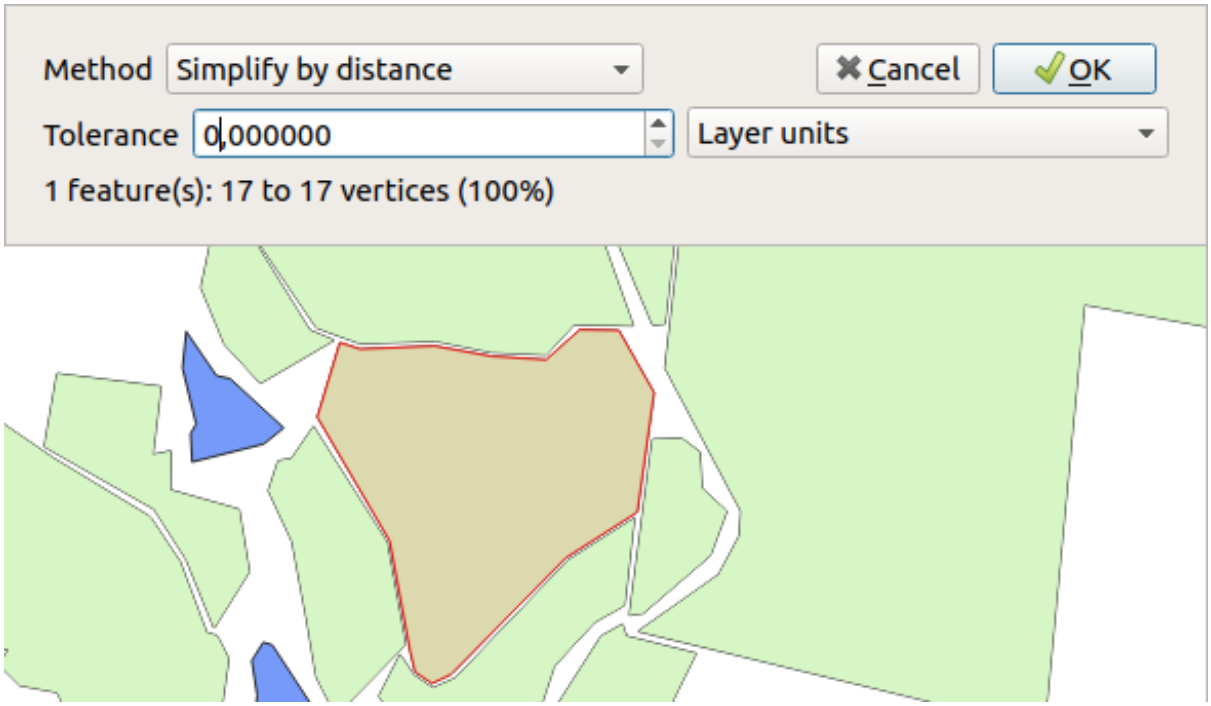


Don't worry if you have joined more, less or different areas of forest.

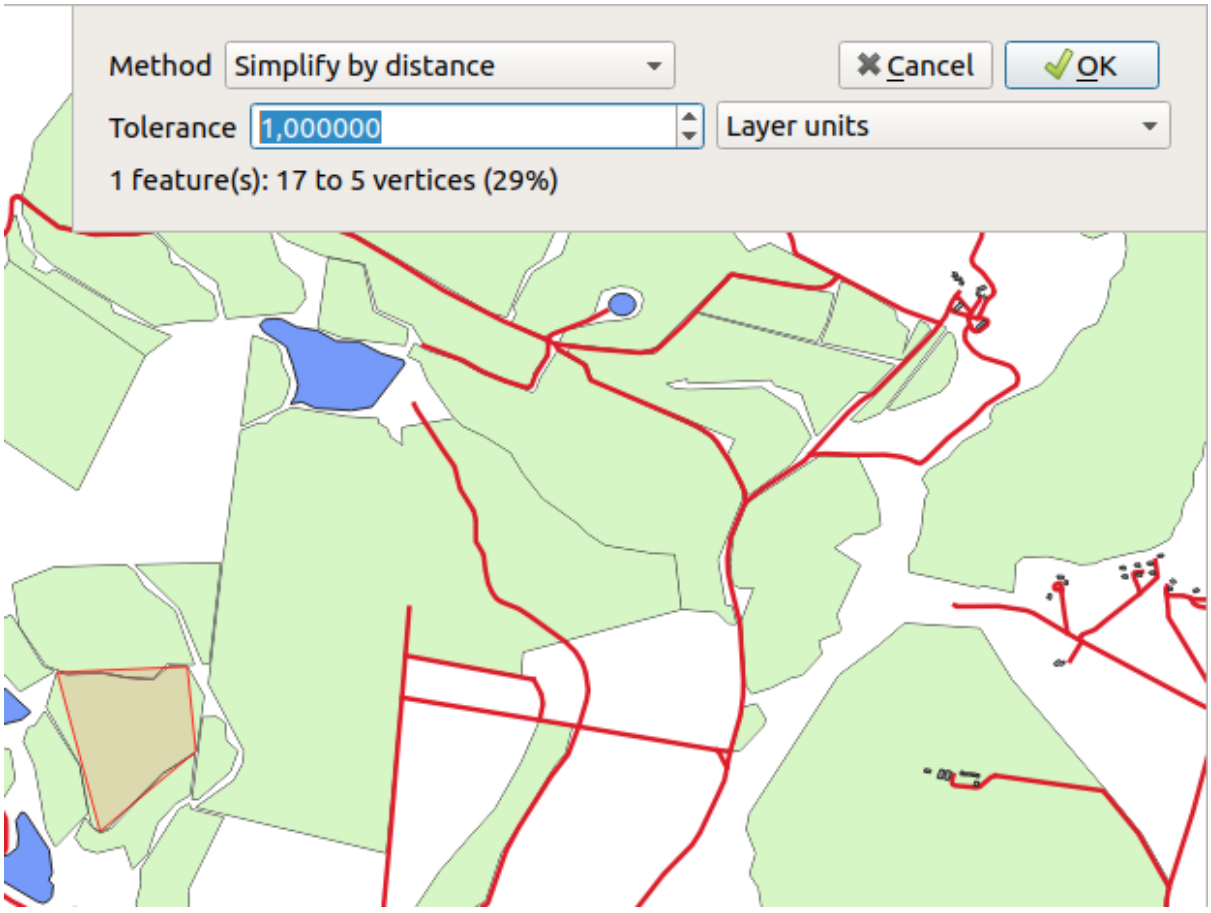
5.2.3 Follow Along: Tool: Simplify Feature

Continuing on the same layer, we will test the Simplify Feature tool:

1. Click on it to activate it.
2. Click on one of the areas which you joined using either the *Vertex Tool* or *Add Feature* tool. You will see this dialog:



3. Modify the *Tolerance* and watch what happens:



This allows you to reduce the number of vertices.

4. Klicken Sie auf *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.

Before you go on, set the polygon back to its original state by undoing the last change.

5.2.4 Try Yourself Tool: Add Ring

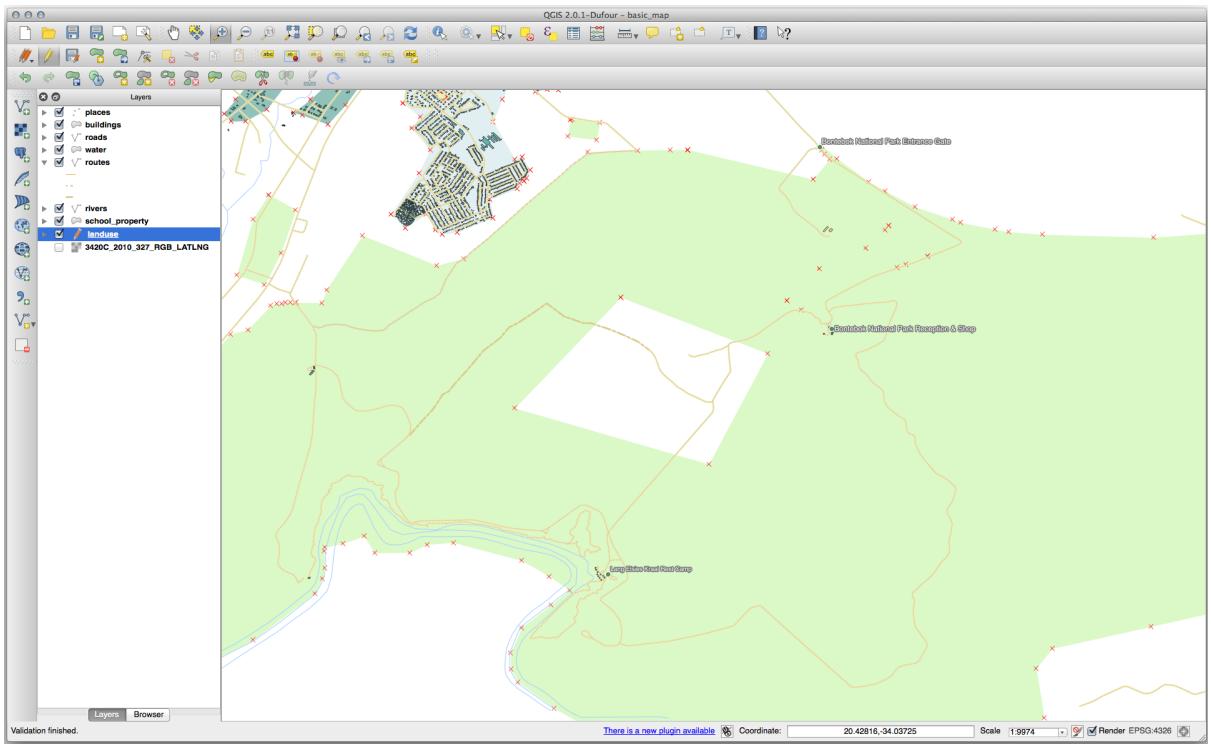
The Add Ring tool allows you to add an interior ring to a polygon feature (cut a hole in the polygon), as long as the hole is completely contained within the polygon (touching the boundary is OK). For example, if you have digitized the outer boundaries of South Africa and you need to add a hole for Lesotho, you would use this tool.

If you experiment with the tool, you may notice that the snapping options can prevent you from creating a ring inside a polygon. So you are advised to turn off snapping before cutting a hole.

1. Disable snapping for the `landuse` layer using the Enable Snapping button (or use the shortcut `s`).
2. Use the Add Ring tool to create a hole in the middle of a polygon geometry.
3. Draw a polygon over the target feature, as if you were using the Add polygon tool.
4. When you right-click, the hole will be visible.
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 Try Yourself Tool: Add Part

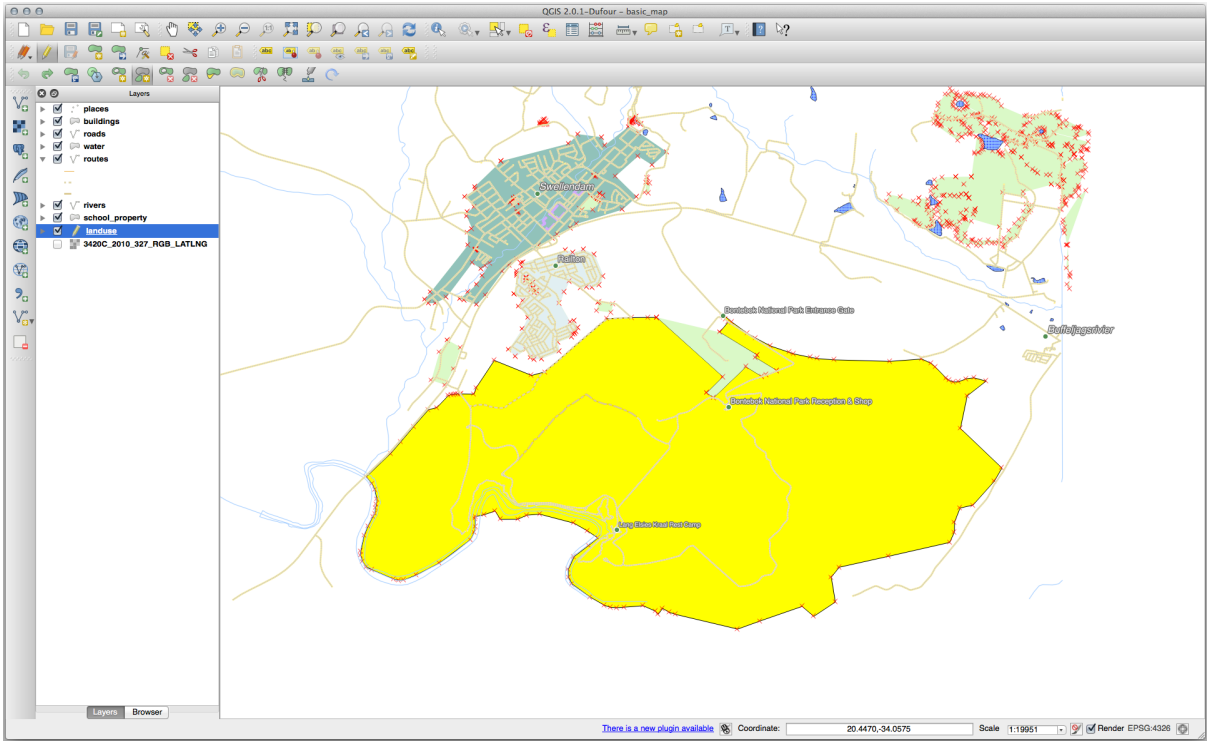
The Add Part tool allows you to add a new part to a feature, that is not directly connected to the main feature. For example, if you have digitized the boundaries of mainland South Africa, but you haven't yet added the Prince Edward Islands, you would use this tool to create them.

1. Select the polygon to which you wish to add the part by using the Select Features by area or single click tool.
2. Use the *Add Part* tool to add an outlying area.
3. Delete the part you just created using the Delete Part tool.

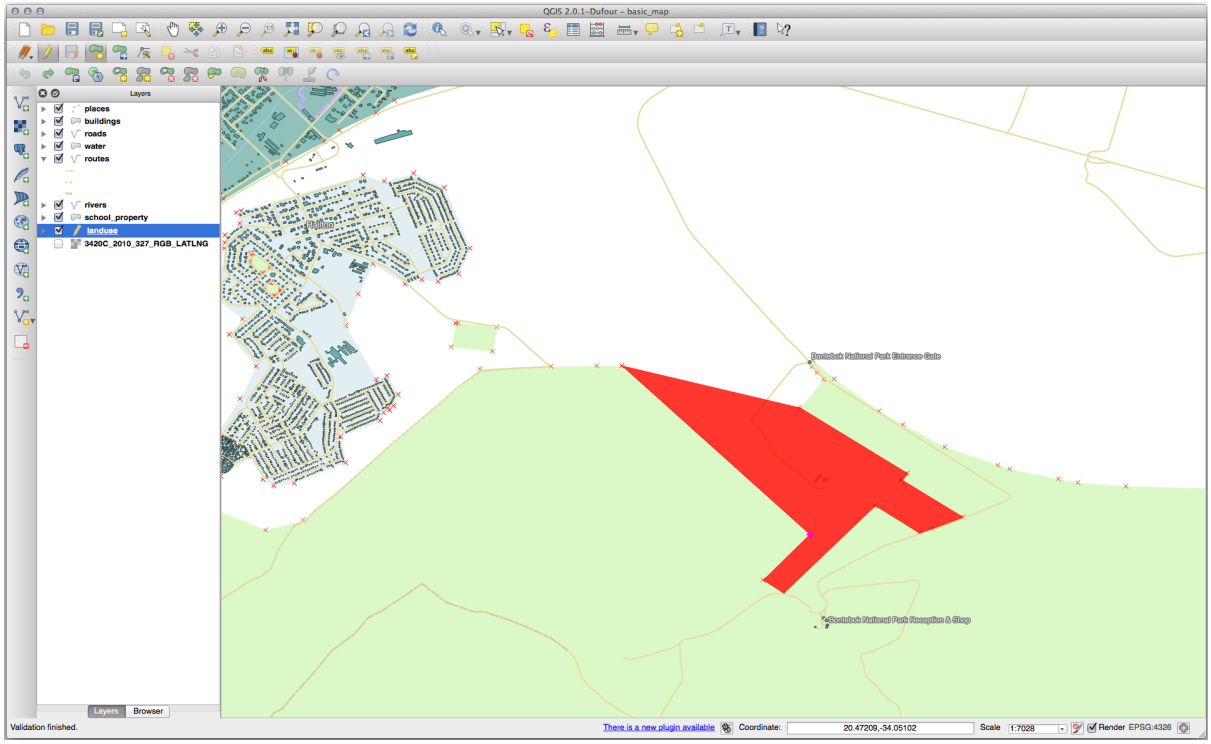
> **Bemerkung:** Click inside the part to delete it.

Answer

1. First select the Bontebok National Park:



2. Now add your new part:



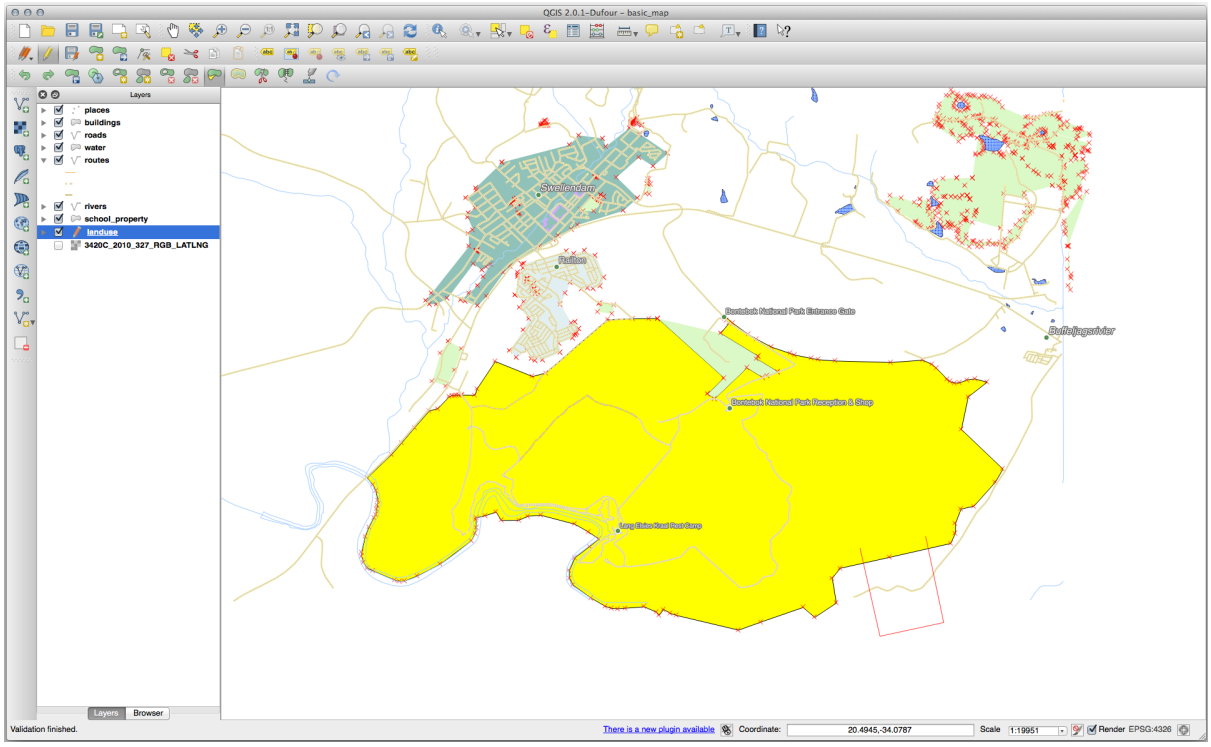
3. Undo your edit before continuing with the exercise for the next tool.

5.2.6 Follow Along: Tool: Reshape Features

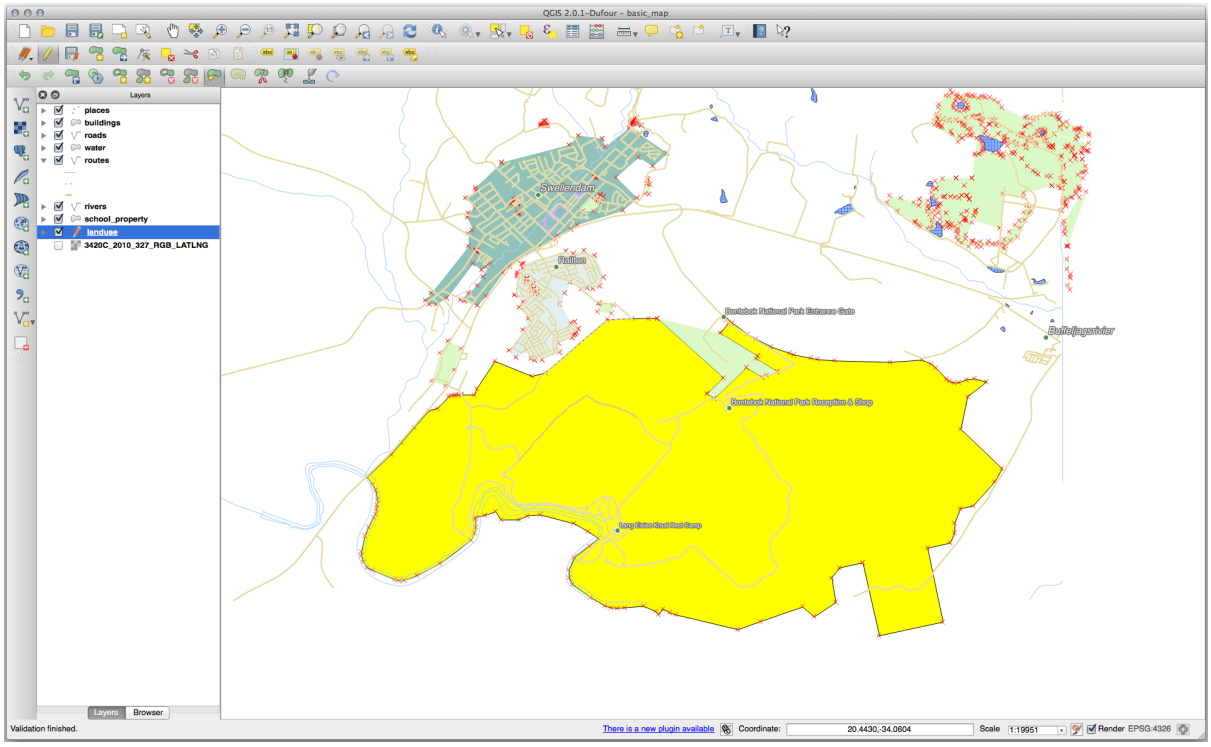
The Reshape Features tool is used to extend a polygon feature or cut away a part of it (along the boundary).

Extending:

1. Select the polygon using the Select Features by area or single click tool.
2. Left-click inside the polygon to start drawing.
3. Draw a shape outside the polygon. The last vertex should be back inside the polygon.
4. Right-click to finish the shape:



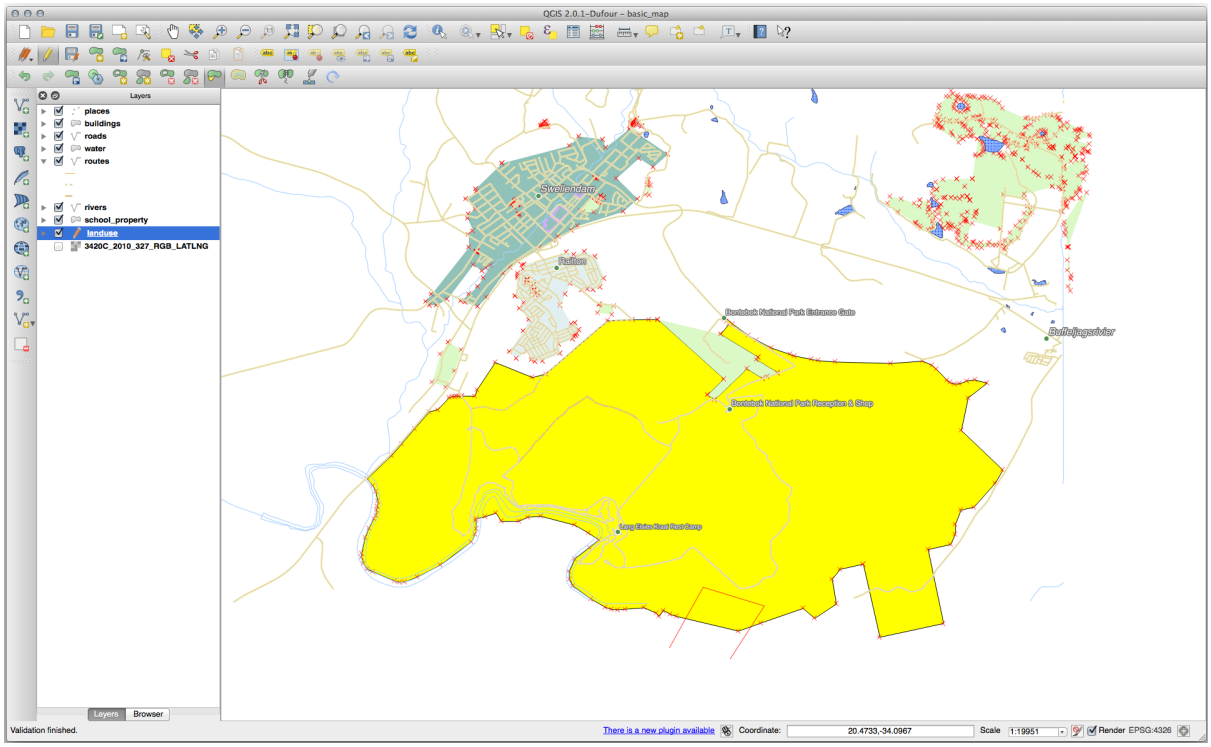
This will give a result similar to:



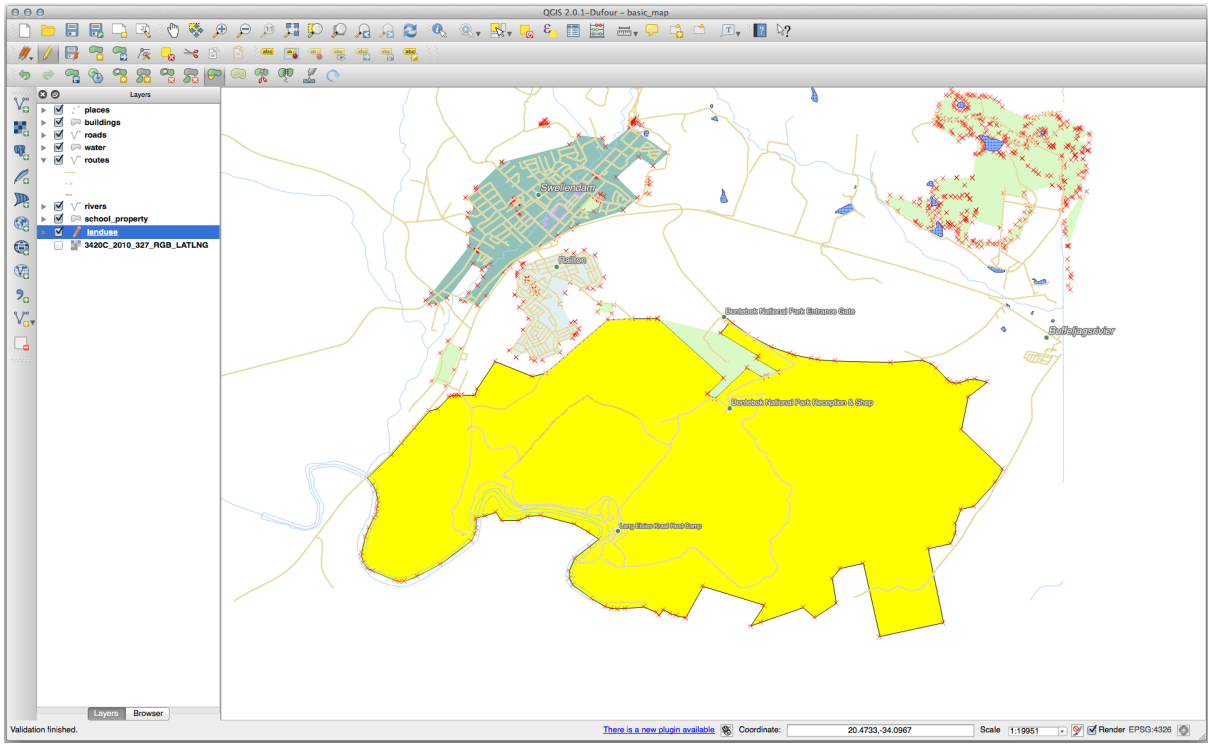
Cut away a part:

1. Select the polygon using the Select Features by area or single click tool.
2. Click outside the polygon.
3. Draw a shape inside the polygon. The last vertex must be back outside the polygon.

4. Right-click outside the polygon:



The result of the above:

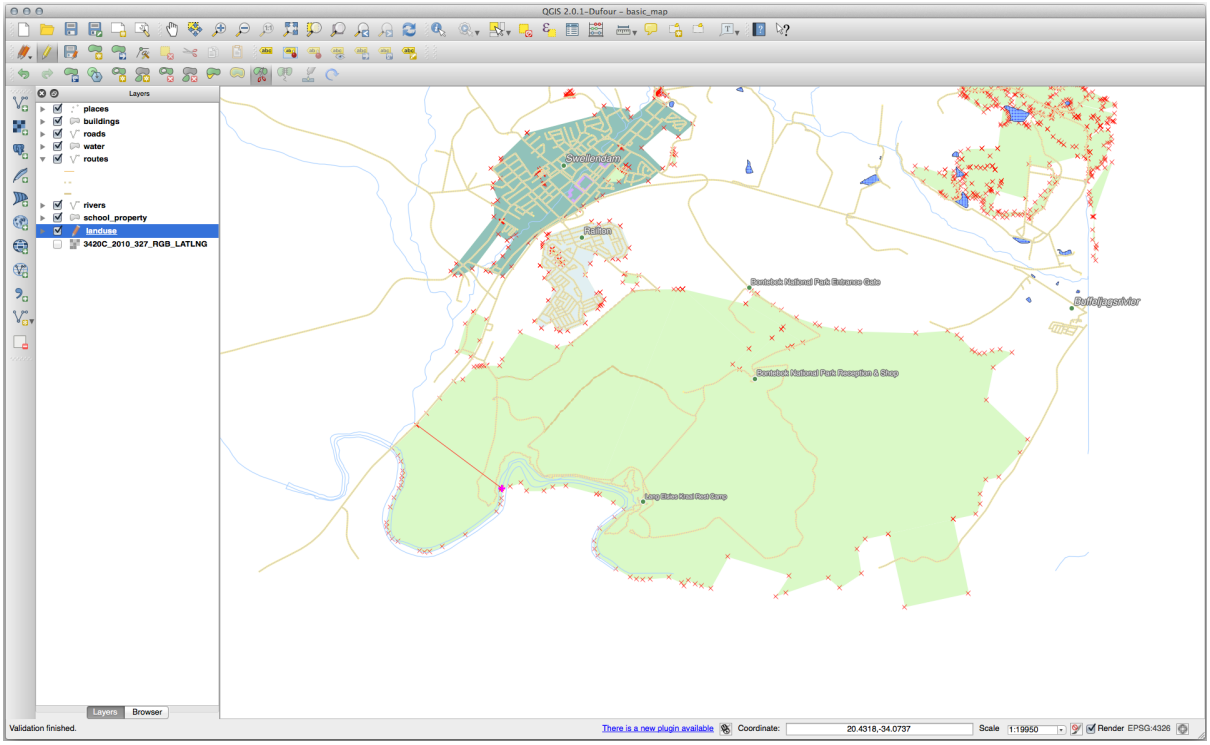


5.2.7 Try Yourself Tool: Split Features

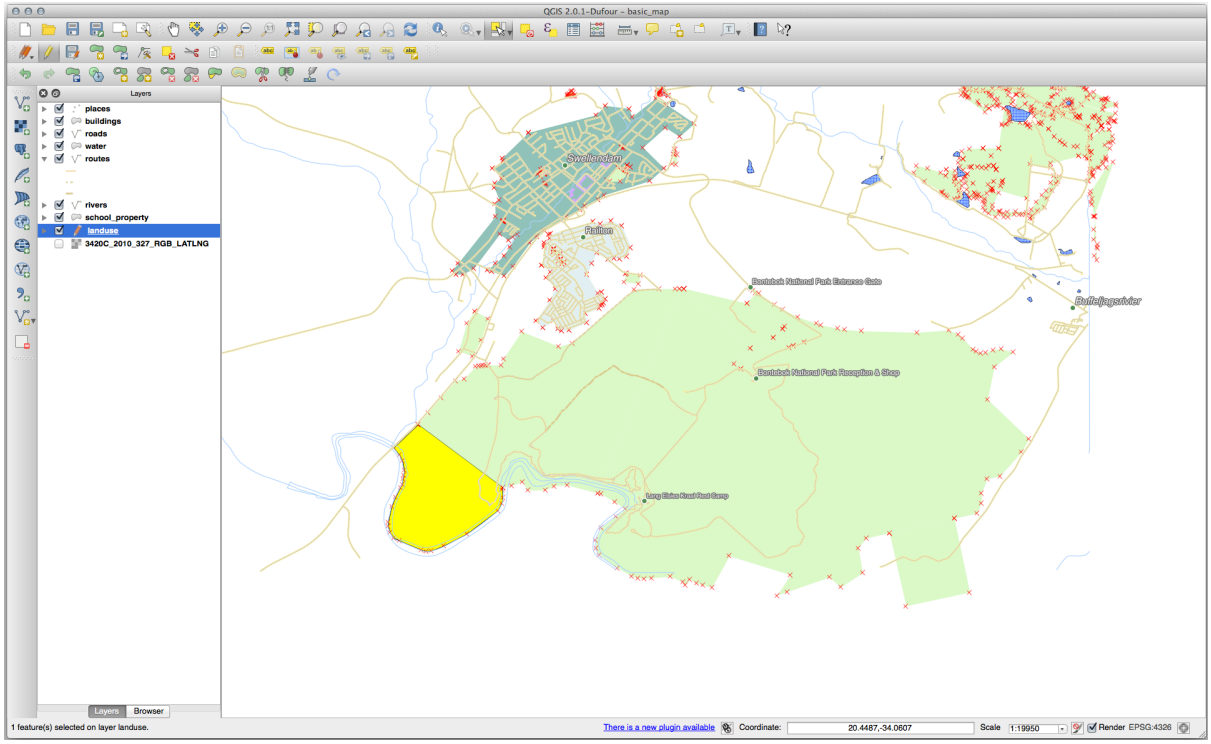
The Split Features tool is similar to the Reshape Features tool, except that it does not delete either of the two parts. Instead, it keeps them both.

We will use the tool to split a corner from a polygon.

1. First, select the `landuse` layer and re-enable snapping for it.
2. Select the Split Features tool and click on a vertex to begin drawing a line.
3. Draw the bounding line.
4. Click a vertex on the „opposite“ side of the polygon you wish to split and right-click to complete the line:



5. At this point, it may seem as if nothing has happened. But remember that the `landuse` layer is rendered without border lines, so the new division line will not be shown.
6. Use the Select Features by area or single click tool to select the part you just split out; the new feature will now be highlighted:



5.2.8 Try Yourself Tool: Merge Features

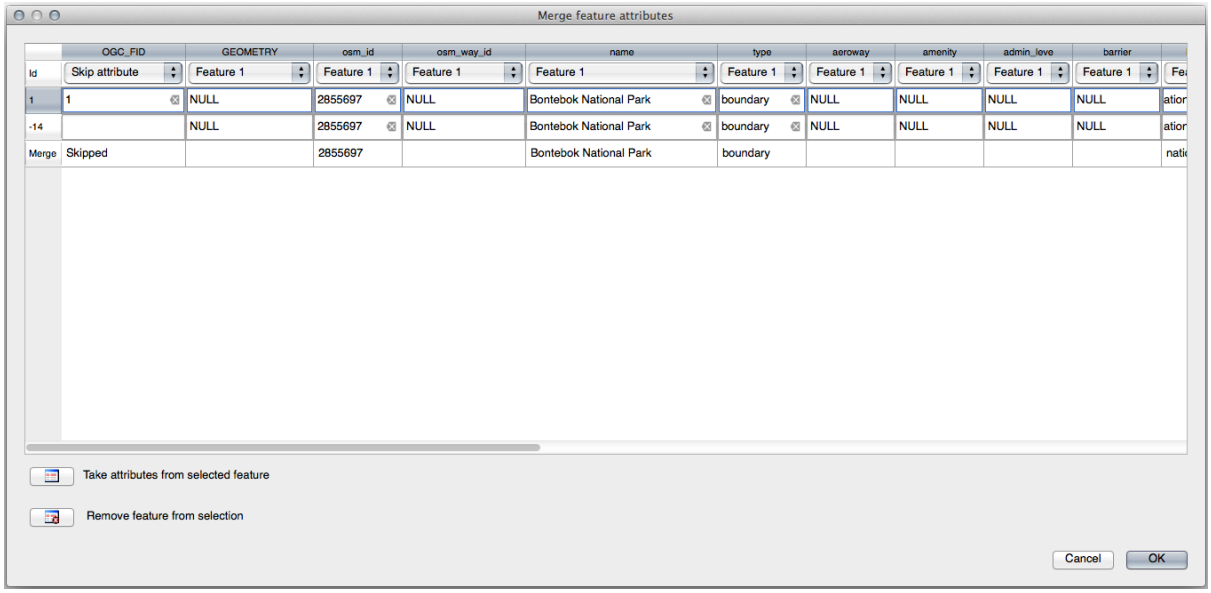
Now we will re-join the feature you just split out to the remaining part of the polygon:

1. Experiment with the Merge Selected Features and Merge Attributes of Selected Features tools.
2. Note the differences.

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 In Conclusion

Topology editing is a powerful tool that allows you to create and modify objects quickly and easily, while ensuring that they remain topologically correct.

5.2.10 What's Next?

Now you know how to digitize the shape of the objects easily, but adding attributes is still a bit of a headache! Next we will show you how to use forms, making attribute editing simpler and more effective.

5.3 Lesson: Formulare

When you add new data via digitizing, you're presented with a dialog that lets you fill in the attributes for that feature. However, this dialog is not, by default, very nice to look at. This can cause a usability problem, especially if you have large datasets to create, or if you want other people to help you digitize and they find the default forms to be confusing.

Fortunately, QGIS lets you create your own custom dialogs for a layer. This lesson shows you how.

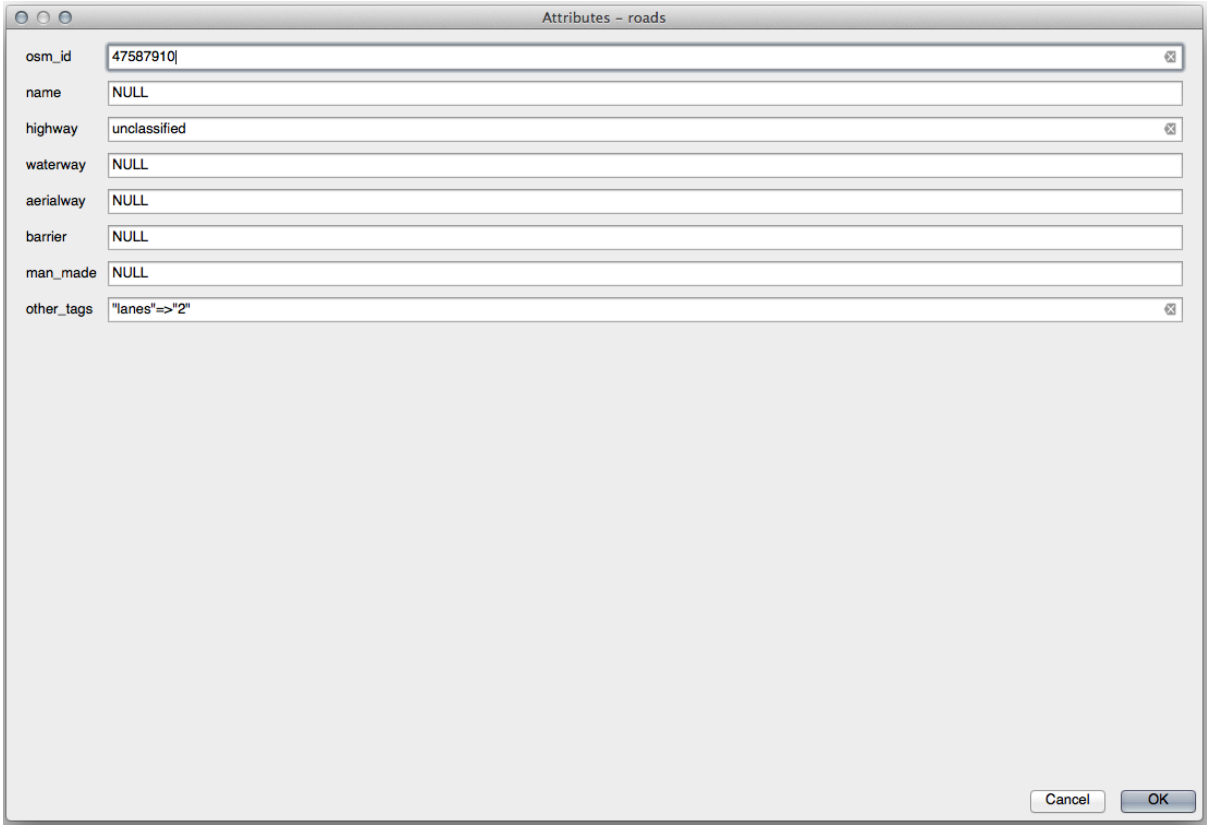
The goal for this lesson: To create a form for a layer.

5.3.1 Follow Along: Using QGIS' Form Design Functionality

1. Select the `roads` layer in the *Layers* panel
2. Enter *Edit Mode* as before
3. Open the `roads` layer's attribute table
4. Right-click on any cell in the table. A short menu will appear, that includes the *Open form* entry.
5. Click on it to see the form that QGIS generates for this layer

Obviously it would be nice to be able to do this while looking at the map, rather than needing to search for a specific street in the *Attribute Table* all the time.

1. Select the *roads* layer in the *Layers* panel
2. Using the Identify Features tool, click on any street in the map.
3. The *Identify Results* panel opens and shows a tree view of the fields values and other general information about the clicked feature.
4. At the top of the panel, check the *Auto open form for single feature results* checkbox in the Identify Settings menu.
5. Now, click again on any street in the map. Along the previous *Identify Results* dialog, you'll see the now-familiar form:

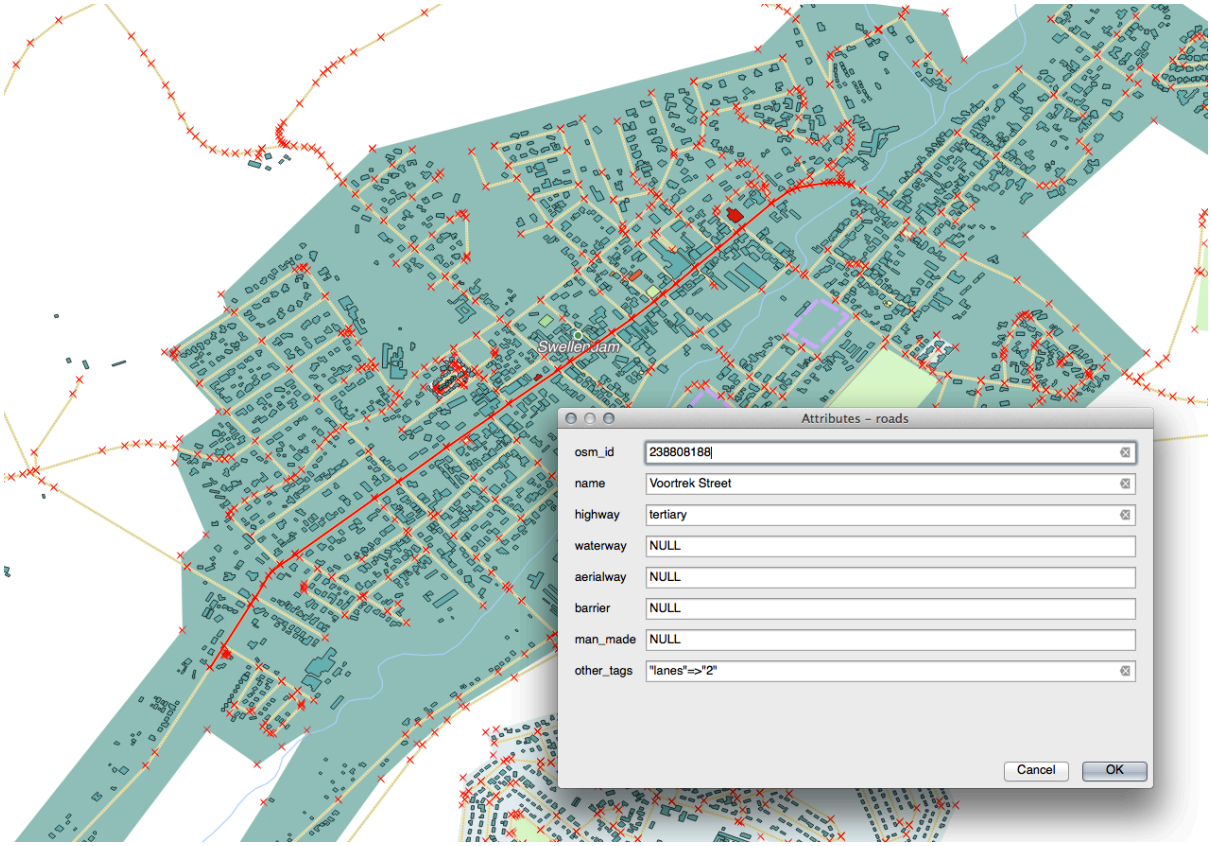


6. Each time you click on a single feature with the *Identify* tool, its form pops up as long as the *Auto open form* is checked.

5.3.2 Try Yourself Using the Form to Edit Values

If you are in edit mode, you can use this form to edit a feature's attributes.

1. Activate edit mode (if it isn't already activated).
2. Using the Identify Features tool, click on the main street running through Swellendam:

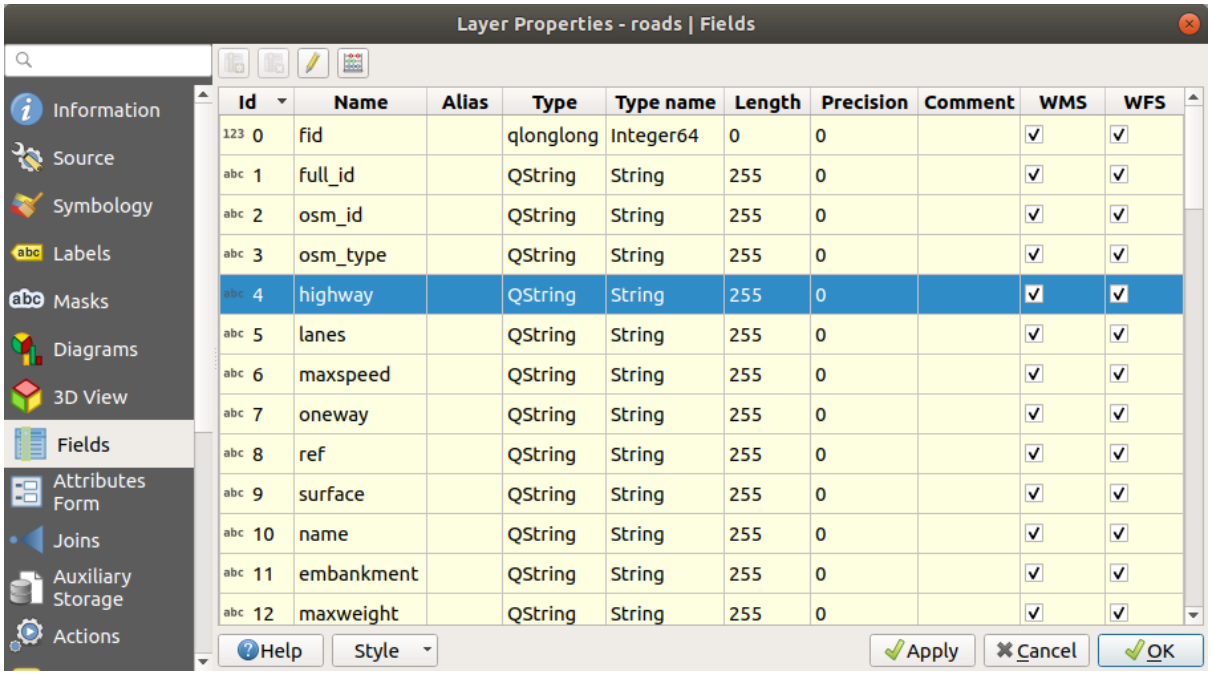


3. Edit its *highway* value to be `secondary`
4. Exit edit mode and save your edits
5. Open the *Attribute Table* and note that the value has been updated in the attributes table and therefore in the source data

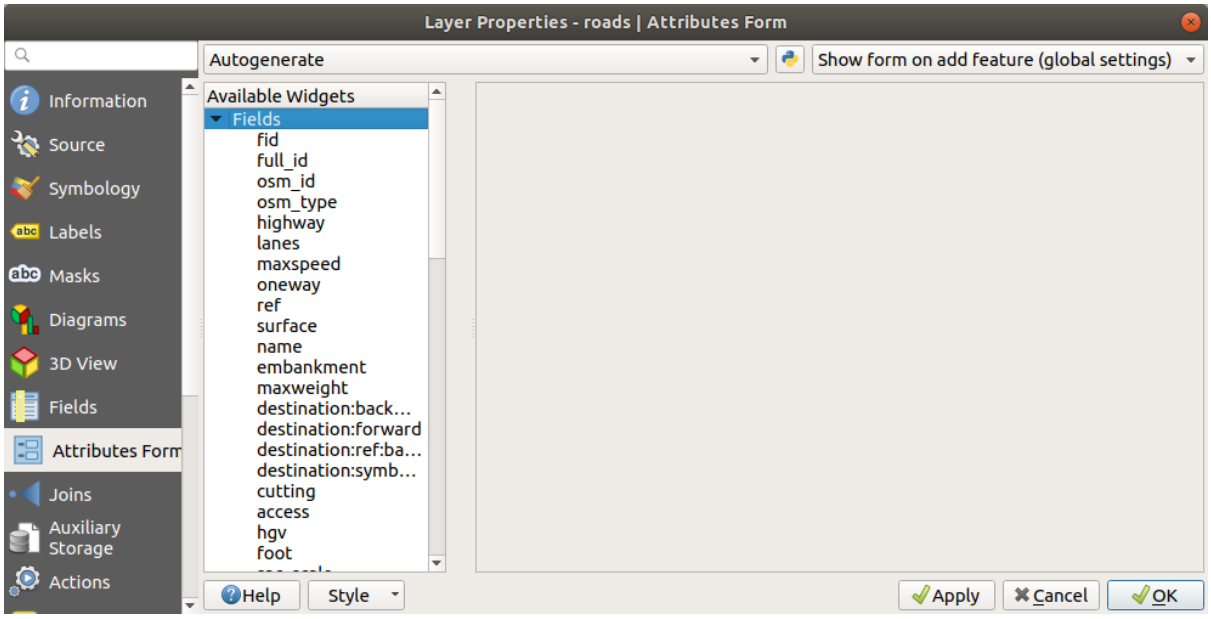
5.3.3 Follow Along: Setting Form Field Types

It's nice to edit things using a form, but you still have to enter everything by hand. Fortunately, forms have different kinds of so-called *widgets* that allow you to edit data in various different ways.

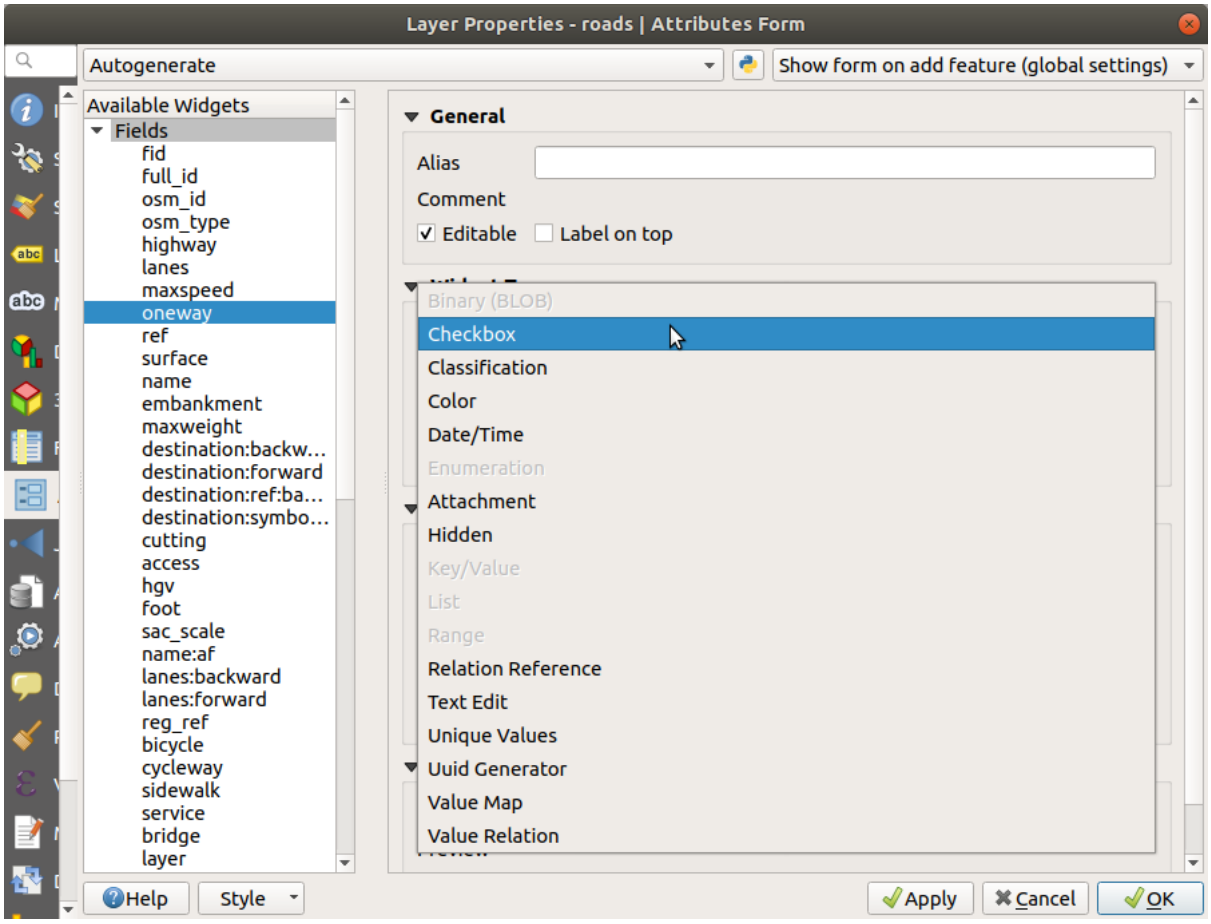
1. Open the *roads* layer's *Properties…*
2. Switch to the *Fields* tab. You'll see this:



3. Switch to the *Attributes Form* tab. You'll see this:



4. Click on the `oneway` row and choose *Checkbox* as *Widget Type* in the list of options:



5. Klicken Sie auf *OK*
6. Enter edit mode (if the `roads` layer is not already in edit mode)
7. Click on the Identify Features tool
8. Click on the same main road you chose earlier

You will now see that the *oneway* attribute has a checkbox next to it denoting `True` (checked) or `False` (unchecked).

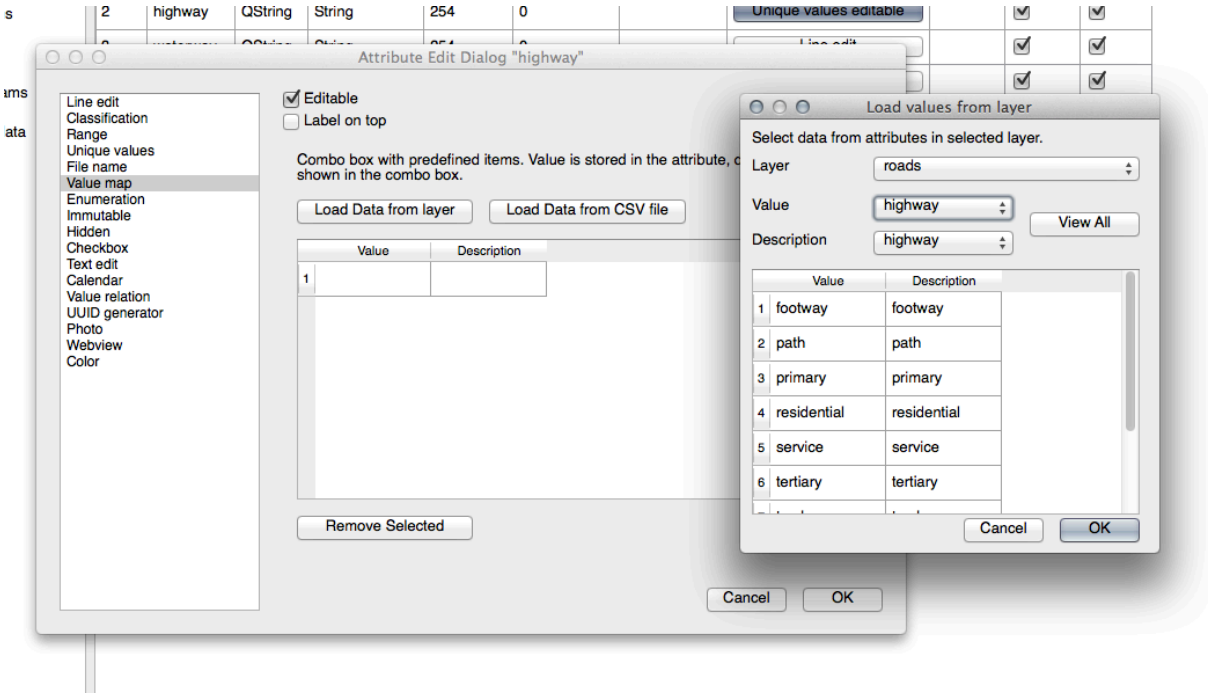
5.3.4 Try Yourself

Set a more appropriate form widget for the *highway* field.

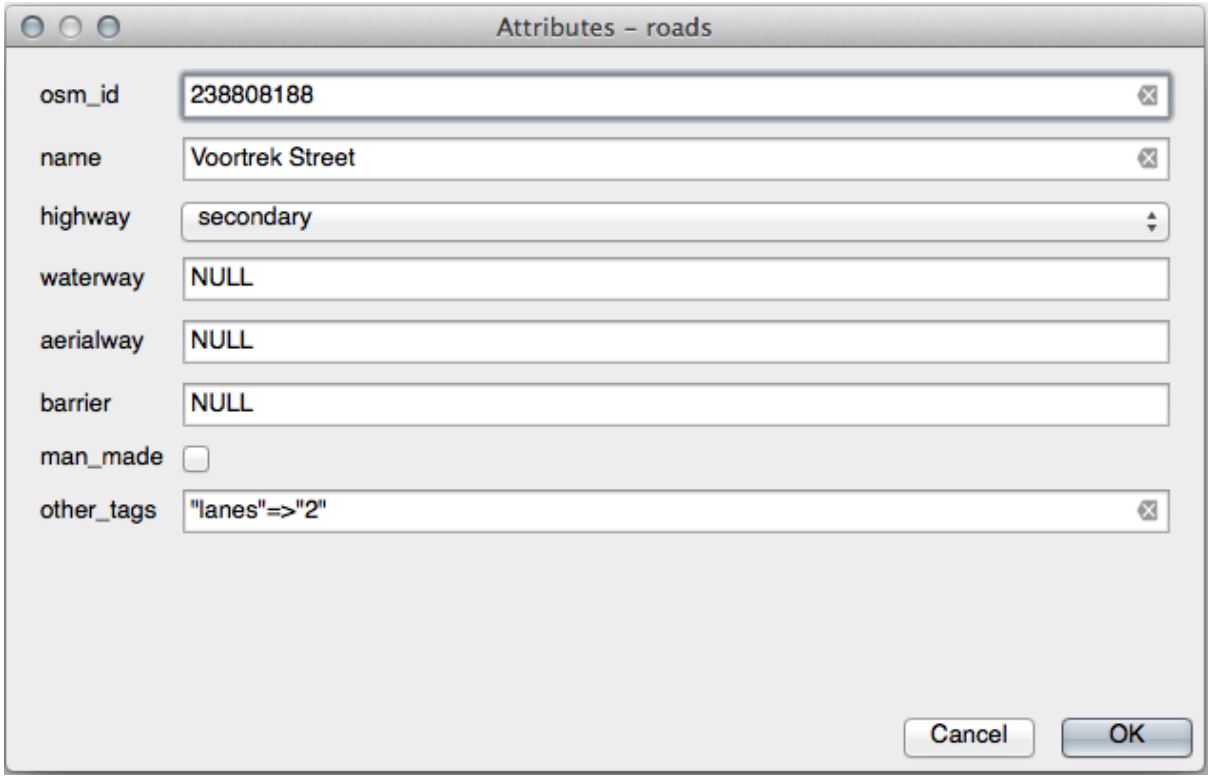
Answer

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:



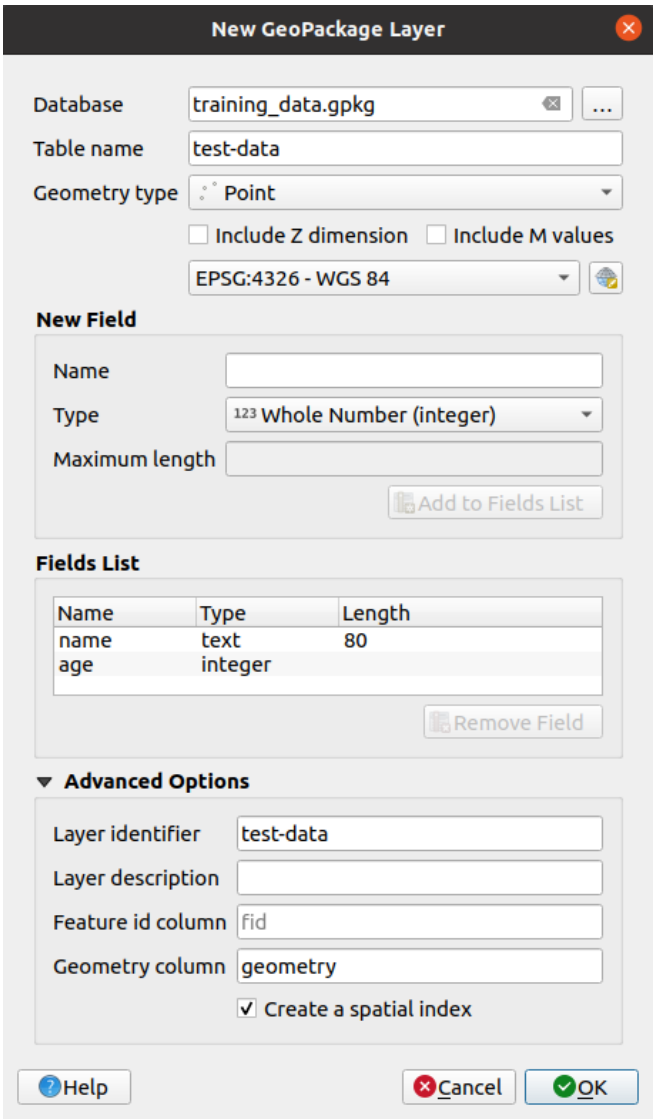
3. Click *OK* three times.
4. If you use the *Identify* tool on a street now while edit mode is active, the dialog you get should look like this:



5.3.5 Try Yourself Creating Test Data

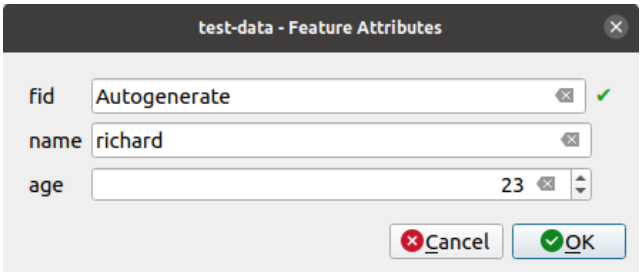
You can also design your own custom form completely from scratch.

1. Create a simple point layer named `test-data` with two attributes:
   - name (text)
   - age (integer)



2. Capture a few points on your new layer using the digitizing tools so that you have a little data to play with. You should be presented with the default QGIS generated attribute capture form each time you capture a new point.

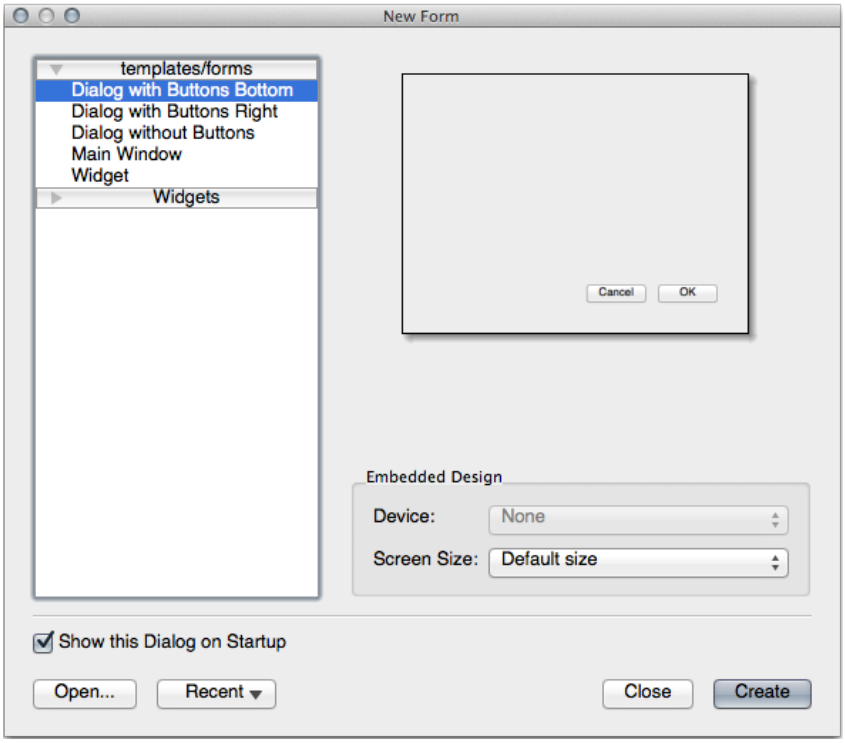
Bemerkung: You may need to disable Snapping if still enabled from earlier tasks.



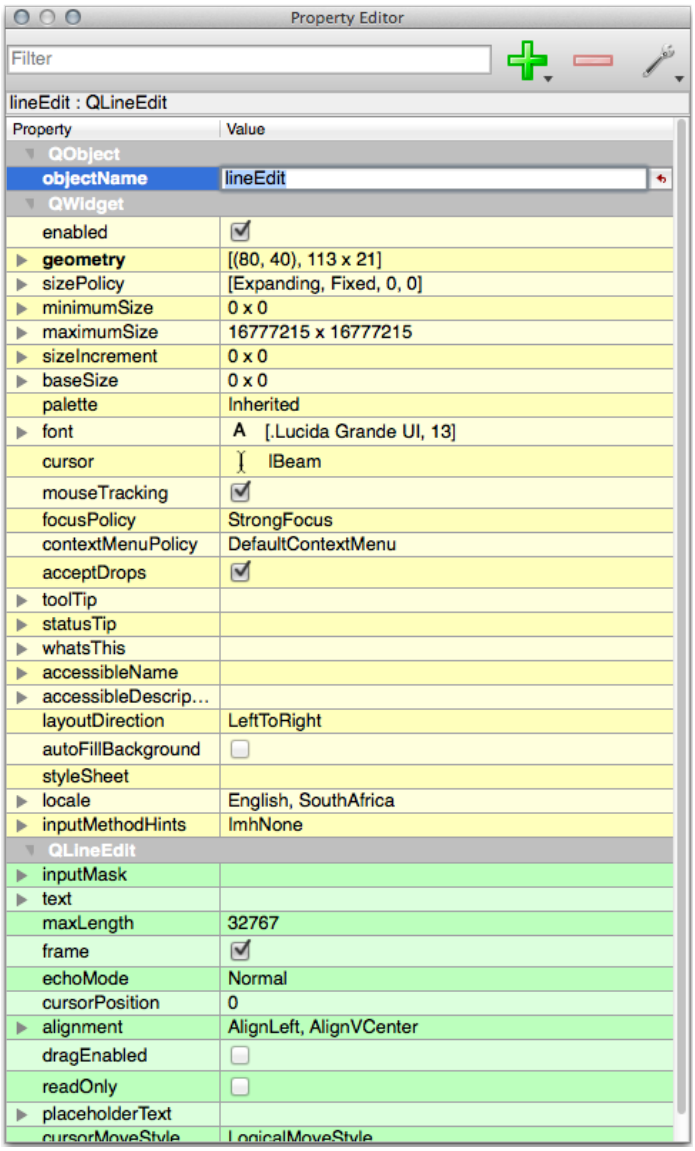
5.3.6 Follow Along: Creating a New Form

Now we want to create our own custom form for the attribute data capture phase. To do this, you need to have *QT Designer* installed (only needed for the person who creates the forms).

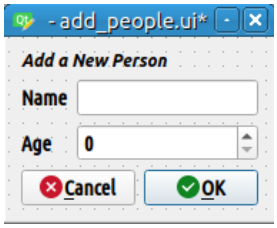
1. Start *QT Designer*.
2. In the dialog that appears, create a new dialog:



3. Look for the *Widget Box* along the left of your screen (default). It contains an item called *Line Edit*.
4. Click and drag this item into your form. This creates a new *Line Edit* in the form.
5. With the new line edit element selected, you'll see its *properties* along the side of your screen (on the right by default):



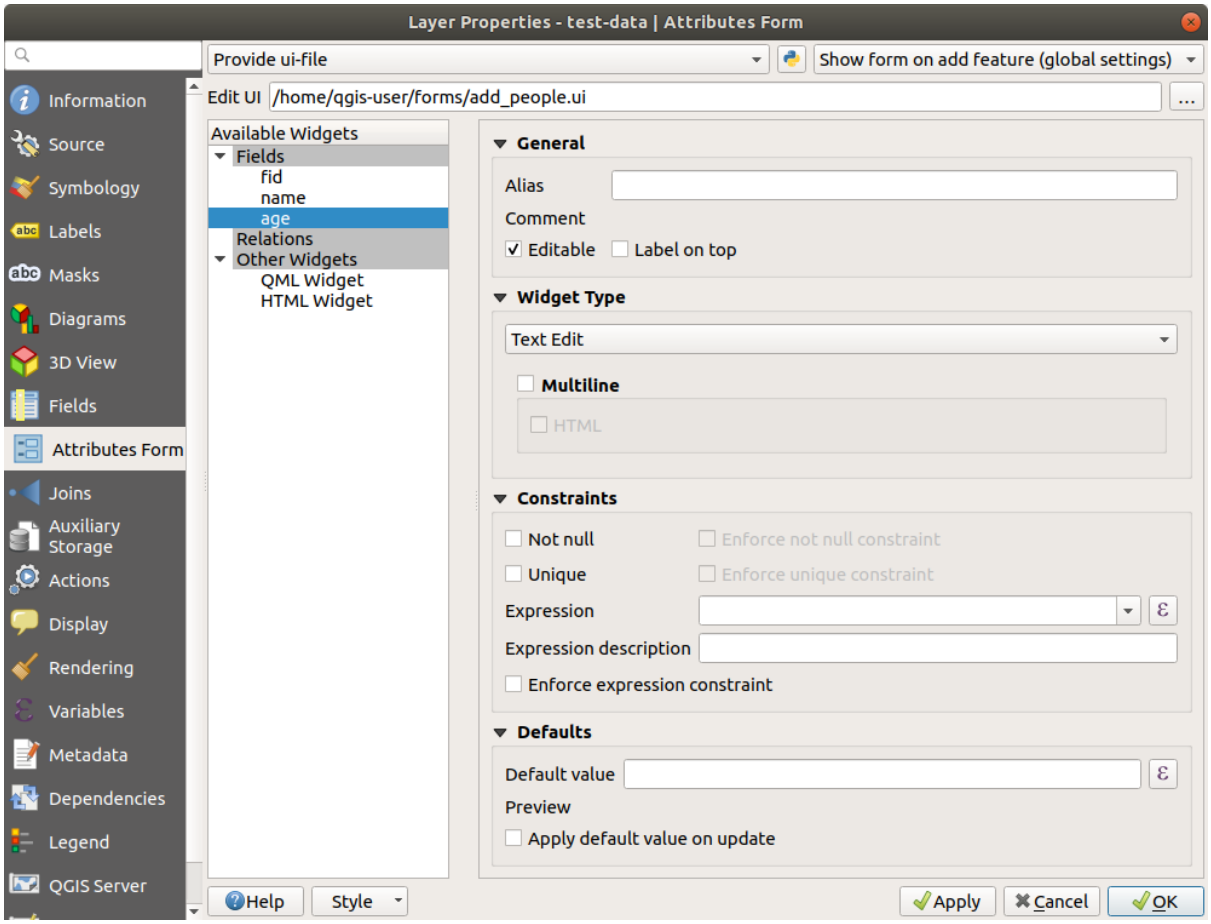
6. Set its name to `name`.
7. Using the same approach, create a new *Spin Box* and set its name to `age`.
8. Add a *Label* with the text `Add a New Person` in a bold font (look in the object *properties* to find out how to set this). Alternatively, you may want to set the title of the dialog itself (rather than adding a label).
9. Add a *Label* for your *Line Edit* and your *Spin Box*.
10. Arrange the elements to your own desire.
11. Click anywhere in your dialog.
12. Find the *Lay Out in a Form Layout* button (in a toolbar along the top edge of the screen, by default). This lays out your dialog automatically.
13. Set the dialog's maximum size (in its properties) to `200` (width) by `150` (height).
14. Your form should now look similar to this:



15. Save your new form as `exercise_data/forms/add_people.ui`
16. When it's done saving, you can close *Qt Designer*

5.3.7 Follow Along: Associating the Form with Your Layer

1. Go back to QGIS
2. Double click the *test-data* layer in the legend to access its properties.
3. Click on the *Attributes Form* tab in the *Layer Properties* dialog
4. In the *Attribute editor layout* dropdown, select *Provide ui-file*.
5. Click the ellipsis button and choose the `add_people.ui` file you just created:



6. Click *OK* on the *Layer Properties* dialog
7. Enter edit mode and capture a new point

8. When you do so, you will be presented with your custom dialog (instead of the generic one that QGIS usually creates).
9. If you click on one of your points using the Identify Features tool, you can now bring up the form by right clicking in the identify results window and choosing *View Feature Form* from the context menu.
10. If you are in edit mode for this layer, that context menu will show *Edit Feature Form* instead, and you can then adjust the attributes in the new form even after initial capture.

5.3.8 In Conclusion

Using forms, you can make life easier for yourself when editing or creating data. By editing widget types or creating an entirely new form from scratch, you can control the experience of someone who digitizes new data for that layer, thereby minimizing misunderstandings and unnecessary errors.

5.3.9 Further Reading

If you completed the advanced section above and have knowledge of Python, you may want to check out this blog entry about creating custom feature forms with Python logic, which allows advanced functions including data validation, autocompletion, etc.

5.3.10 What's Next?

Opening a form on identifying a feature is one of the standard actions that QGIS can perform. However, you can also direct it to perform custom actions that you define. This is the subject of the next lesson.

5.4 Lesson: Aktionen

Now that you have seen a default action in the previous lesson, it is time to define your own actions.

An action is something that happens when you click on a feature. It can add a lot of extra functionality to your map, allowing you to retrieve additional information about an object, for example. Assigning actions can add a whole new dimension to your map!

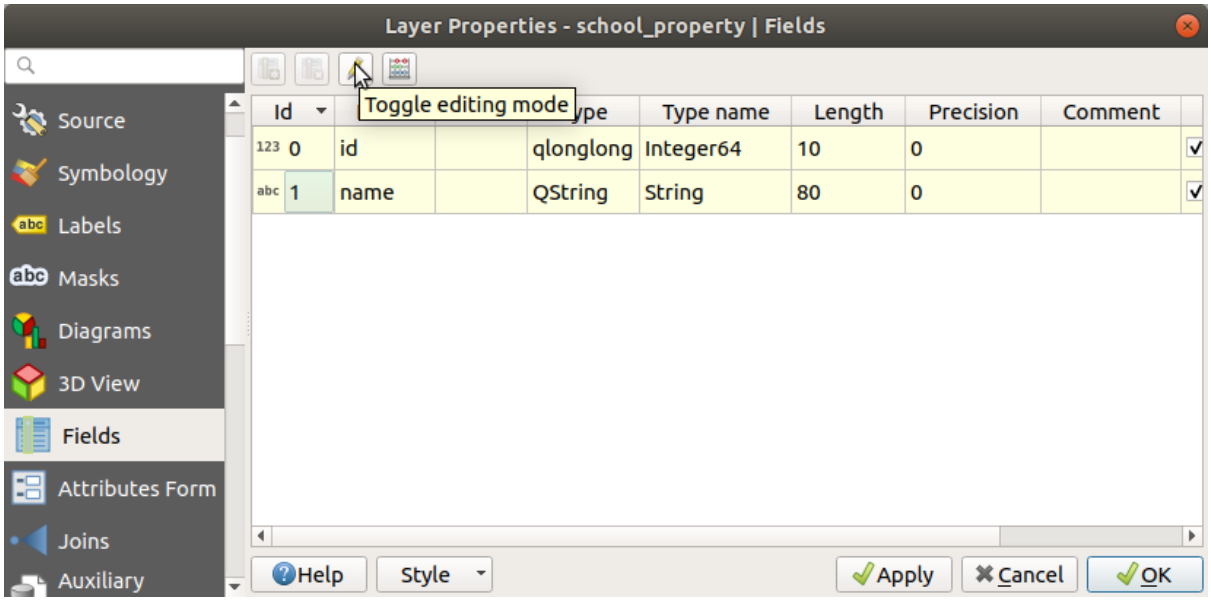
The goal for this lesson: To learn how to add custom actions.

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

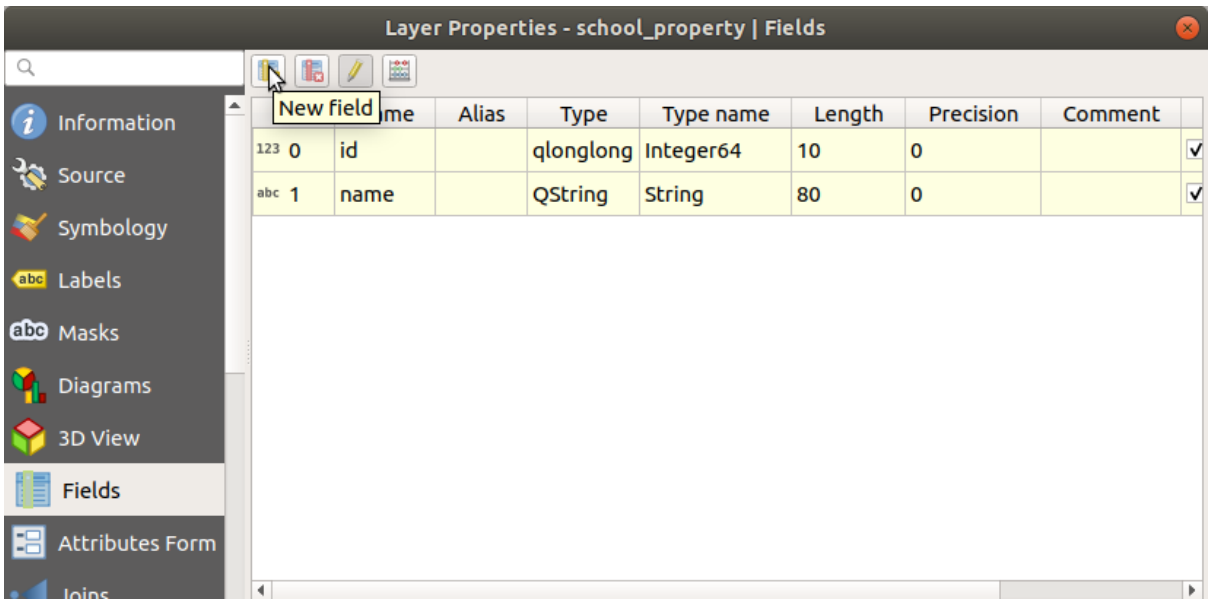
5.4.1 Follow Along: Add a Field for Images

The `school_property` layer has no way to associate an image with a property yet. First we will create a field for this purpose.

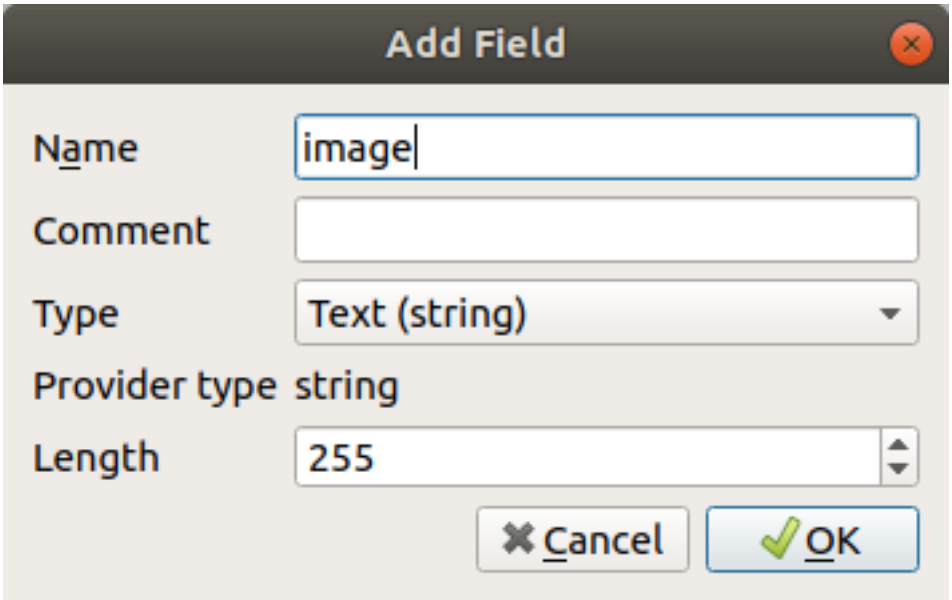
1. Open the *Layer Properties* dialog.
2. Click on the *Fields* tab.
3. Toggle editing mode:



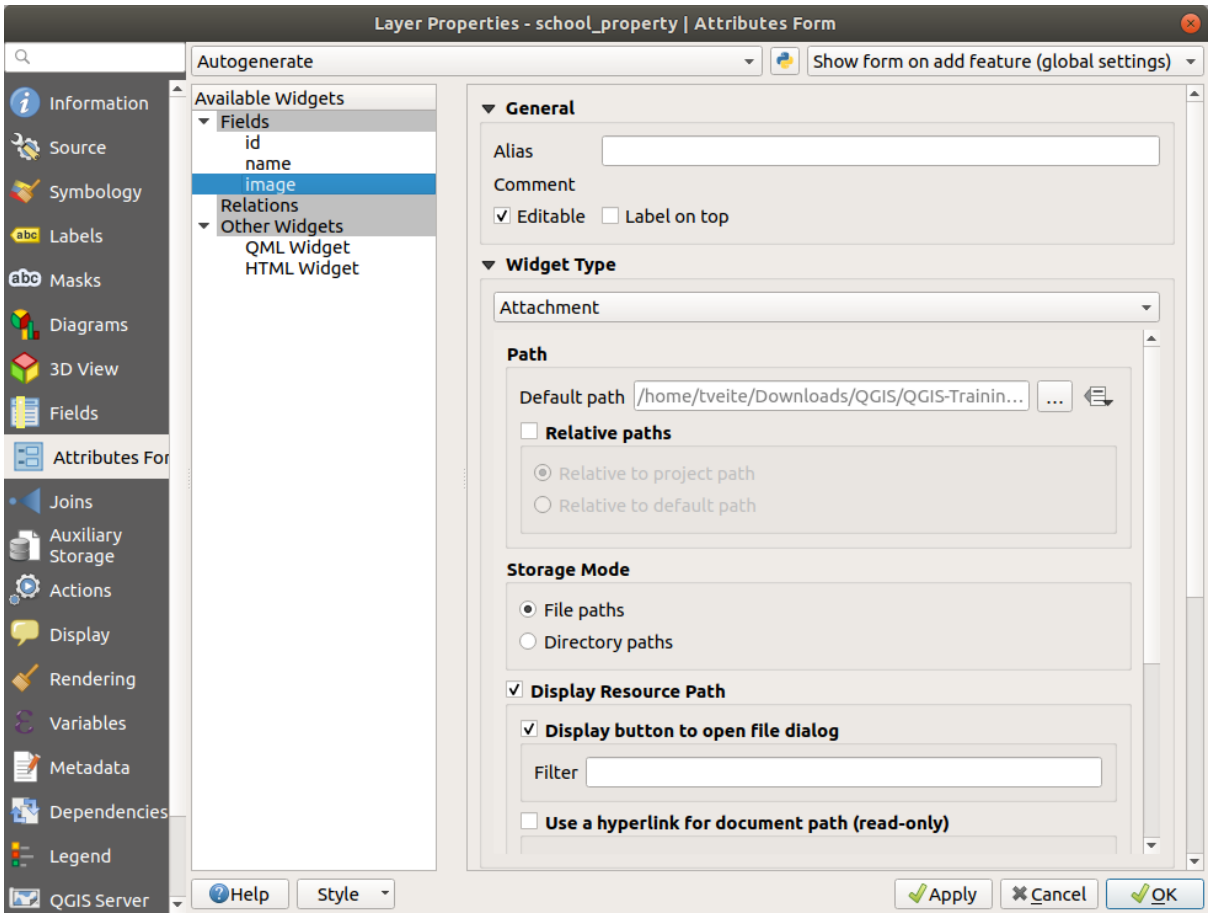
4. Add a new column:



5. Enter the values below:

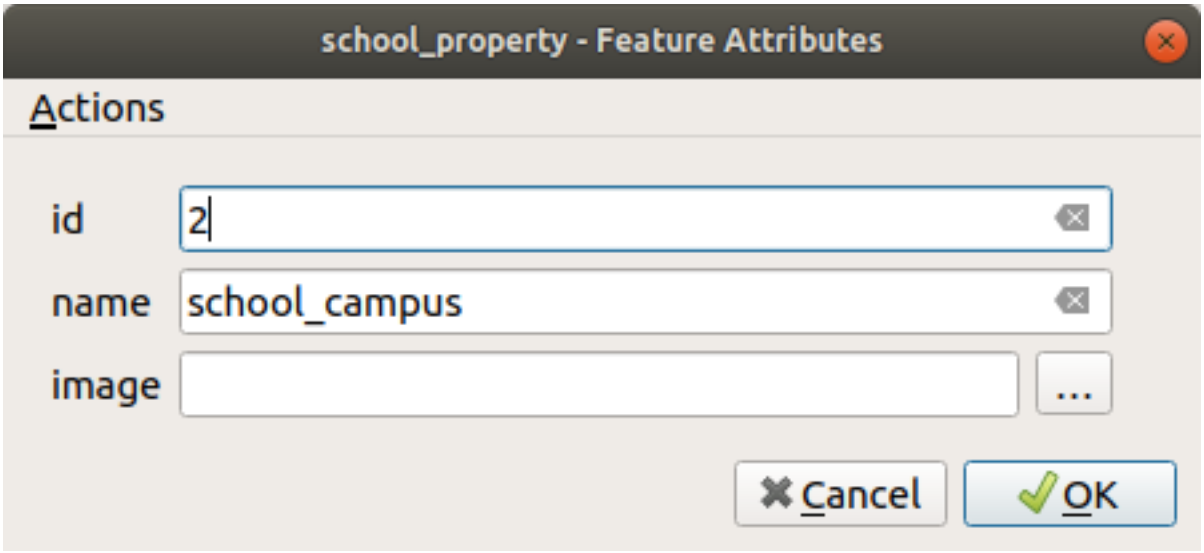


6. After the field has been created, move to the *Attributes Form* tab and select the `image` field.
7. Set *Widget Type* to *Attachment*:



8. Click *OK* in the *Layer Properties* dialog.
9. Use the *Identify* tool to click on one of the three features in the *school_property* layer.

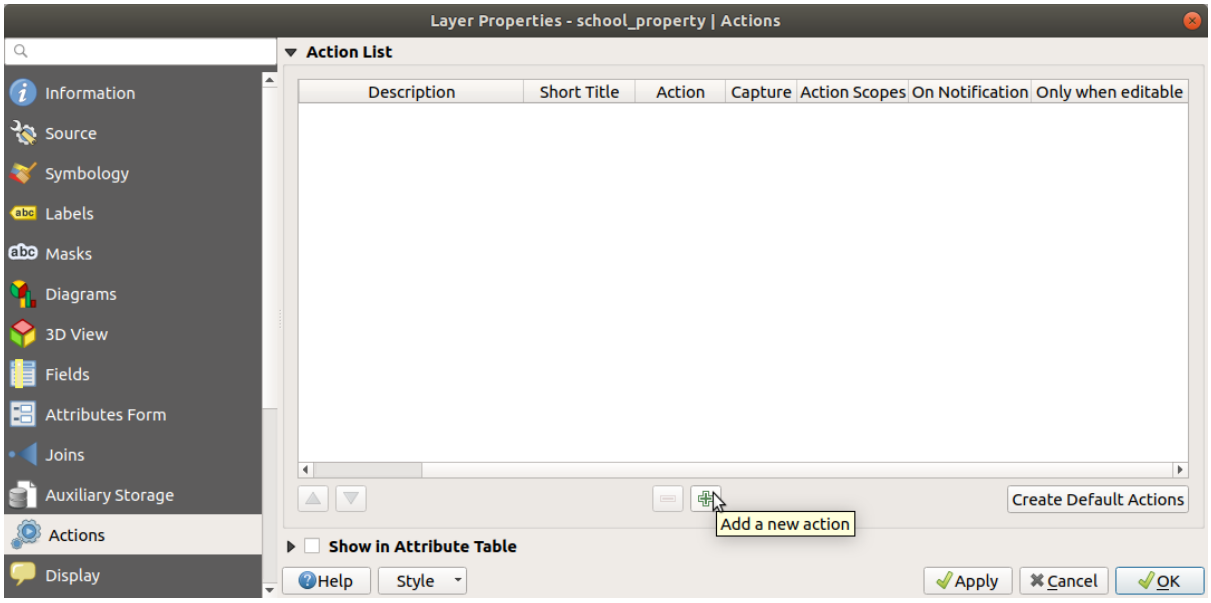
   Since you are still in edit mode, the dialog should be active and look like this:



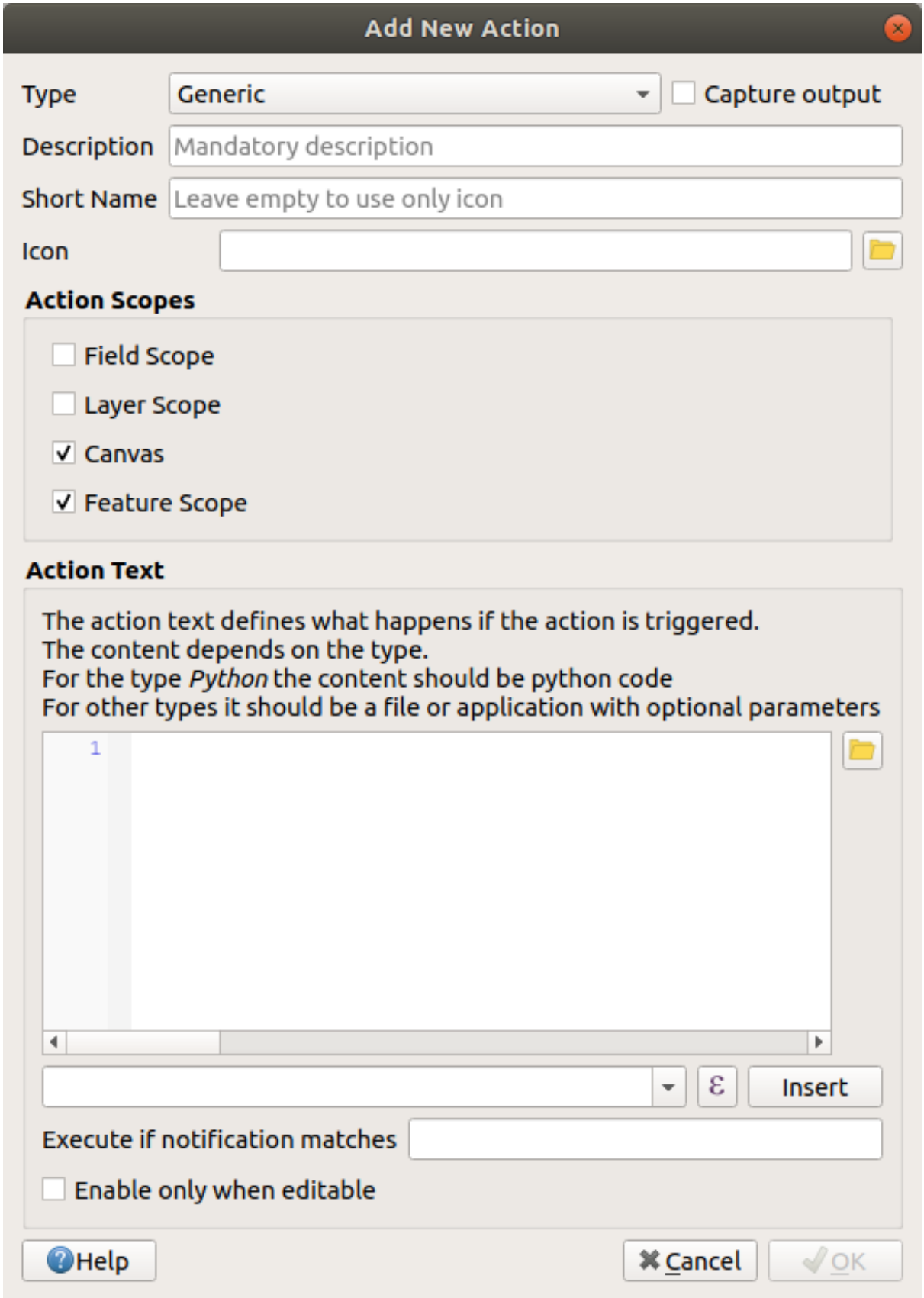
10. Click on the browse button (the … next to the *image* field).
11. Select the path for your image. The images are in `exercise_data/school_property_photos/` and are named the same as the features they should be associated with.
12. Click *OK*.
13. Associate all of the images with the correct features using this method.
14. Save your edits and exit edit mode.

5.4.2 Follow Along: Creating an Action

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



2. In the *Add New Action* dialog, enter the words `Show Image` into the *Description* field:



What to do next varies according to your operating system, so choose the appropriate course to follow:

- Windows

Click on the *Type* dropdown and choose *Open*.

- Ubuntu Linux

  Under *Action*, write `eog` for the *Gnome Image Viewer*, or write `display` to use *ImageMagick*. Remember to put a space after the command!

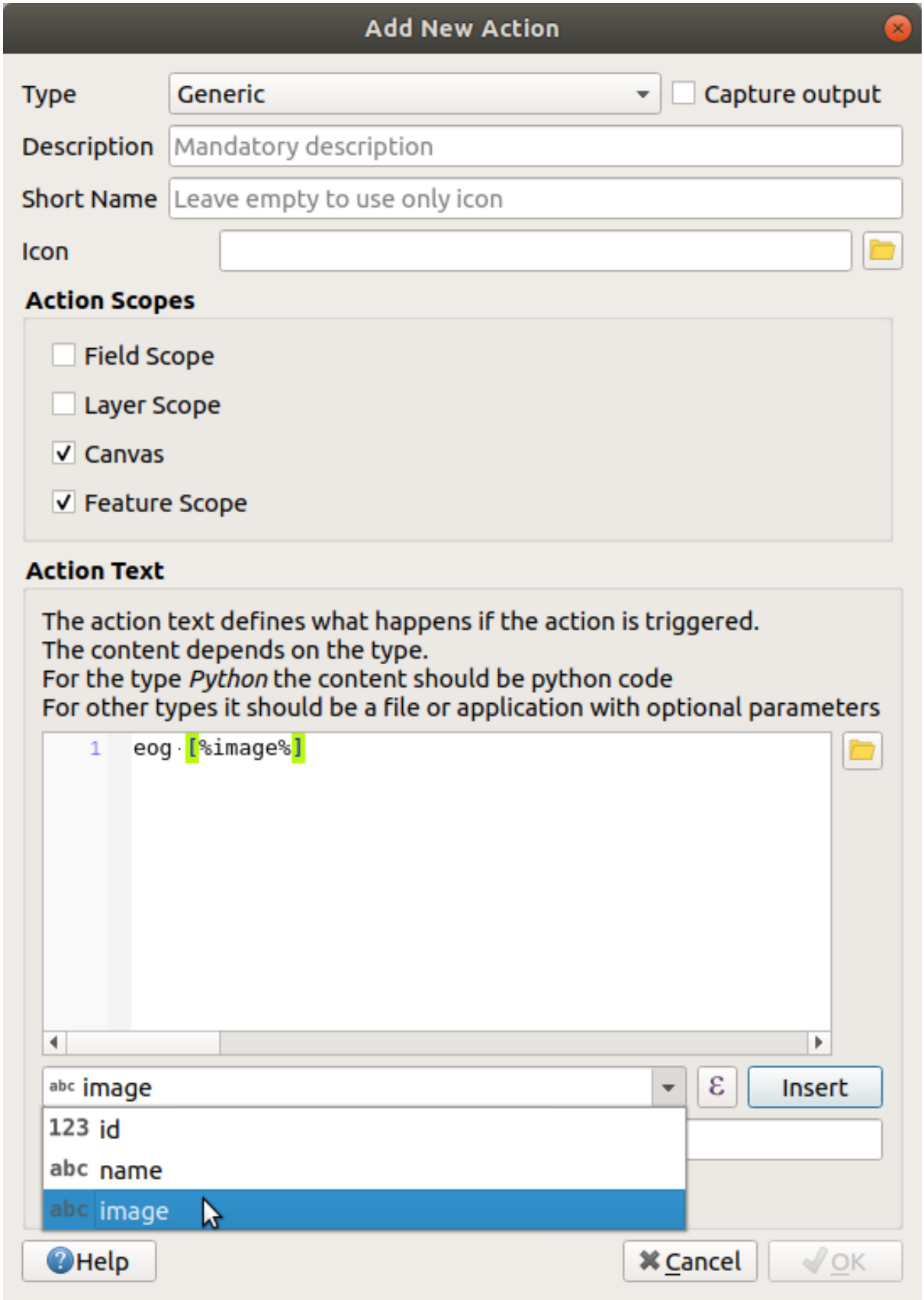
- macOS

  1. Click on the *Type* dropdown and choose *Mac*.
  2. Under *Action*, write `open`. Remember to put a space after the command!

Now you can continue writing the command.

You want to open the image, and QGIS knows where the image is. All it needs to do is to tell the *Action* where the image is.

3. Select *image* from the list:

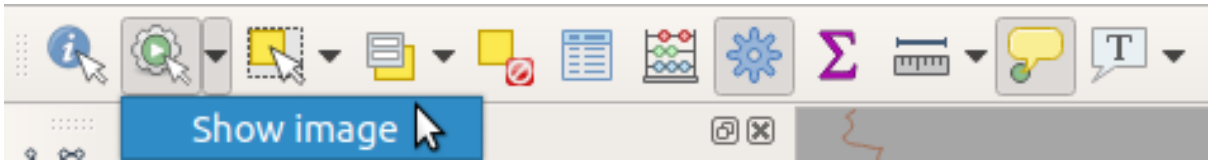


4. Click the *Insert field* button. QGIS will add the phrase `[% "image" %]` in the *Action Text* field.
5. Click the *OK* button to close the *Add New Action* dialog

6. Click *OK* to close the *Layer Properties* dialog

Now it is time to test the new action:

1. Click on the *school_property* layer in the *Layers* panel so that it is highlighted.
2. Find the Run feature action button (in the *Attributes Toolbar*).
3. Click on the down arrow to the right of this button. There is only one action defined for this layer so far, which is the one you just created.



4. Click the button itself to activate the tool.
5. Using this tool, click on any of the three school properties.

   The image for that property should open.

5.4.3 Follow Along: Searching the Internet

Let's say we are looking at the map and want to know more about the area that a farm is in. Suppose you know nothing of the area in question and want to find general information about it. Your first impulse, considering that you're using a computer right now, would probably be to Google the name of the area. So let's tell QGIS to do that automatically for us!

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

   We will be using the `name` field for each of our landuse areas to search Google.
2. Close the attribute table.
3. Go back to *Actions* in *Layer Properties*.
4. Click on the *Create Default Actions* button to add a number of pre-defined actions.
5. Remove all the actions but the *Open URL* action with the short name *Search Web* using the Remove the selected action button below.
6. Double-click on the remaining action to edit it
7. Change the *Description* to `Google Search`, and remove the content of the *Short Name* field.
8. Make sure that *Canvas* is among the checked *Action scopes*.

   What to do next varies according to your operating system, so choose the appropriate course to follow:

   - Windows

     Under *Type*, choose *Open*. This will tell Windows to open an Internet address in your default browser, such as Internet Explorer.
   - Ubuntu Linux

     Under *Action*, write `xdg-open`. This will tell Ubuntu to open an Internet address in your default browser, such as Chrome or Firefox.
   - macOS

     Under *Action*, write `open`. This will tell macOS to open an Internet address in your default browser, such as Safari.

Now you can continue writing the command

Whichever command you used above, you need to tell it which Internet address to open next. You want it to visit Google, and to search for a phrase automatically.

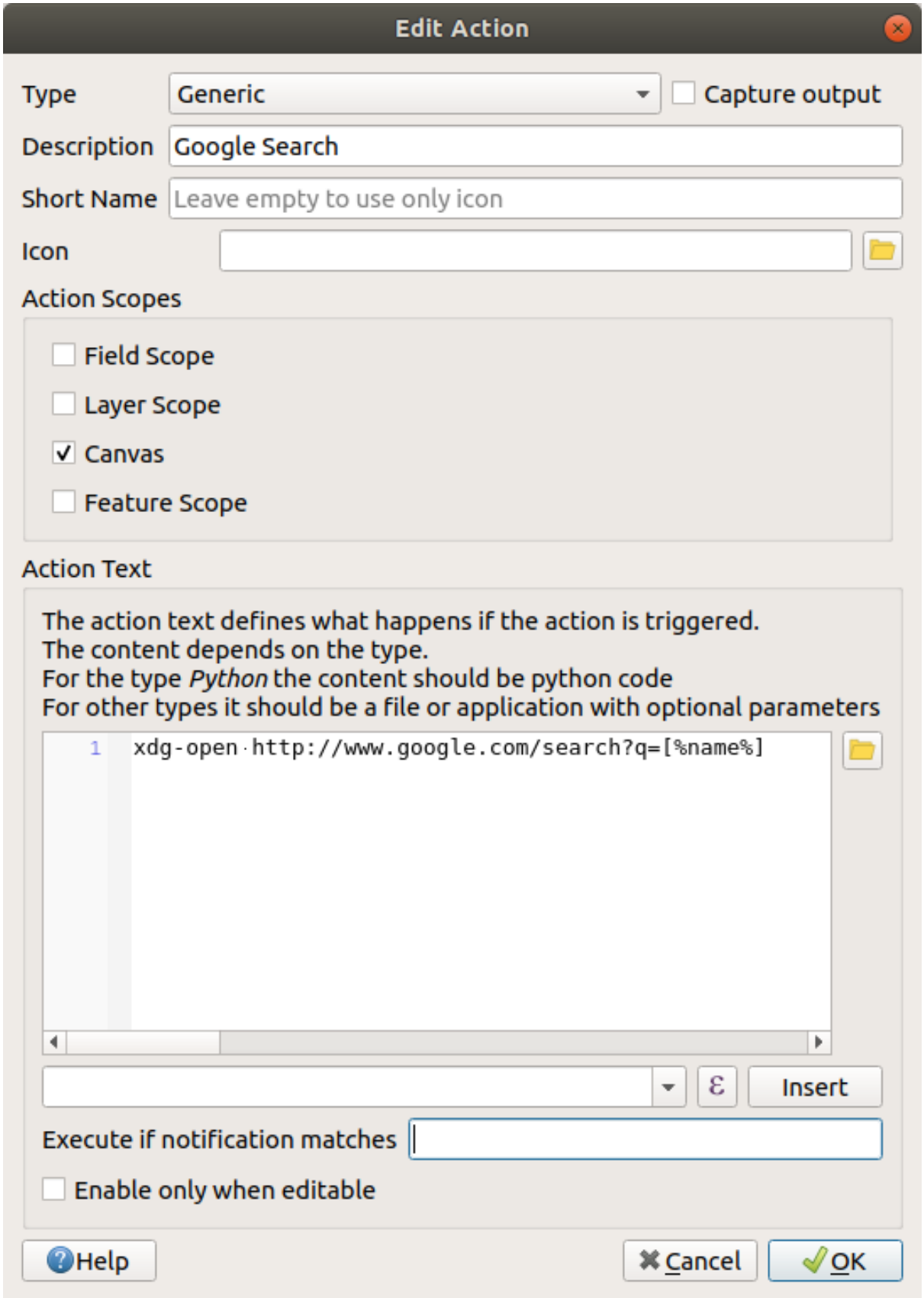
Usually when you use Google, you enter your search phrase into the Google Search bar. But in this case, you want your computer to do this for you. The way you tell Google to search for something (if you don't want to use its search bar directly) is by giving your Internet browser the address `https://www.google.com/search?q=SEARCH_PHRASE`, where `SEARCH_PHRASE`` is what you want to search for. Since we don't know what phrase to search for yet, we will just enter the first part (without the search phrase).

9. In the *Action* field, write `https://www.google.com/search?q=`. Remember to add a space after your initial command before writing this in!

   Now you want QGIS to tell the browser to tell Google to search for the value of `name` for any feature that you could click on.

10. Select the *name* field.

11. Click *Insert* button:



What this means is that QGIS is going to open the browser and send it to the address `https://www.google.com/search?q=[% "name" %]`. `[% "name" %]` tells QGIS to use the contents of the `name` field as the phrase to search for.

So if, for example, the landuse area you click on is named `Marloth Nature Reserve`, QGIS is going to send the browser to `https://www.google.com/search?q=Marloth%20Nature%20Reserve`, which will cause your browser to visit Google, which will in turn search for „Marloth Nature Reserve“.

12. If you have not done so already, set everything up as explained above.
13. Click the *OK* button to close the *Add New Action* dialog
14. Click *OK* to close the *Layer Properties* dialog

Now to test the new action.

1. With the *landuse* layer active in the *Layers* panel, click on the down arrow to the right of the Run feature action button, and select the only action (`Google Search`) defined for this layer.
2. Click on any landuse area you can see on the map. Your browser will now open, and will start a Google search for the place that is recorded as that area's `name` value.

Bemerkung: If your action doesn't work, check that everything was entered correctly; typos are common with this kind of work!

5.4.4 Follow Along: Open a Webpage Directly in QGIS

Above, you've seen how to open a webpage in an external browser. There are some shortcomings with this approach in that it adds an unknowable dependency – will the end-user have the software required to execute the action on their system? As you've seen, they don't necessarily even have the same kind of base command for the same kind of action, if you don't know which OS they will be using. With some OS versions, the above commands to open the browser might not work at all. This could be an insurmountable problem.

However, QGIS sits on top of the incredibly powerful and versatile Qt library. Also, QGIS actions can be arbitrary, tokenized (i.e. using variable information based on the contents of a field attribute) Python commands!

Now you will see how to use a python action to show a web page. It is the same general idea as opening a site in an external browser, but it requires no browser on the user's system since it uses the Qt QWebView class (which is a webkit based html widget) to display the content in a pop-up window.

Let us use Wikipedia this time. So the URL you request will look like this:

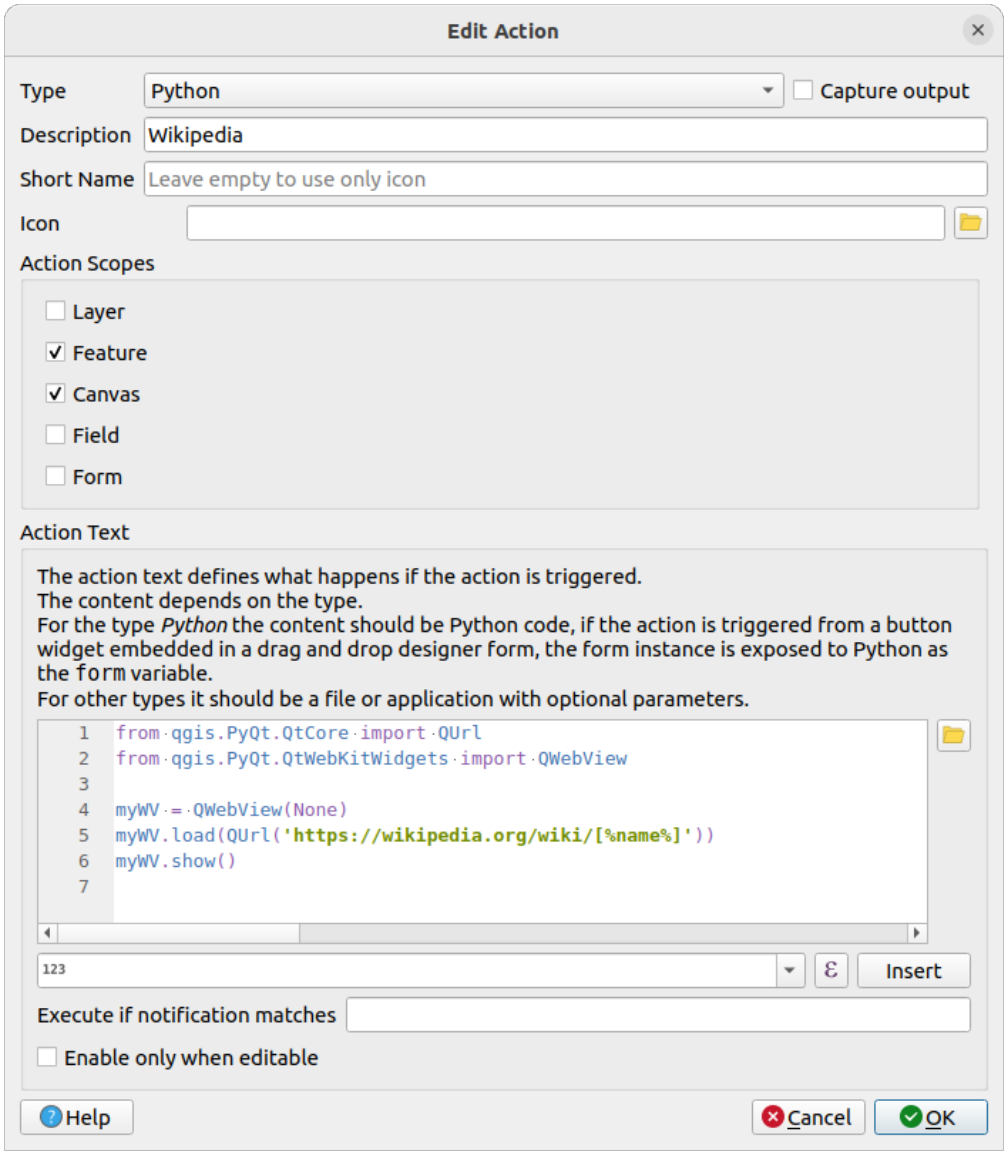
`https://wikipedia.org/wiki/SEARCH_PHRASE`

To create the layer action:

1. Open the *Layer Properties* dialog and head over to the *Actions* tab.
2. Set up a new action using the following properties for the action:
   - *Type*: `Python`
   - *Description*: `Wikipedia`
   - *Scope*: `Feature`, `Canvas`
   - *Action Text*:

```
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()
```



There are a couple of things going on here:

- `[%name%]` will be replaced by the actual attribute value when the action is invoked (as before).
- The code simply creates a new `QWebView` instance, sets its URL, and then calls `show()` on it to make it visible as a window on the user's desktop.

You could also use this approach to display an image without requiring that the users have a particular image viewer on their system.

3. Try to use the methods described above to load a Wikipedia page using the Wikipedia action you just created.

5.4.5 In Conclusion

Actions allow you to give your map extra functionality, useful to the end-user who views the same map in QGIS. Due to the fact that you can use shell commands for any operating system, as well as Python, the sky is the limit in terms of the functions you could incorporate!

5.4.6 What's Next?

Now that you've done all kinds of vector data creation, you will learn how to analyze the data to solve problems. That is the topic of the next module.

KAPITEL 6

Module: Vektoranalyse

Nun, da Sie einige Funktionen bearbeitet haben, sollten Sie wissen, was man sonst mit ihnen machen kann. Kenndaten mit Attributen zu besitzen, ist nett, aber am Ende sagt das nichts anderes aus als eine normale Nicht-GIS-Karte.

Der Hauptvorteil eines GIS ist: * Ein GIS kann Fragen beantworten. *

Für die nächsten drei Module werden wir uns bemühen, eine * Forschungsfrage * mit der Hilfe von GIS-Funktionen zu beantworten. Zum Beispiel sind Sie ein Makler und Sie sind auf der Suche nach einer Wohnimmobilie in | majorUrbanName | Für Kunden mit folgenden Kriterien:

1. Sie muss sich in Swellendam befinden.
2. Sie muss innerhalb einer vernünftigen Fahrstrecke einer Schule (ca. 1km) liegen.
3. Sie muss größer als 100m^2 sein.
4. Näher als 50m an einer Hauptstraße.
5. Näher als 500m an einem Restaurant.

Innerhalb der nächsten Module werden wir die Leistung von GIS-Analyse-Tools nutzen, um geeignete Flächeneigenschaften für diese neue Wohnsiedlung zu lokalisieren.

6.1 Lesson: Reprojecting and Transforming Data

Let us talk about Coordinate Reference Systems (CRSs) again. We have touched on this briefly before, but haven't discussed what it means practically.

The goal for this lesson: To reproject and transform vector datasets.

6.1.1 Follow Along: Projections

The CRS that all the data, as well as the map itself are in right now is called *WGS84*. This is a very common Geographic Coordinate System (GCS) for representing data. But there's a problem, as we will see.

1. Save your current map
2. Then open the map of the world which you will find under `exercise_data/world/world.qgs`
3. Zoom in to South Africa by using the *Zoom In* tool
4. Try setting a scale in the *Scale* field, which is in the *Statusbar* along the bottom of the screen. While over South Africa, set this value to `1:5 000 000` (one to five million).
5. Pan around the map while keeping an eye on the *Scale* field

Notice the scale changing? That's because you are moving away from the one point that you zoomed into at `1:5 000 000`, which was at the center of your screen. All around that point, the scale is different.

To understand why, think about a globe of the Earth. It has lines running along it from North to South. These longitude lines are far apart at the equator, but they meet at the poles.

In a GCS, you are working on this sphere, but your screen is flat. When you try to represent the sphere on a flat surface, distortion occurs, similar to what would happen if you cut open a tennis ball and tried to flatten it out. What this means on a map is that the longitude lines stay equally far apart from each other, even at the poles (where they are supposed to meet). This means that, as you travel away from the equator on your map, the scale of the objects that you see gets larger and larger. What this means for us, practically, is that there is no constant scale on our map!

To solve this, let's use a Projected Coordinate System (PCS) instead. A PCS „projects" or converts the data in a way that makes allowance for the scale change and corrects it. Therefore, to keep the scale constant, we should reproject our data to use a PCS.

6.1.2 Follow Along: „On the Fly" Reprojection

By default, QGIS reprojects data „on the fly". What this means is that even if the data itself is in another CRS, QGIS can project it as if it were in a CRS of your choice.

You can change the CRS of the project by clicking on the Current projection button in the bottom right corner of QGIS.

1. In the dialog that appears, type the word `global` into the *Filter* field. A few CRSs should appear in the *Predefined Reference Systems* field below.
2. Select *WGS 84 / NSIDC EASE-Grid 2.0 Global | EPSG:6933* entry by clicking on it, and then click *OK*.

   Notice how the shape of South Africa changes. All projections work by changing the apparent shapes of objects on Earth.
3. Zoom to a scale of `1:5 000 000` again, as before.
4. Pan around the map.

   Notice how the scale stays the same!

„On the fly" reprojection is also used for combining datasets that are in different CRSs.

1. Add another vector layer to your map which has the data for South Africa only. You will find it as `exercise_data/world/RSA.shp`.
2. Load it. A quick way to see its CRS is by hovering the mouse over the layer in the legend. It is `EPSG:3410`.

   What do you notice?

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

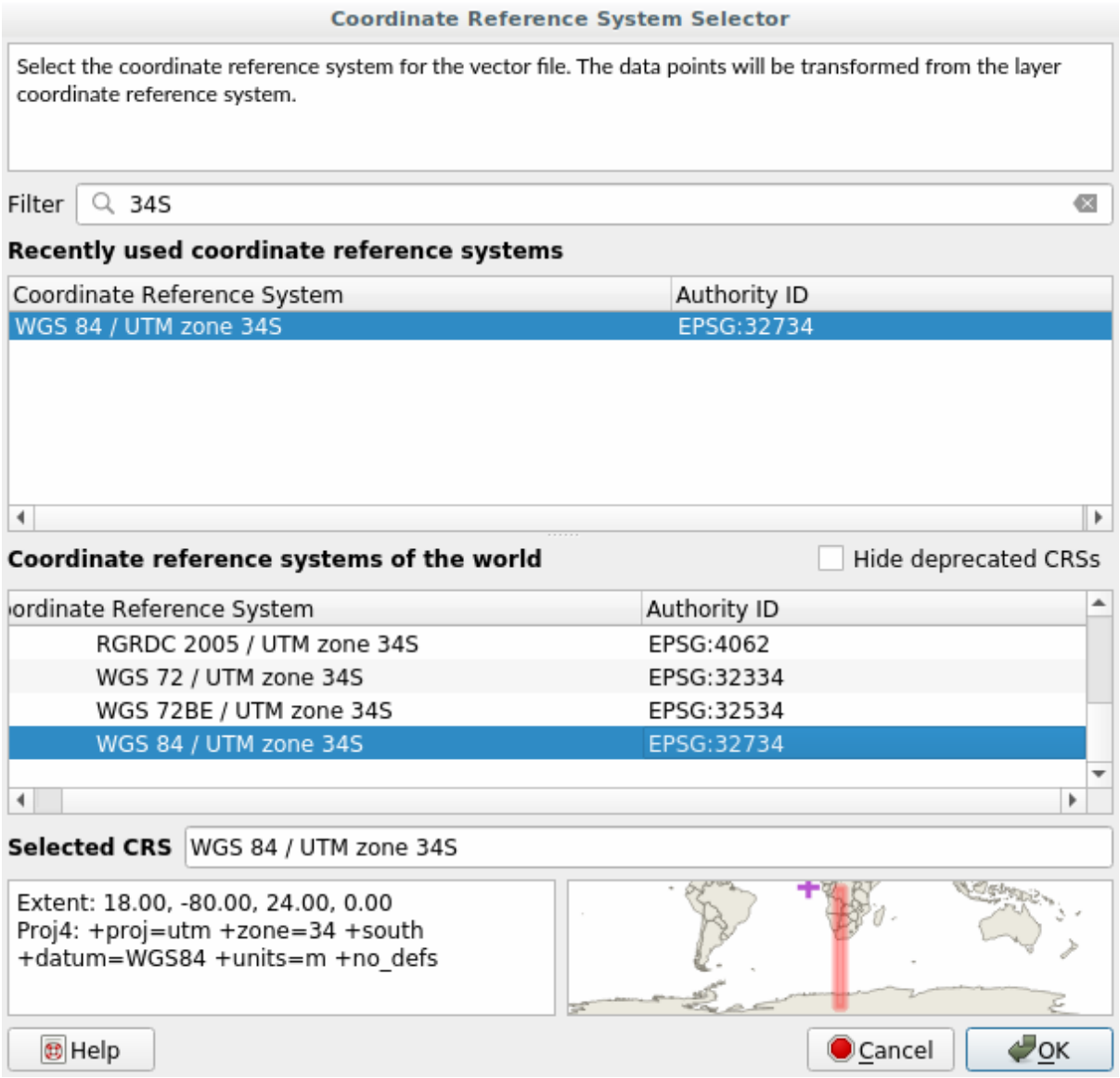
6.1.3 Follow Along: Saving a Dataset to Another CRS

Sometimes you need to export an existing dataset with another CRS. As we will see in the next lesson, if you need to make distance calculations on layer, it is always better to have the layer in a projected coordinate system.

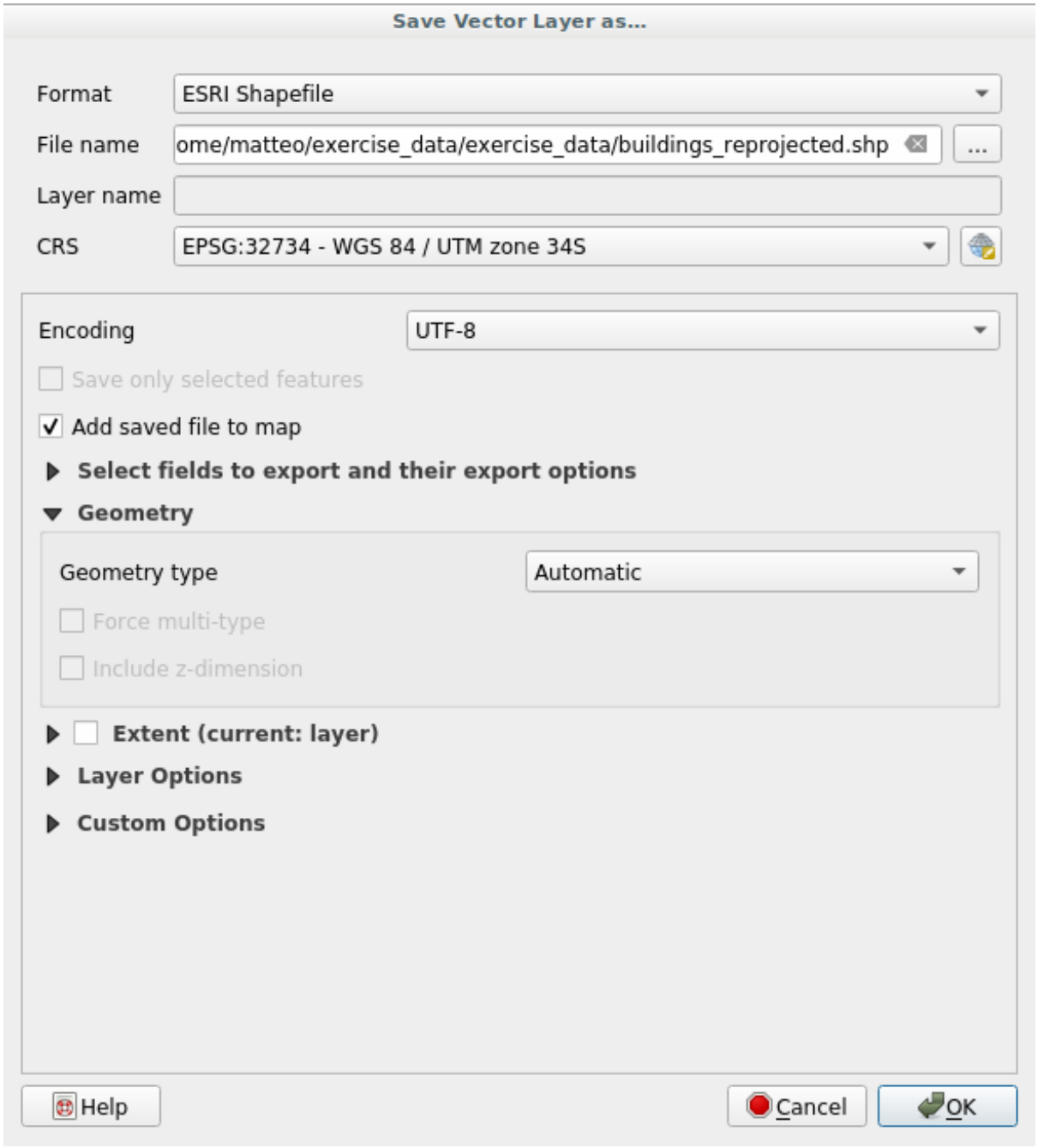
Be aware that the ‚on the fly' reprojection is related to the **project** and not to single layers. This means that a layer can have a different CRS from the project even if you see it in the *correct* position.

You can easily export the layer with another CRS.

1. Add the `buildings` dataset from `training_data.gpkg`
2. Right-click on the `buildings` layer in the *Layers* panel
3. Select *Export* ► *Save Features As…* in the menu that appears. You will be shown the *Save Vector Layer as…* dialog.
4. Click on the *Browse* button next to the *File name* field
5. Navigate to `exercise_data/` and specify the name of the new layer as `buildings_reprojected.shp`.
6. Change the value of the *CRS*. Only the recent CRSs used will be shown in the drop-down menu. Click on the Select projection button next to the drop-down menu.
7. The *Coordinate Reference System Selector* dialog will appear. In its *Filter* field, search for `34S`.
8. Select *WGS 84 / UTM zone 34S | EPSG:32734* from the list



9. Leave the other options unchanged. The *Save Vector Layer as…* dialog now looks like this:



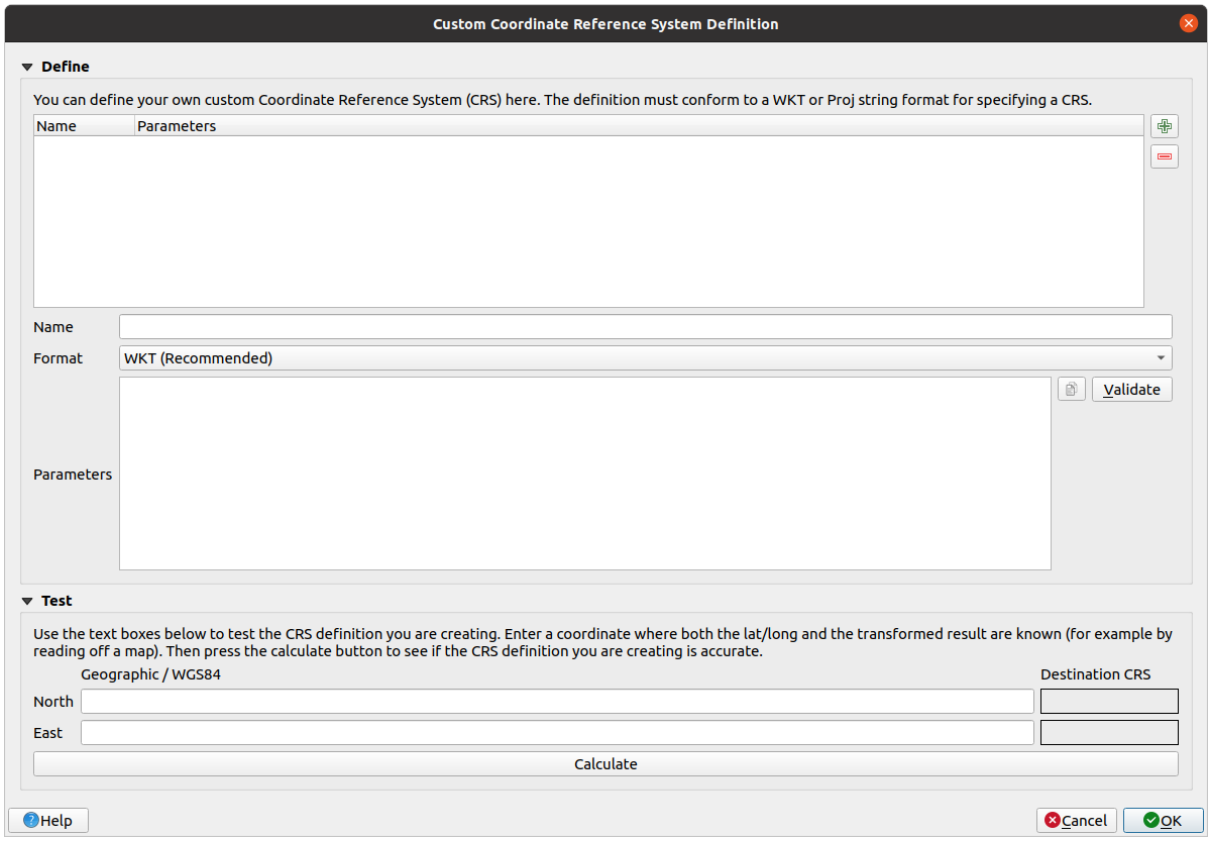
10. Klicken Sie auf *OK*

You can now compare the old and new projections of the layer and see that they are in two different CRS but they are still overlapping.

6.1.4 Follow Along: Creating Your Own Projection

There are many more projections than just those included in QGIS by default. You can also create your own projections.

1. Start a new map
2. Load the `world/oceans.shp` dataset
3. Go to *Settings* ► *Custom Projections…* and you will see this dialog.



4. Click on the Add new CRS button to create a new projection
5. An interesting projection to use is called `Van der Grinten I`. Enter its name in the *Name* field.

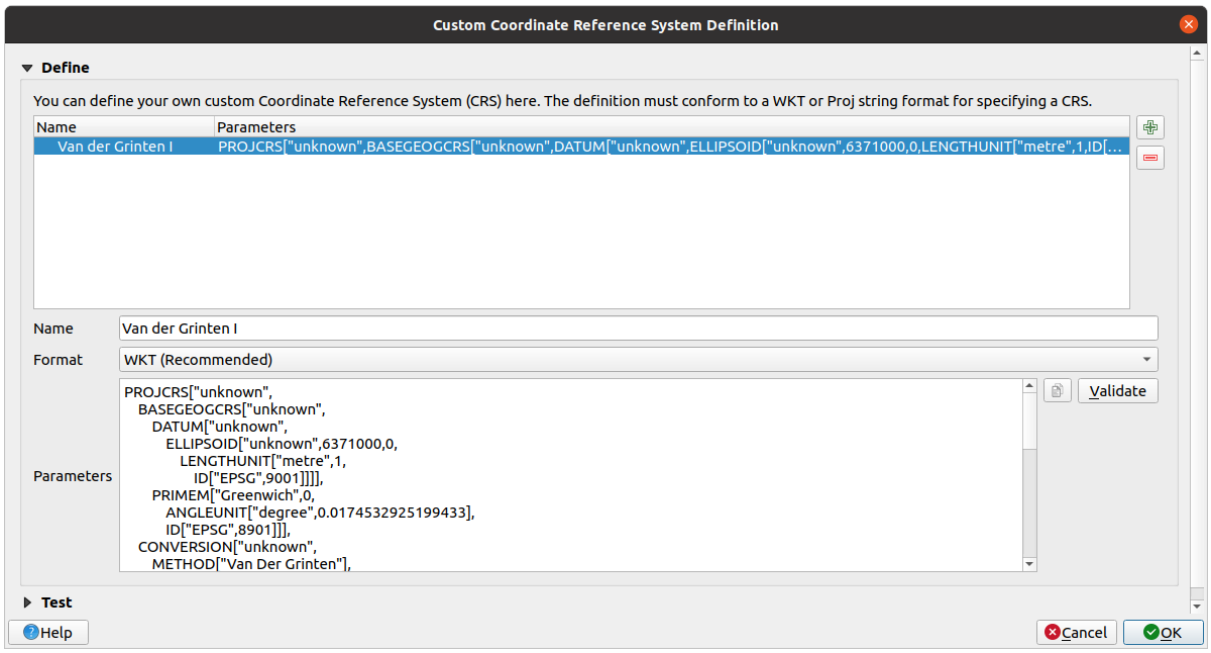
   This projection represents the Earth on a circular field instead of a rectangular one, as most other projections do.
6. In *Format*, select *WKT (Recommended)*
7. Add the following string in the *Parameters* field:

```
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]]],
    CONVERSION["unknown",
        METHOD["Van Der Grinten"],
```

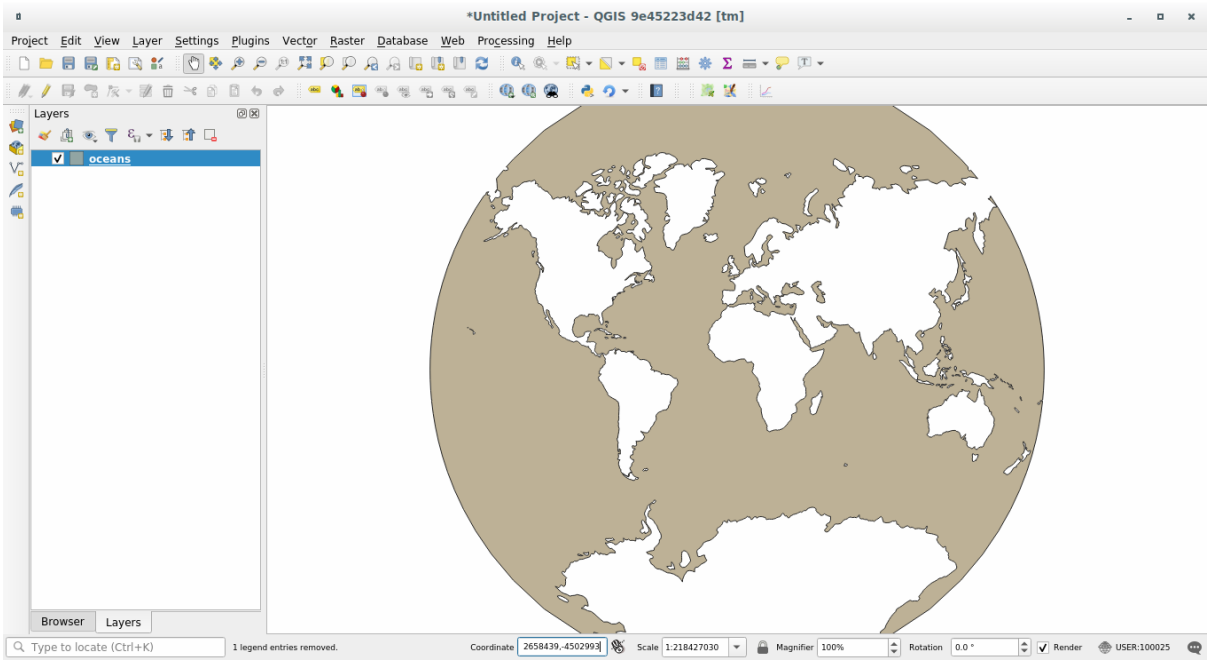
(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        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. Klicken Sie auf *OK*
9. Click on the Current CRS button at the right of the status bar to change the project CRS
10. Choose your newly defined projection (search for its name in the *Filter* field)
11. On applying this projection, the map will be reprojected thus:



6.1.5 In Conclusion

Different projections are useful for different purposes. By choosing the correct projection, you can ensure that the features on your map are being represented accurately.

6.1.6 Further Reading

Materials for the *Advanced* section of this lesson were taken from this article.

Read further information on Coordinate Reference Systems.

6.1.7 What's Next?

In the next lesson you will learn how to analyze vector data using QGIS' various vector analysis tools.

6.2 Lesson: Vector Analysis

Vector data can also be analyzed to reveal how different features interact with each other in space. There are many different analysis-related functions, so we won't go through them all. Rather, we will pose a question and try to solve it using the tools that QGIS provides.

The goal for this lesson: To ask a question and solve it using analysis tools.

6.2.1 The GIS Process

Before we start, it would be useful to give a brief overview of a process that can be used to solve a problem. The way to go about it is:

1. State the Problem
2. Get the Data
3. Analyze the Problem
4. Present the Results

6.2.2 The Problem

Let's start off the process by deciding on a problem to solve. For example, you are an estate agent and you are looking for a residential property in Swellendam for clients who have the following criteria:

1. It needs to be in Swellendam
2. It must be within reasonable driving distance of a school (say 1km)
3. It must be more than 100m squared in size
4. Closer than 50m to a main road
5. Closer than 500m to a restaurant

6.2.3 The Data

To answer these questions, we are going to need the following data:

1. The residential properties (buildings) in the area
2. The roads in and around the town
3. The location of schools and restaurants
4. The size of buildings

These data are available through OSM, and you should find that the dataset you have been using throughout this manual also can be used for this lesson.

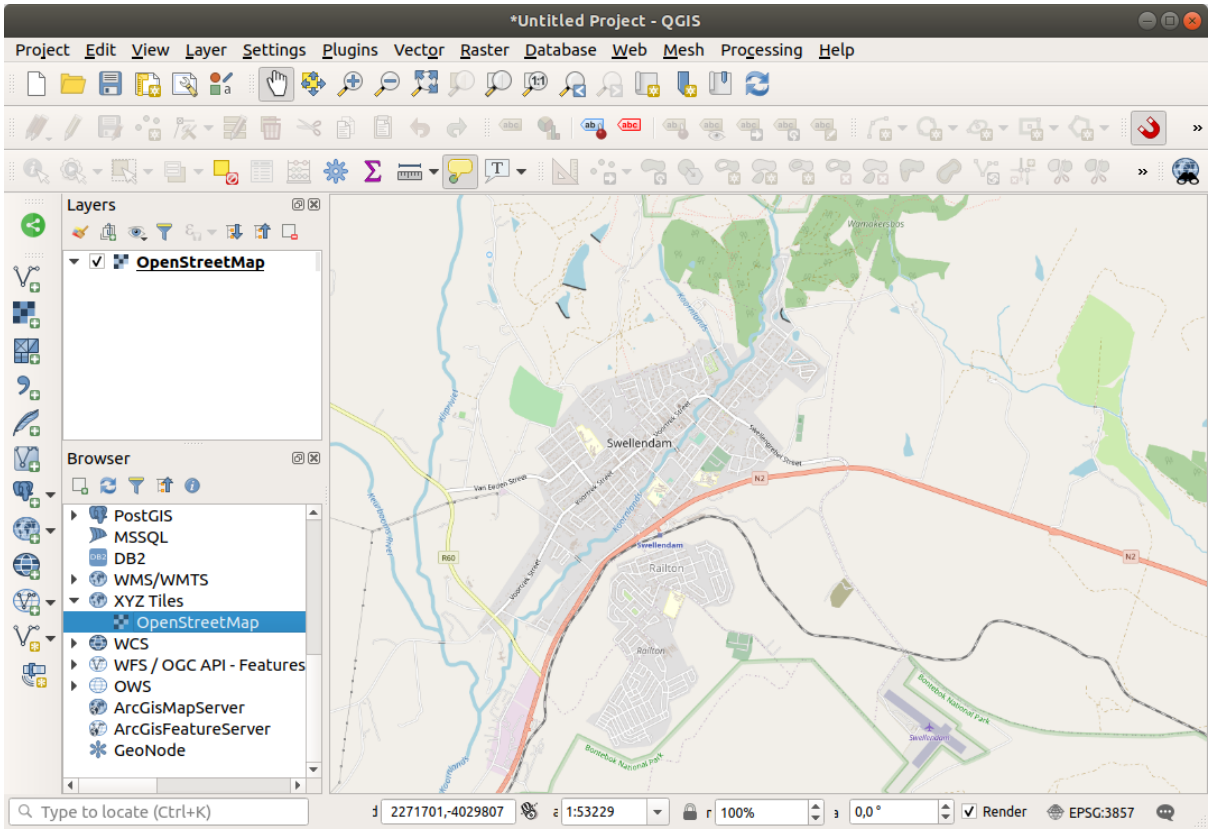
If you want to download data from another area, jump to the *Introduction Chapter* to read how to do it.

Bemerkung: Although OSM downloads have consistent data fields, the coverage and detail does vary. If you find that your chosen region does not contain information on restaurants, for example, you may need to chose a different region.

6.2.4 Follow Along: Start a Project and get the Data

We first need to load the data to work with.

1. Start a new QGIS project
2. If you want, you can add a background map. Open the *Browser* and load the *OSM* background map from the *XYZ Tiles* menu.



3. In the `training_data.gpkg` Geopackage database, you will find most the datasets we will use in this chapter:
    1. `buildings`
    2. `roads`
    3. `restaurants`
    4. `schools`

    Load them, and also `landuse.sqlite`.
4. Zoom to the layer extent to see Swellendam, South Africa

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

    Some roads in OSM datasets are listed as `unclassified`, `tracks`, `path` and `footway`. We want to exclude these from our dataset and focus on the other road types, more suitable for this exercise.

    Moreover, OSM data might not be updated everywhere, and we will also exclude `NULL` values.
5. Right click on the `roads` layer and choose *Filter…*.
6. In the dialog that pops up we filter these features with the following expression:

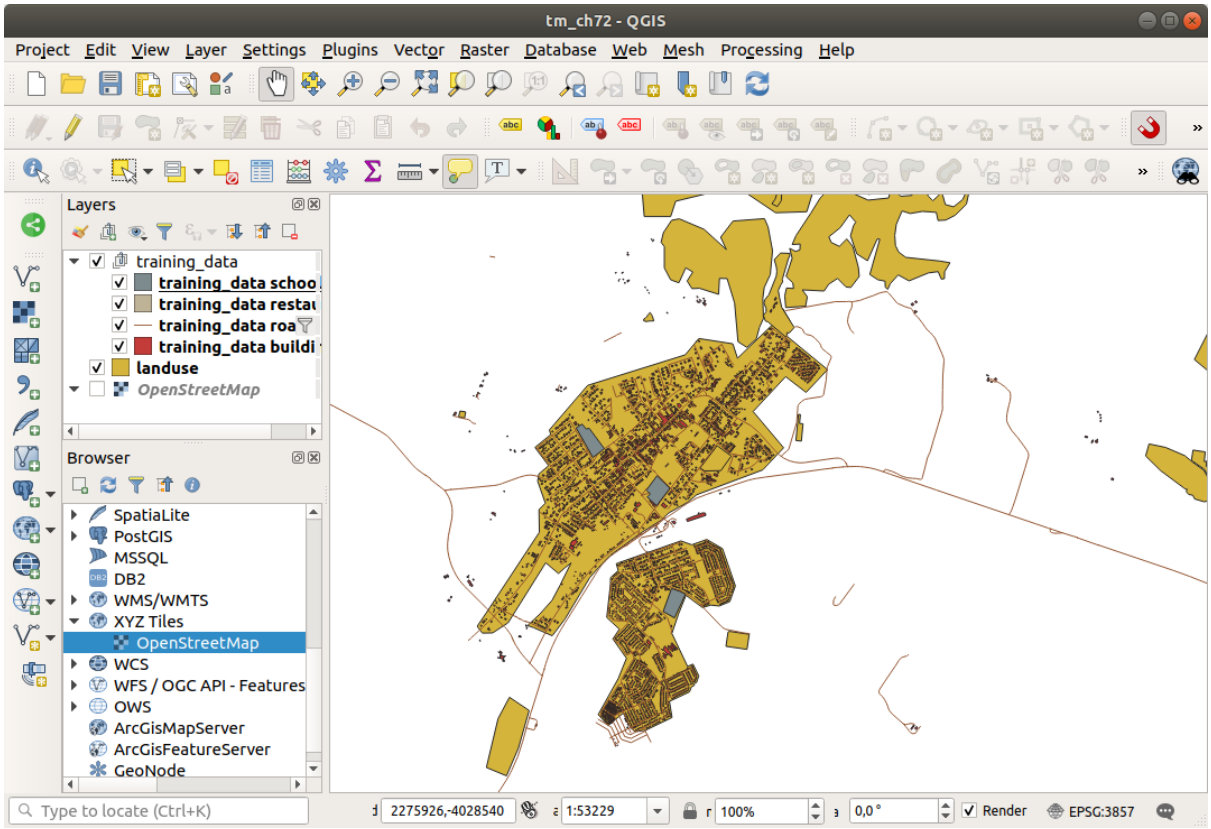
```
"highway" NOT IN ('footway', 'path', 'unclassified', 'track') AND "highway" IS␣
↪NOT NULL
```

The concatenation of the two operators `NOT` and `IN` excludes all the features that have these attribute values in the `highway` field.

`IS NOT NULL` combined with the `AND` operator excludes roads with no value in the `highway` field.

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

The map with all the data should look like the following one:



6.2.5 Try Yourself Convert Layers' CRS

Because we are going to be measuring distances within our layers, we need to change the layers' CRS. To do this, we need to select each layer in turn, save the layer to a new one with our new projection, then import that new layer into our map.

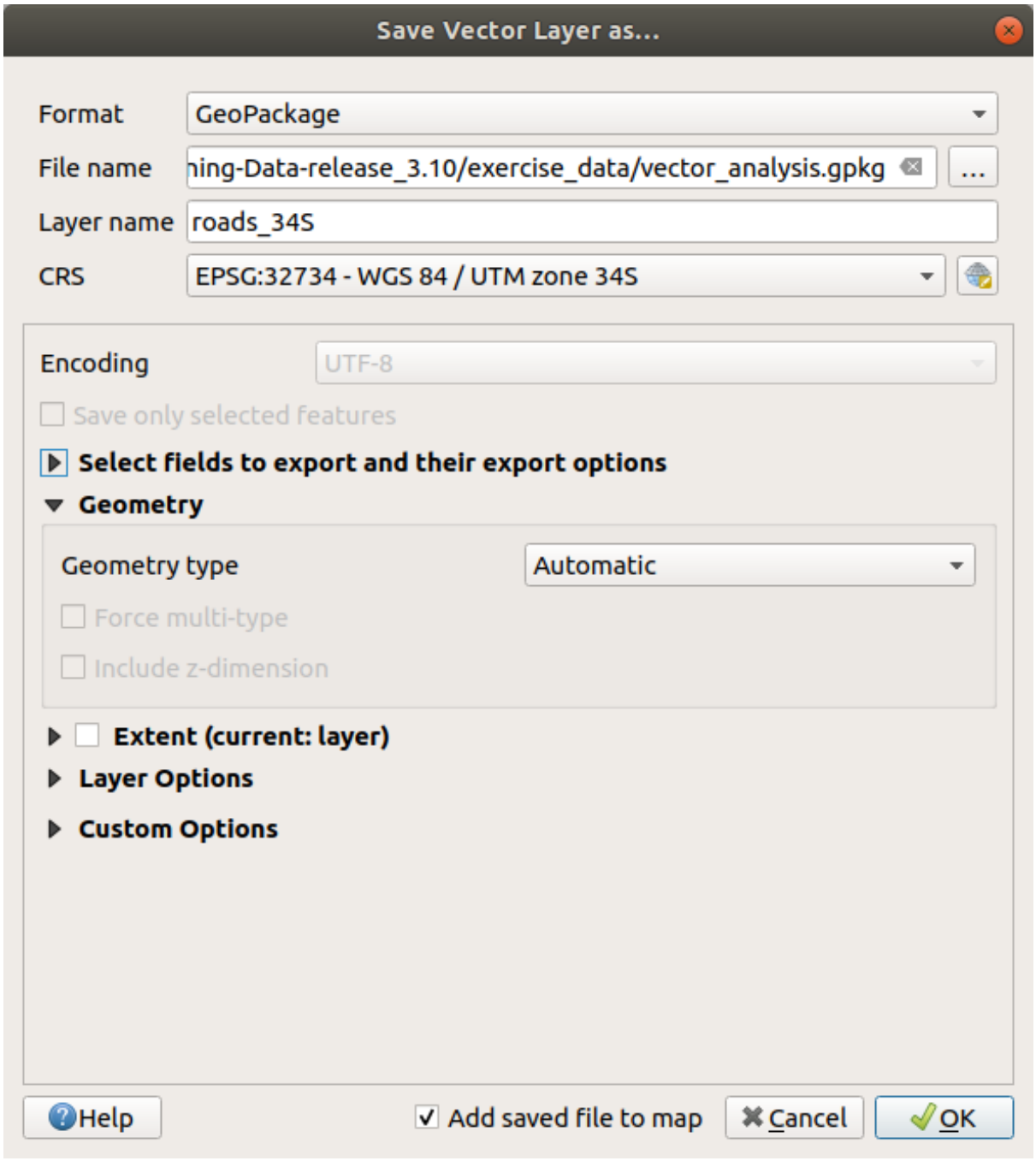
You have many different options, e.g. you can export each layer as an ESRI Shapefile format dataset, you can append the layers to an existing GeoPackage file, or you can create another GeoPackage file and fill it with the new reprojected layers. We will show the last option, so the `training_data.gpkg` will remain clean. Feel free to choose the best workflow for yourself.

Bemerkung: In this example, we are using the *WGS 84 / UTM zone 34S* CRS, but you should use a UTM CRS which is more appropriate for your region.

---

1. Right click the *roads* layer in the *Layers* panel
2. Click *Export –> Save Features As…*

3. In the *Save Vector Layer As* dialog choose *GeoPackage* as *Format*
4. Click on … for the *File name*, and name the new GeoPackage `vector_analysis`
5. Change the *Layer name* to `roads_34S`
6. Change the *CRS* to *WGS 84 / UTM zone 34S*
7. Click on *OK*:



This will create the new GeoPackage database and add the `roads_34S` layer.

8. Repeat this process for each layer, creating a new layer in the `vector_analysis.gpkg` GeoPackage file with `_34S` appended to the original name.

   On macOS, press the *Replace* button in the dialog that pops up to allow QGIS to overwrite the existing GeoPackage.

Bemerkung: When you choose to save a layer to an existing GeoPackage, QGIS will **add** that layer next to the existing layers in the GeoPackage, if no layer of the same name already exists.

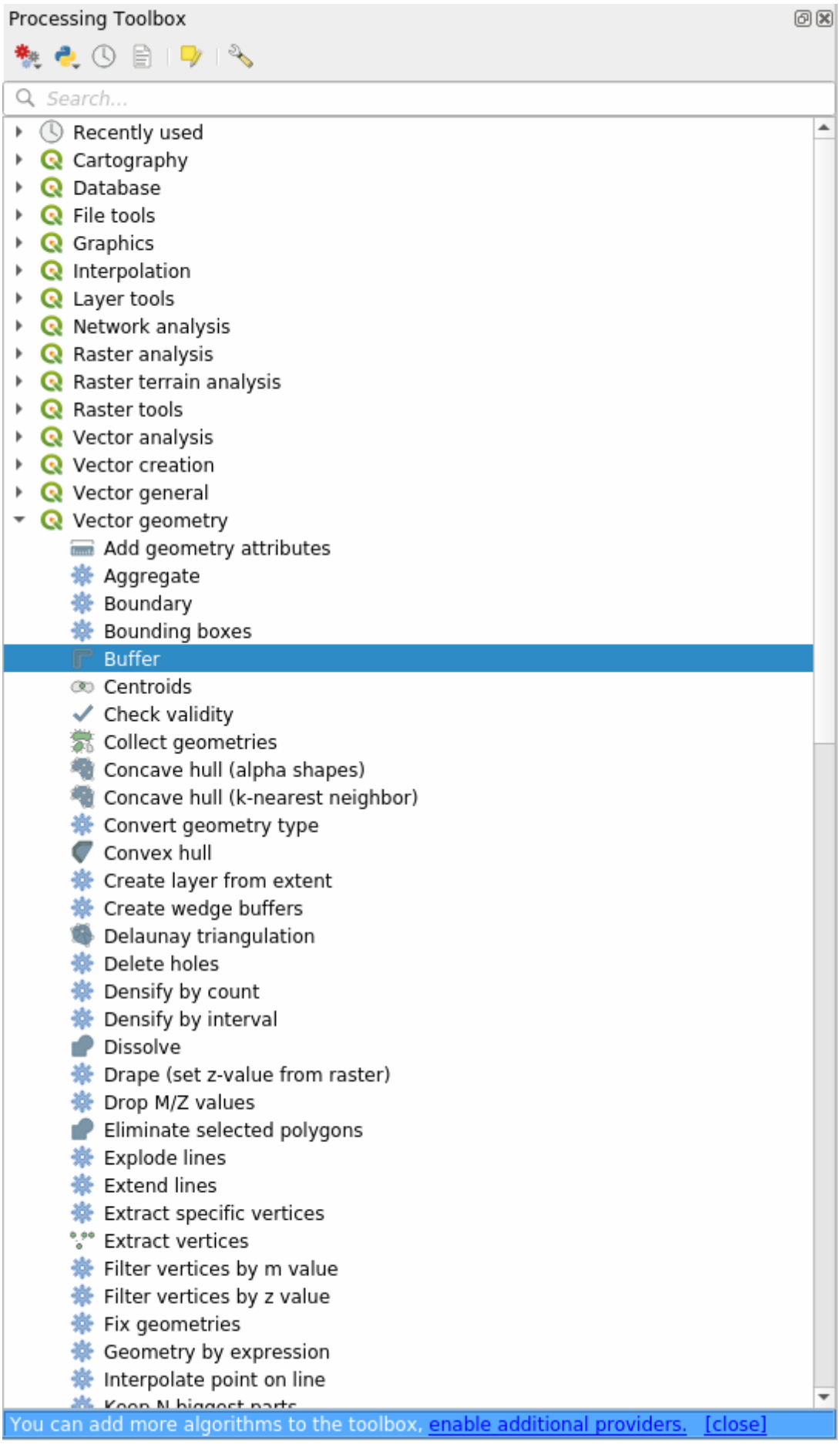
9. Remove each of the old layers from the project
10. Once you have completed the process for all the layers, right click on any layer and click *Zoom to layer extent* to focus the map to the area of interest.

Now that we have converted OSM data to a UTM projection, we can begin our calculations.

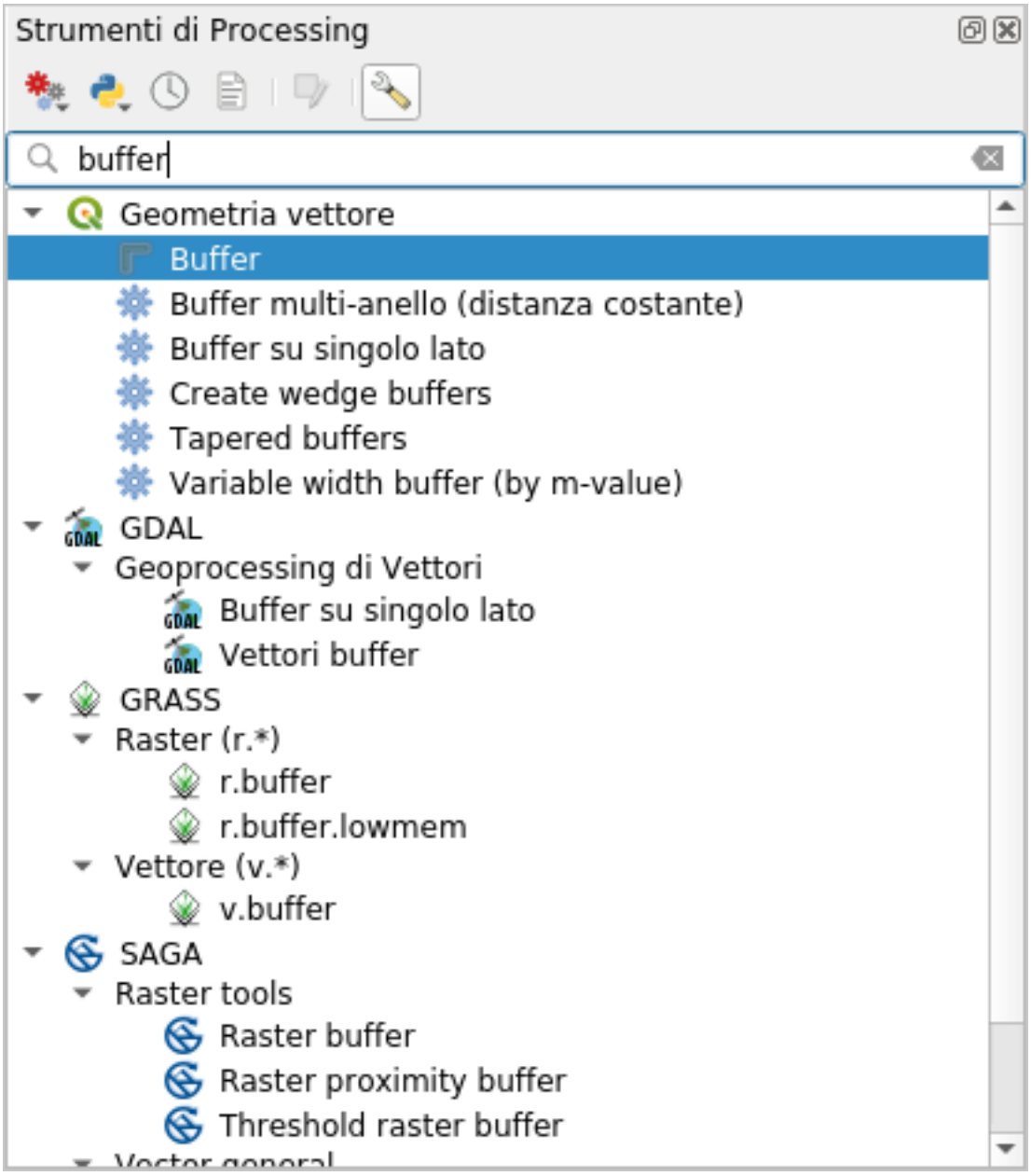
6.2.6 Follow Along: Analyzing the Problem: Distances From Schools and Roads

QGIS allows you to calculate distances between any vector object.

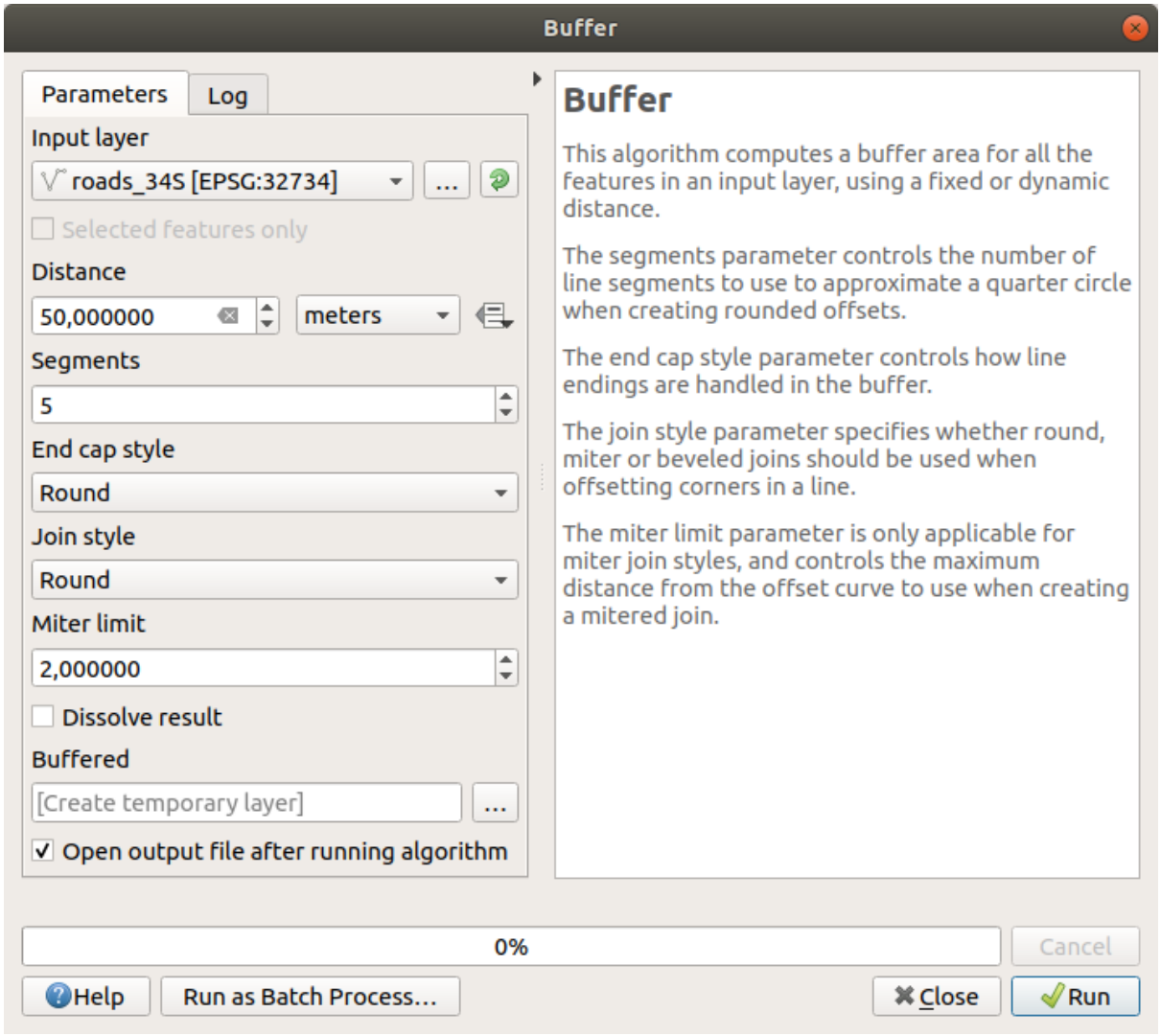
1. Make sure that only the `roads_34S` and `buildings_34S` layers are visible (to simplify the map while you're working)
2. Click on the *Processing* ► *Toolbox* to open the analytical *core* of QGIS. Basically, **all** algorithms (for vector **and** raster analysis) are available in this toolbox.
3. We start by calculating the area around the `roads_34S` by using the *Buffer* algorithm. You can find it in the *Vector Geometry* group.


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]

Or you can type `buffer` in the search menu in the upper part of the toolbox:



4. Double click on it to open the algorithm dialog
5. Select `roads_34S` as *Input layer*, set *Distance* to 50 and use the default values for the rest of the parameters.



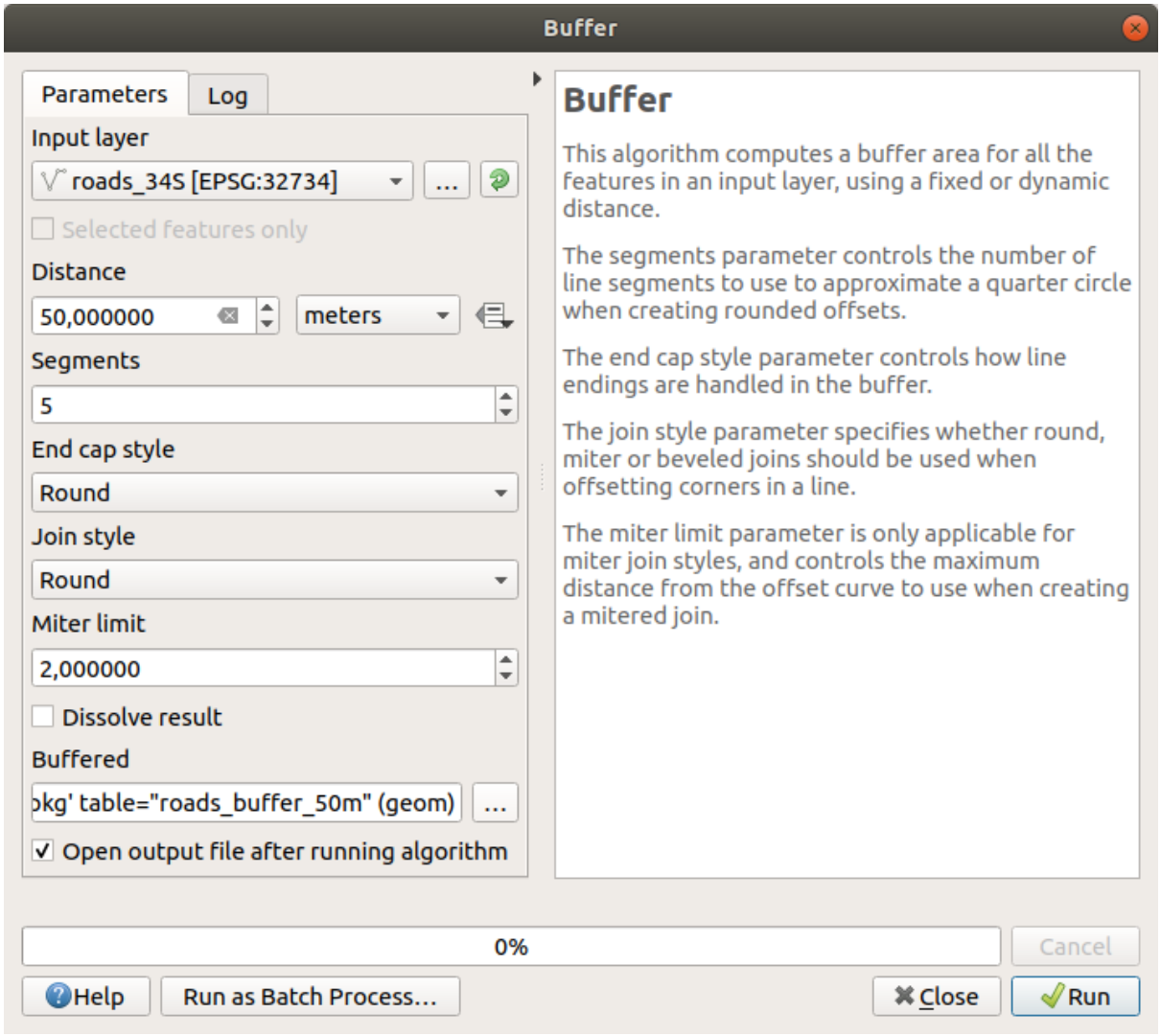
6. The default *Distance* is in meters because our input dataset is in a Projected Coordinate System that uses meter as its basic measurement unit. You can use the combo box to choose other projected units like kilometers, yards, etc.

   Bemerkung: If you are trying to make a buffer on a layer with a Geographical Coordinate System, Processing will warn you and suggest to reproject the layer to a metric Coordinate System.

   ---

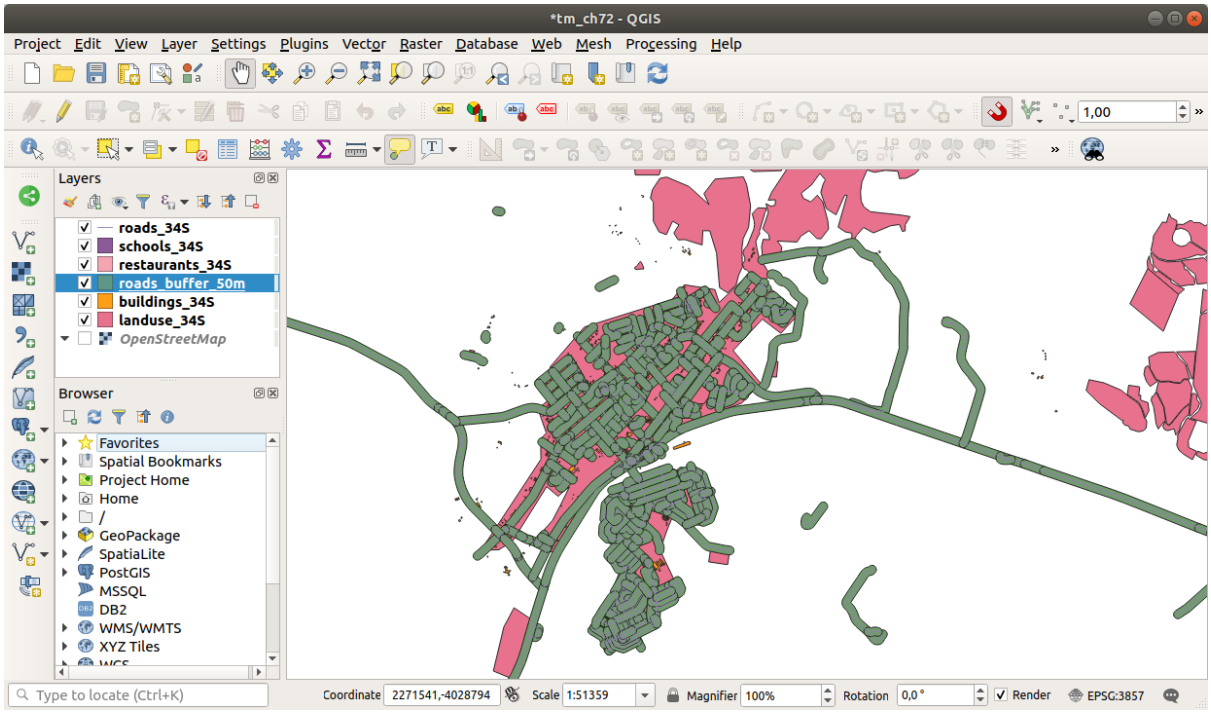
7. By default, *Processing* creates temporary layers and adds them to the *Layers* panel. You can also append the result to the GeoPackage database by:

   1. Clicking on the … button and choose *Save to GeoPackage…*
   2. Naming the new layer `roads_buffer_50m`
   3. Saving it in the `vector_analysis.gpkg` file



8. Click on *Run*, and then close the *Buffer* dialog

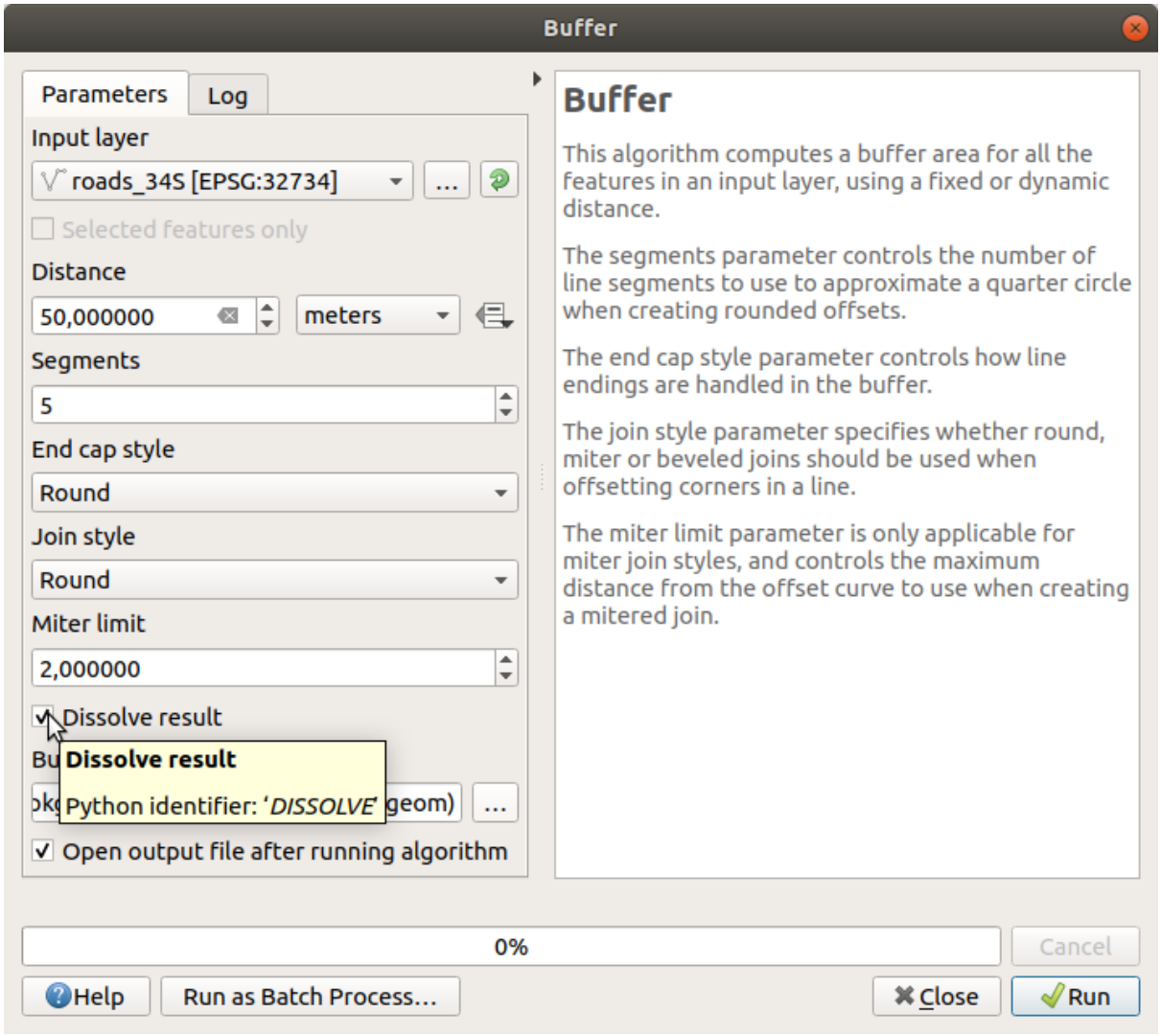
   Now your map will look something like this:



If your new layer is at the top of the *Layers* list, it will probably obscure much of your map, but this gives you all the areas in your region which are within 50m of a road.

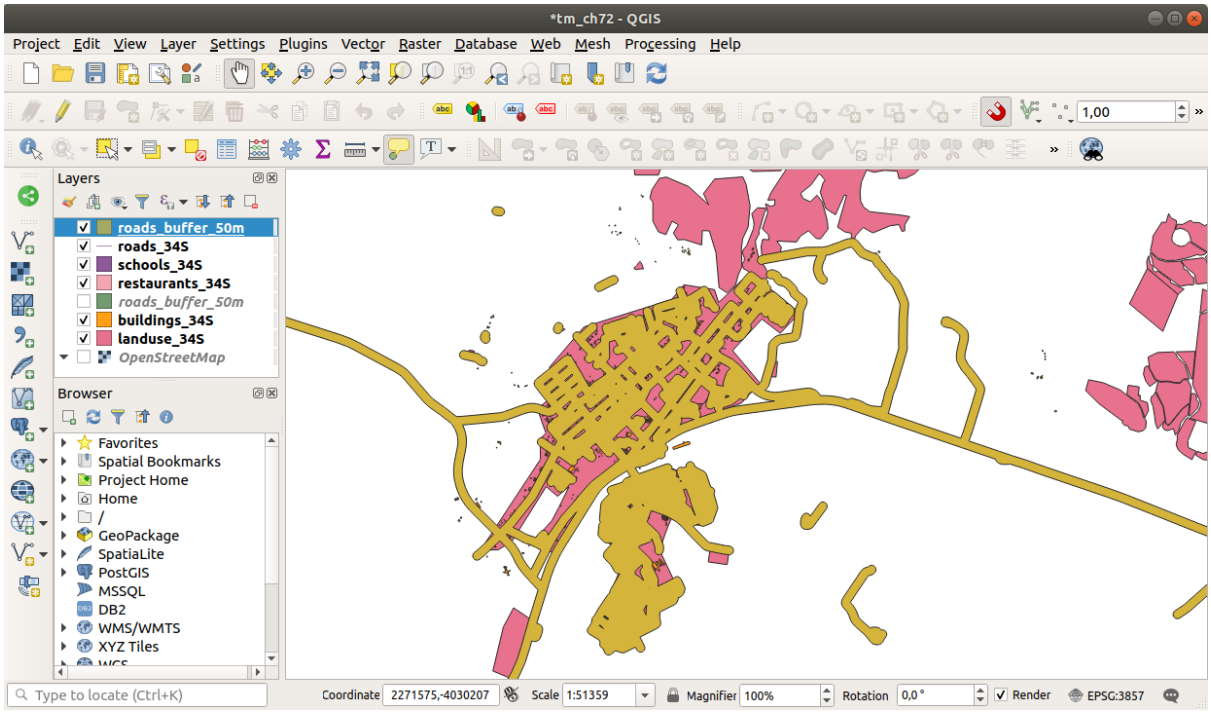
Notice that there are distinct areas within your buffer, which correspond to each individual road. To get rid of this problem:

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



2. Save the output as *roads_buffer_50m_dissolved*
3. Click *Run* and close the *Buffer* dialog

Once you have added the layer to the *Layers* panel, it will look like this:



Now there are no unnecessary subdivisions.

Bemerkung: The *Short Help* on the right side of the dialog explains how the algorithm works. If you need more information, just click on the *Help* button in the bottom part to open a more detailed guide of the algorithm.

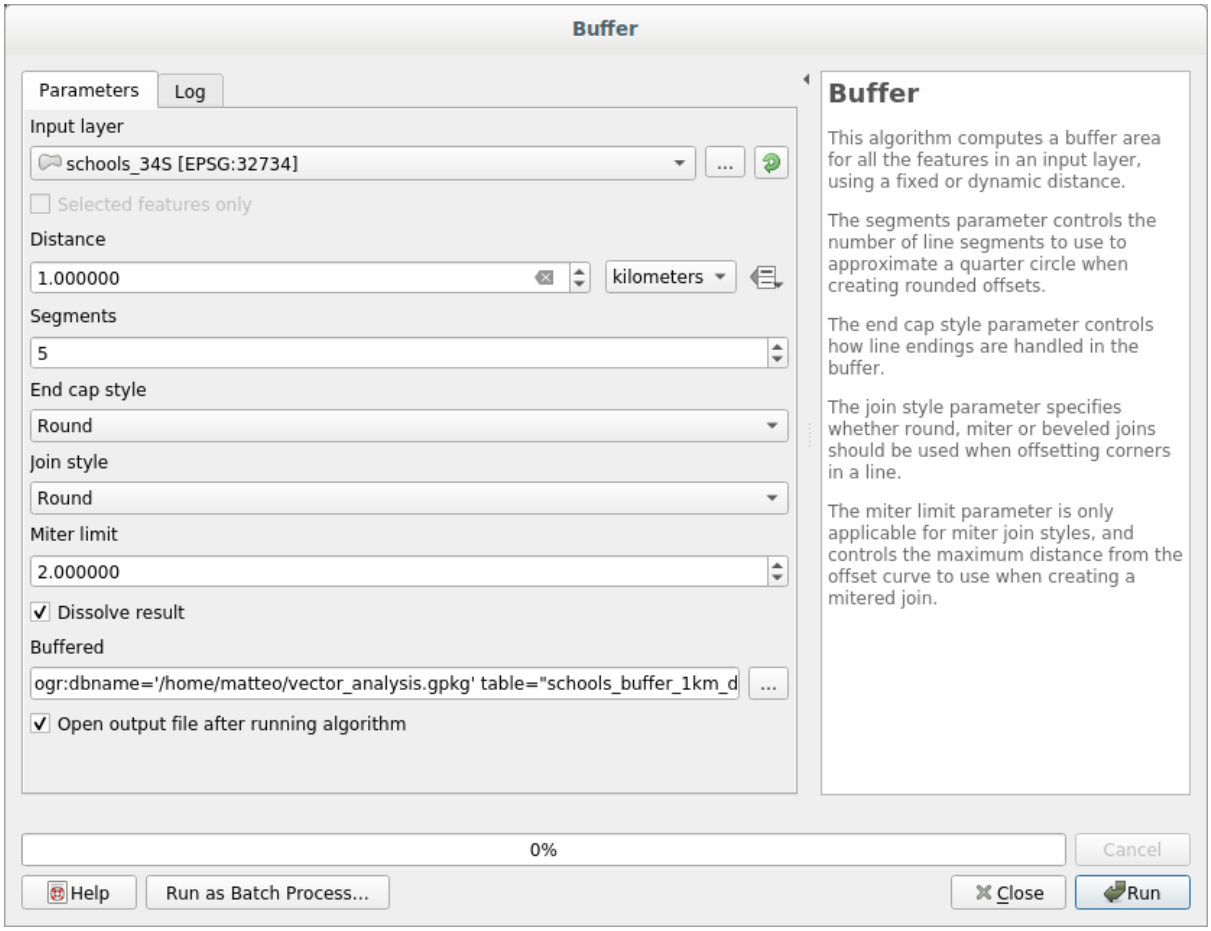
6.2.7 Try Yourself Distance from schools

Use the same approach as above and create a buffer for your schools.

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

Answer

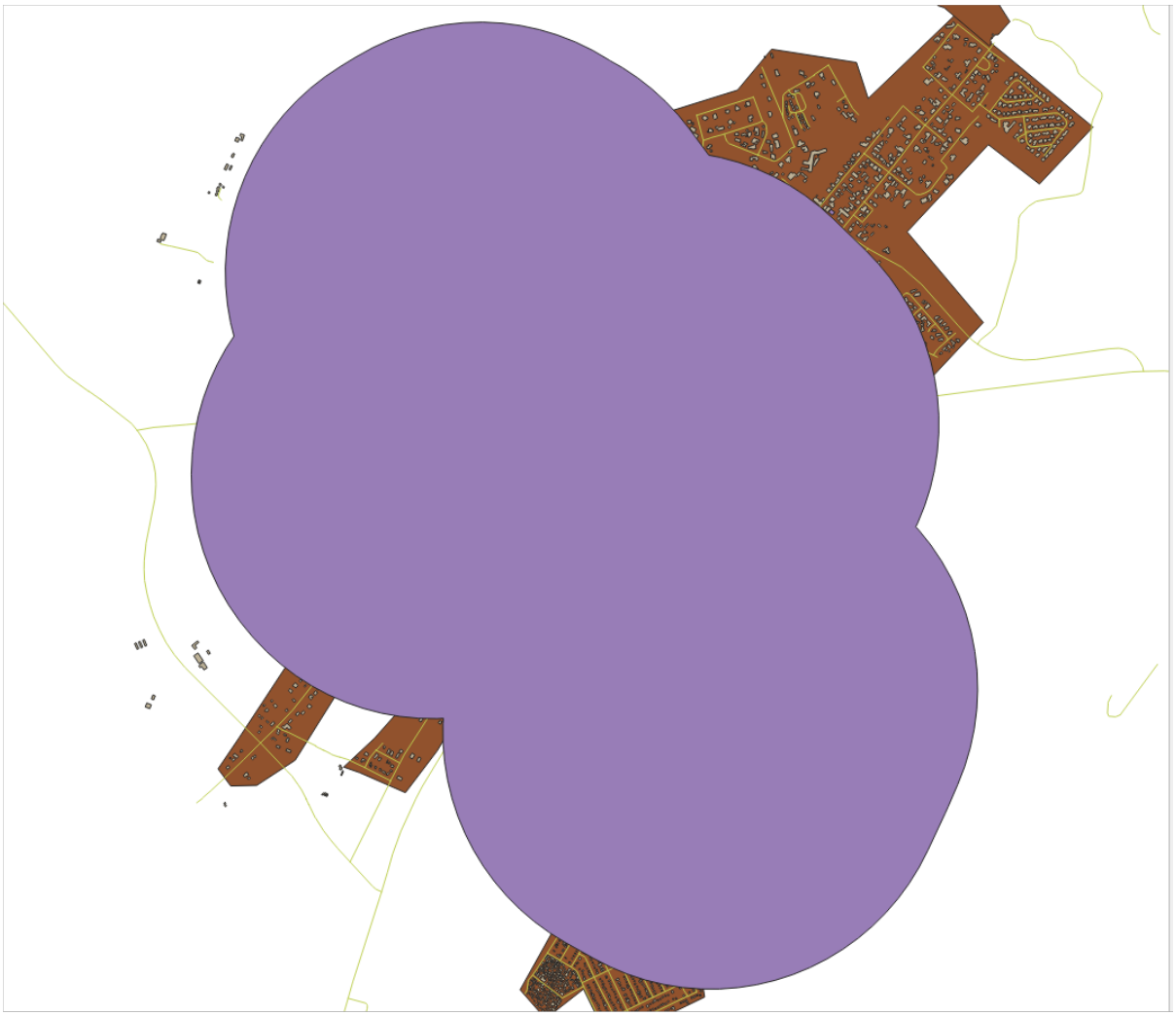
- Your buffer dialog should look like this:



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:

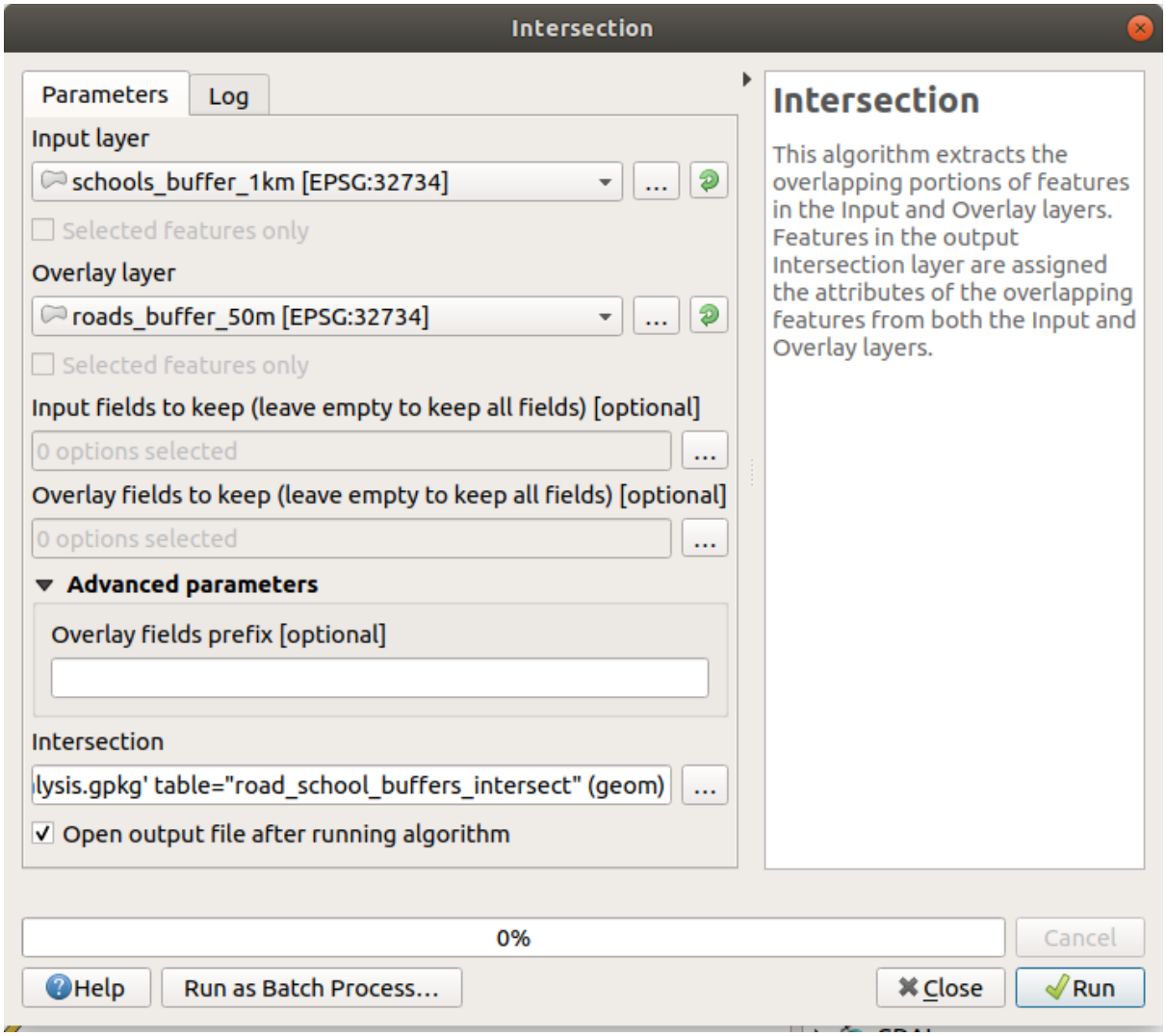


The first image shows the buffer with the *Segments to approximate* value set to *5* and the second shows the value set to *20*. In our example, the difference is subtle, but you can see that the buffer's edges are smoother with the higher value.

6.2.8 Follow Along: Overlapping Areas

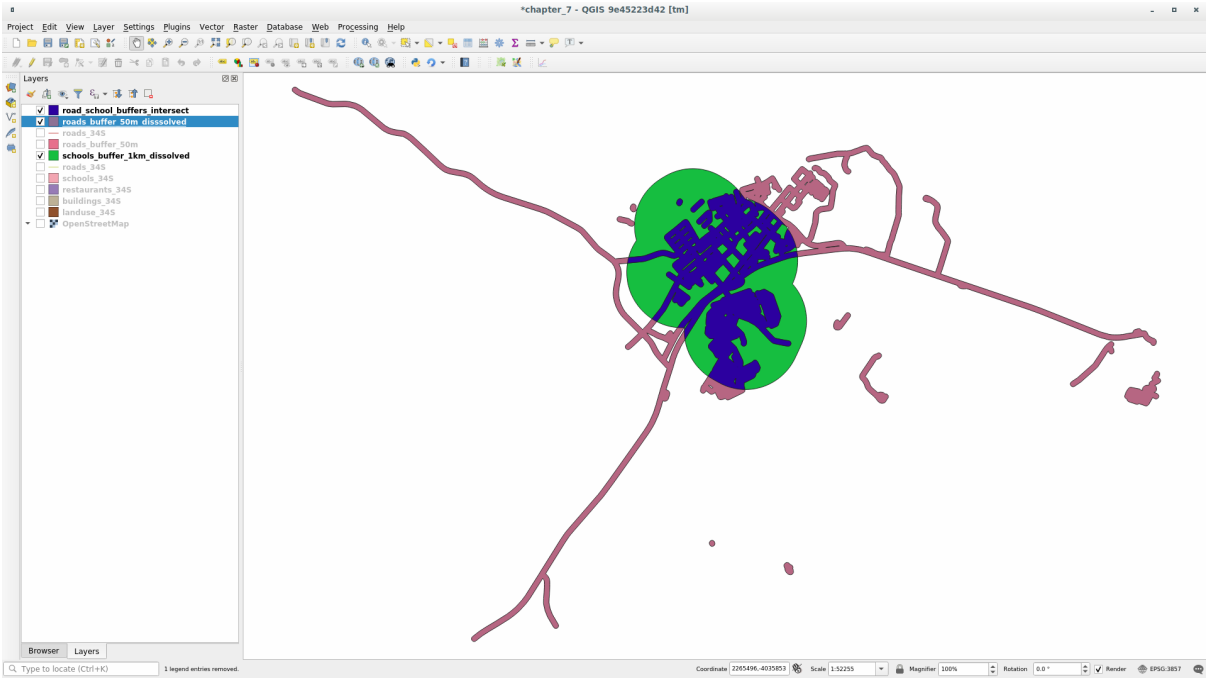
Now we have identified areas where the road is less than 50 meters away and areas where there is a school within 1 km (direct line, not by road). But obviously, we only want the areas where both of these criteria are satisfied. To do that, we will need to use the *Intersect* tool. You can find it in *Vector Overlay* group in the *Processing Toolbox*.

1. Use the two buffer layers as *Input layer* and *Overlay layer*, choose `vector_analysis.gpkg` GeoPackage in *Intersection* with *Layer name* `road_school_buffers_intersect`. Leave the rest as suggested (default).

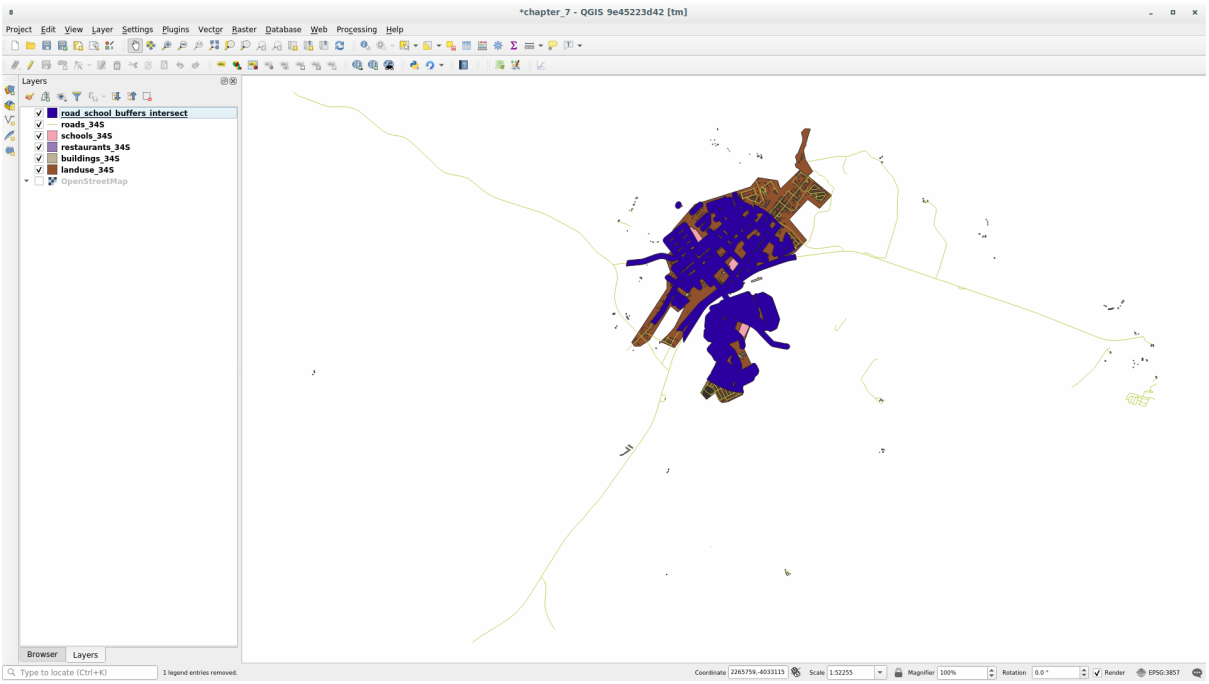


2. Klicken Sie auf *Starten*.

   In the image below, the blue areas are where both of the distance criteria are satisfied.



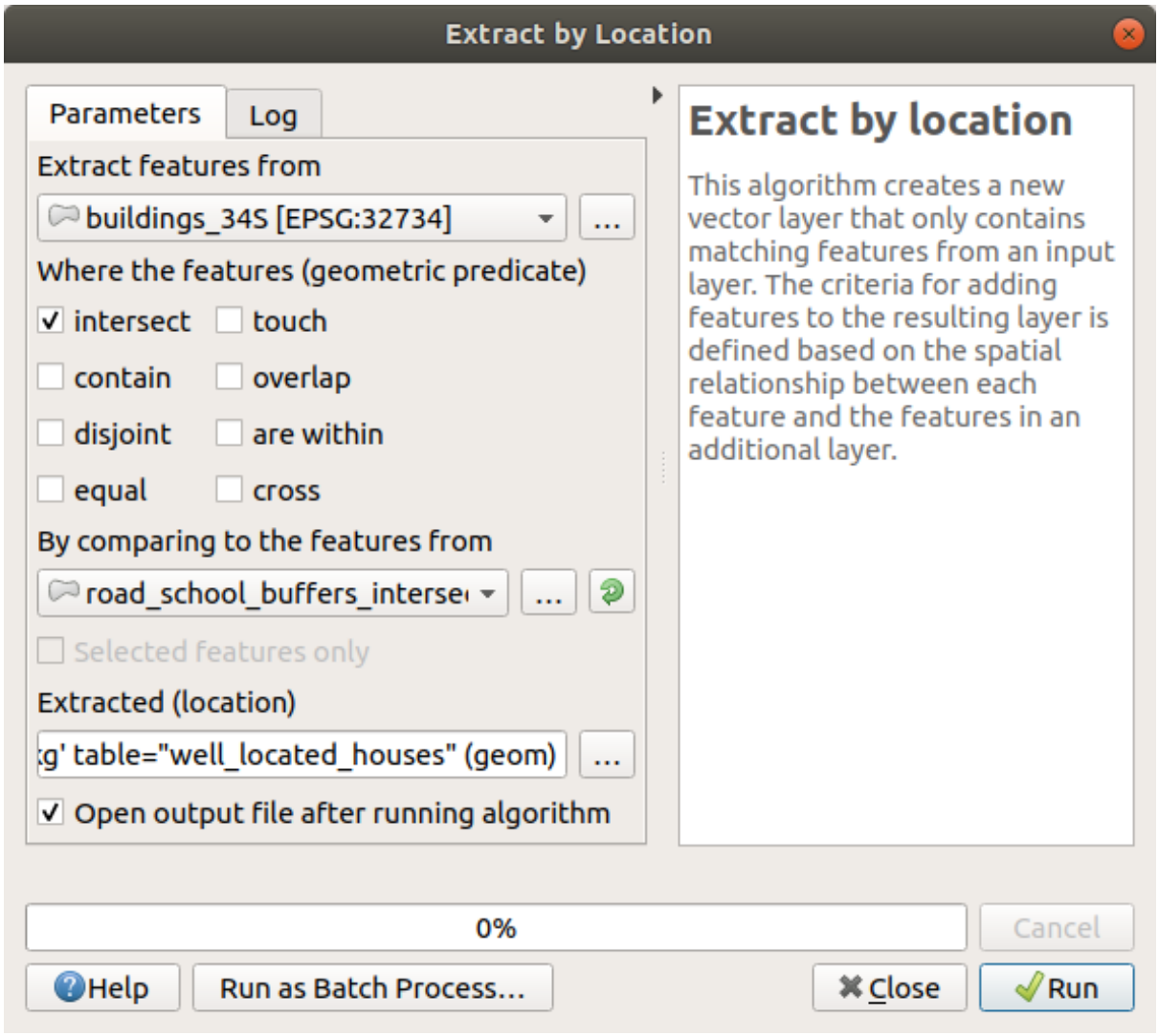
3. You may remove the two buffer layers and only keep the one that shows where they overlap, since that's what we really wanted to know in the first place:



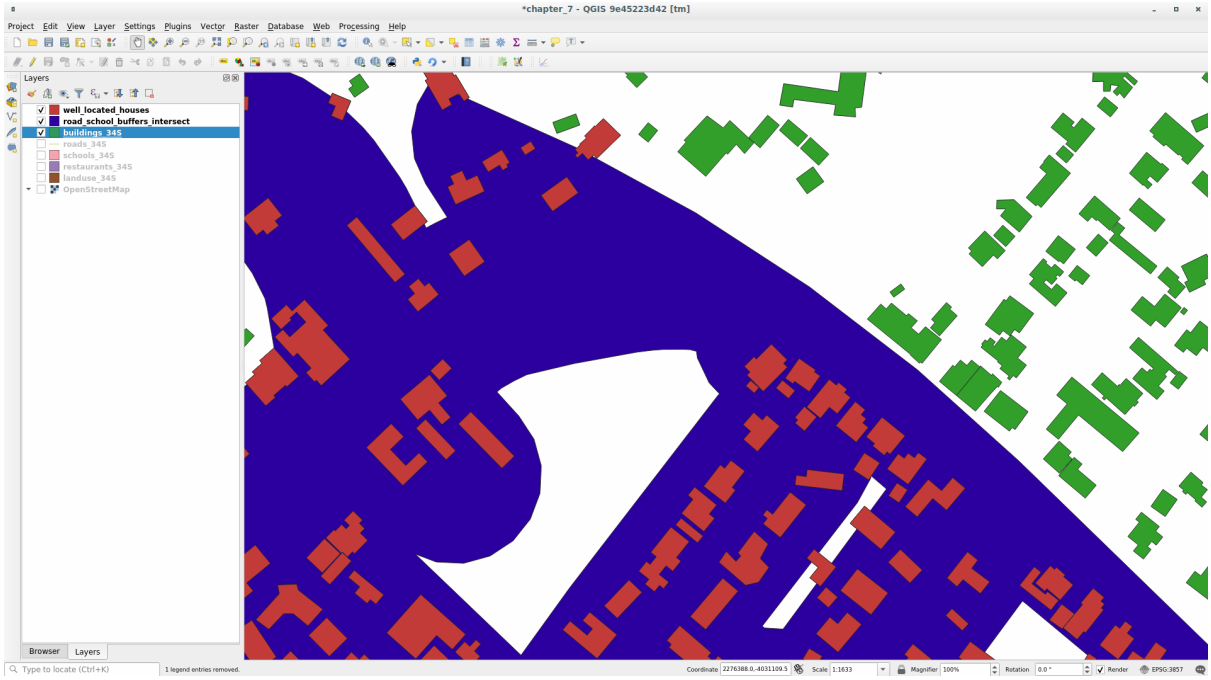
6.2.9 Follow Along: Extract the Buildings

Now you've got the area that the buildings must overlap. Next, you want to extract the buildings in that area.

1. Look for the menu entry *Vector Selection* ► *Extract by location* within the *Processing Toolbox*
2. Select `buildings_34S` in *Extract features from*. Check *intersect* in *Where the features (geometric predicate)*, select the buffer intersection layer in *By comparing to the features from*. Save to the `vector_analysis.gpkg`, and name the layer `well_located_houses`.



3. Click *Run* and close the dialog
4. You will probably find that not much seems to have changed. If so, move the *well_located_houses* layer to the top of the layers list, then zoom in.



The red buildings are those which match our criteria, while the buildings in green are those which do not.

5. Now you have two separated layers and can remove `buildings_34S` from the layer list.

6.2.10 Try Yourself Further Filter our Buildings

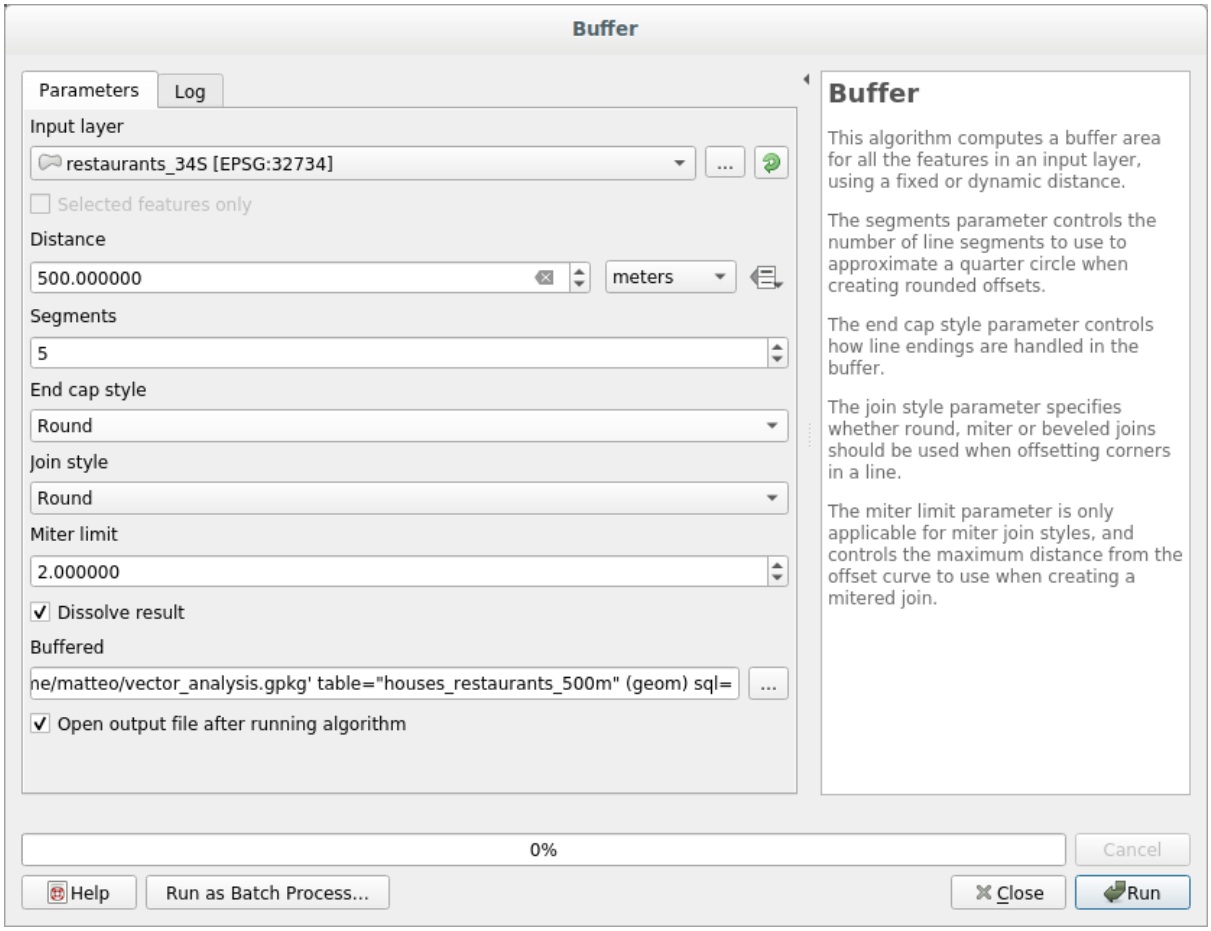
We now have a layer which shows us all the buildings within 1km of a school and within 50m of a road. We now need to reduce that selection to only show buildings which are within 500m of a restaurant.

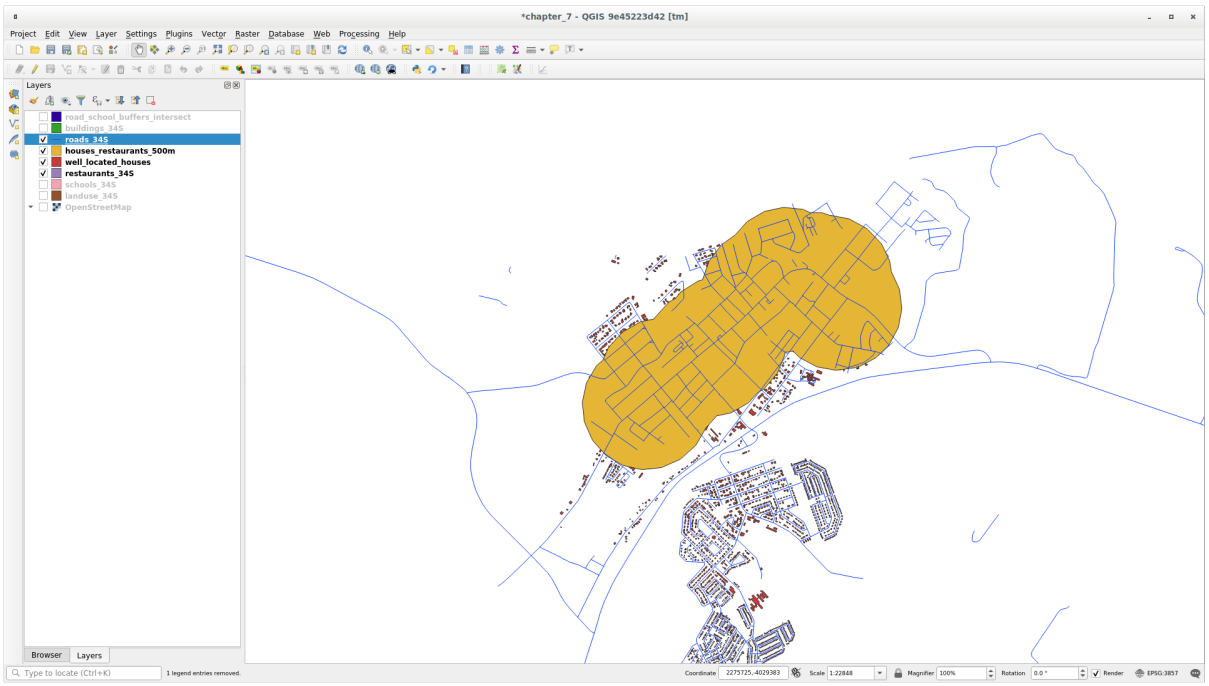
Using the processes described above, create a new layer called *houses_restaurants_500m* which further filters your *well_located_houses* layer to show only those which are within 500m of a restaurant.

Answer

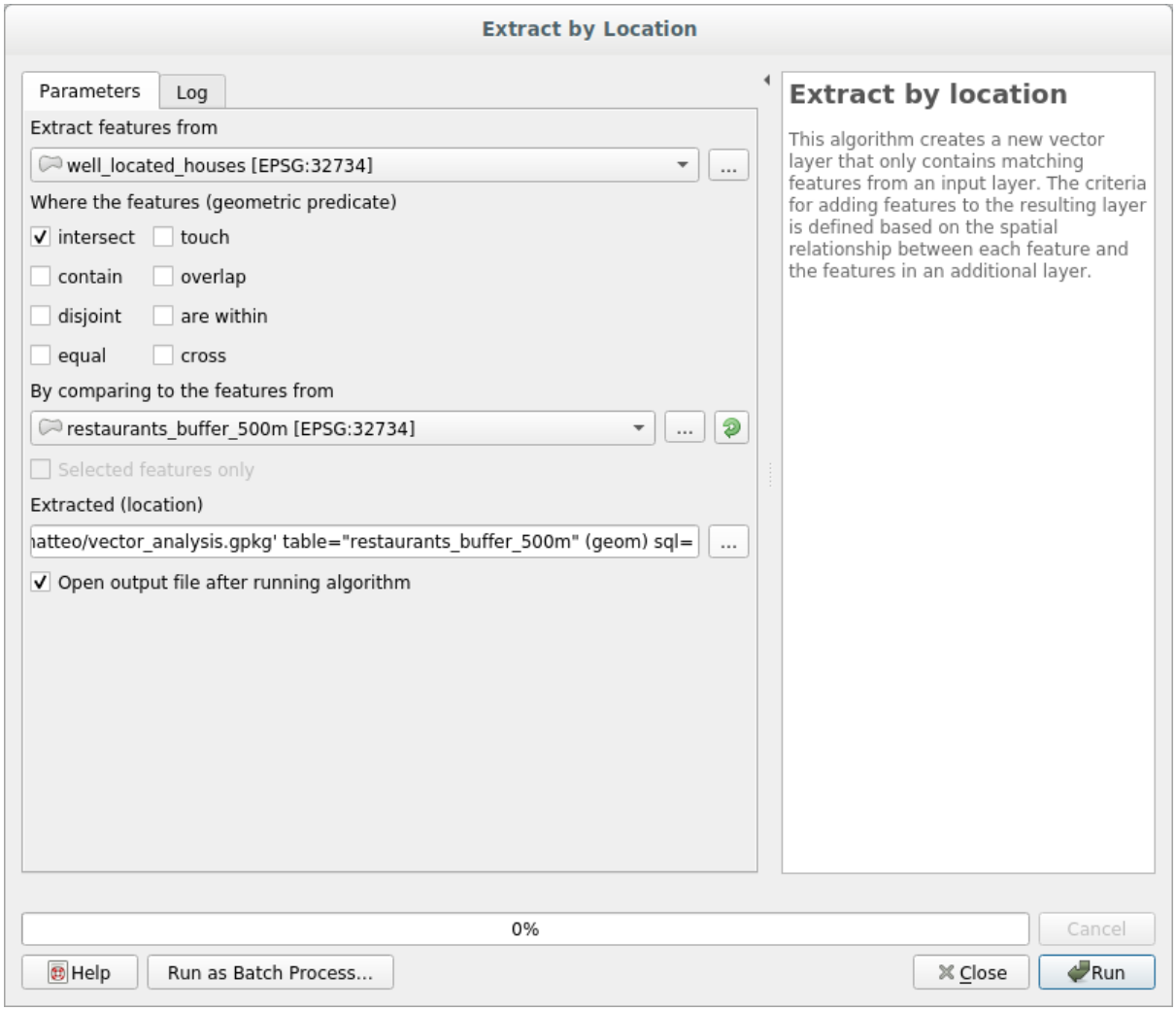
To create the new *houses_restaurants_500m* layer, we go through a two step process:

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

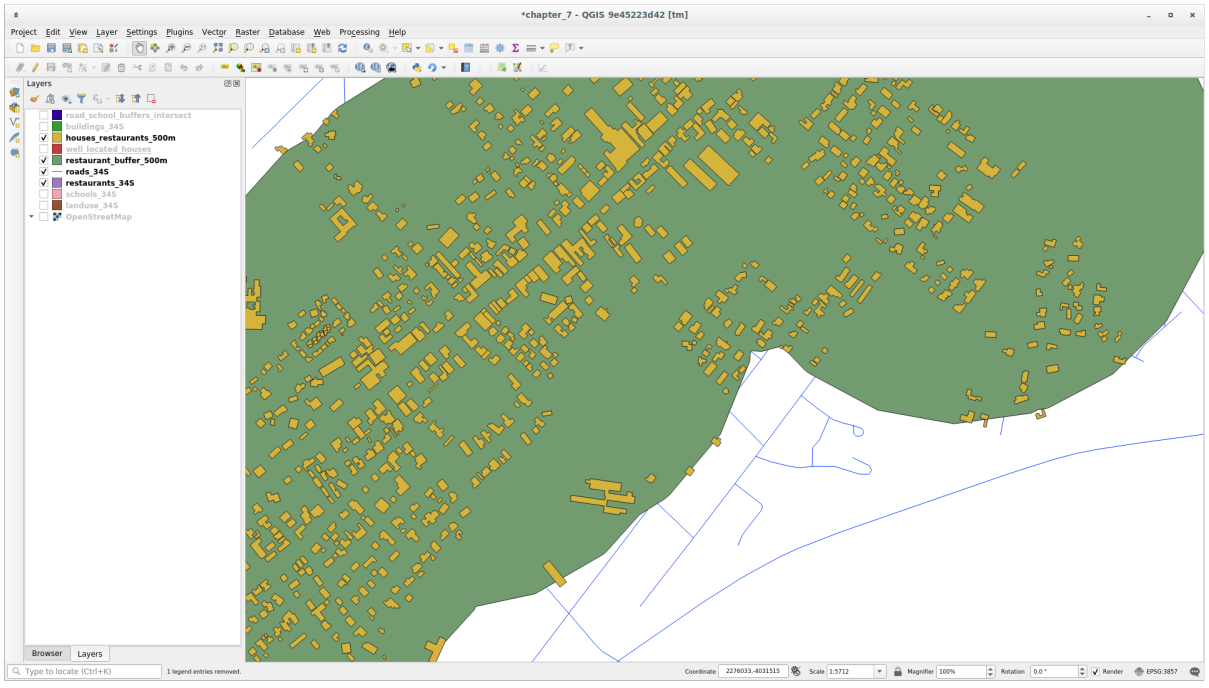




2. Next, extract buildings within that buffer area:



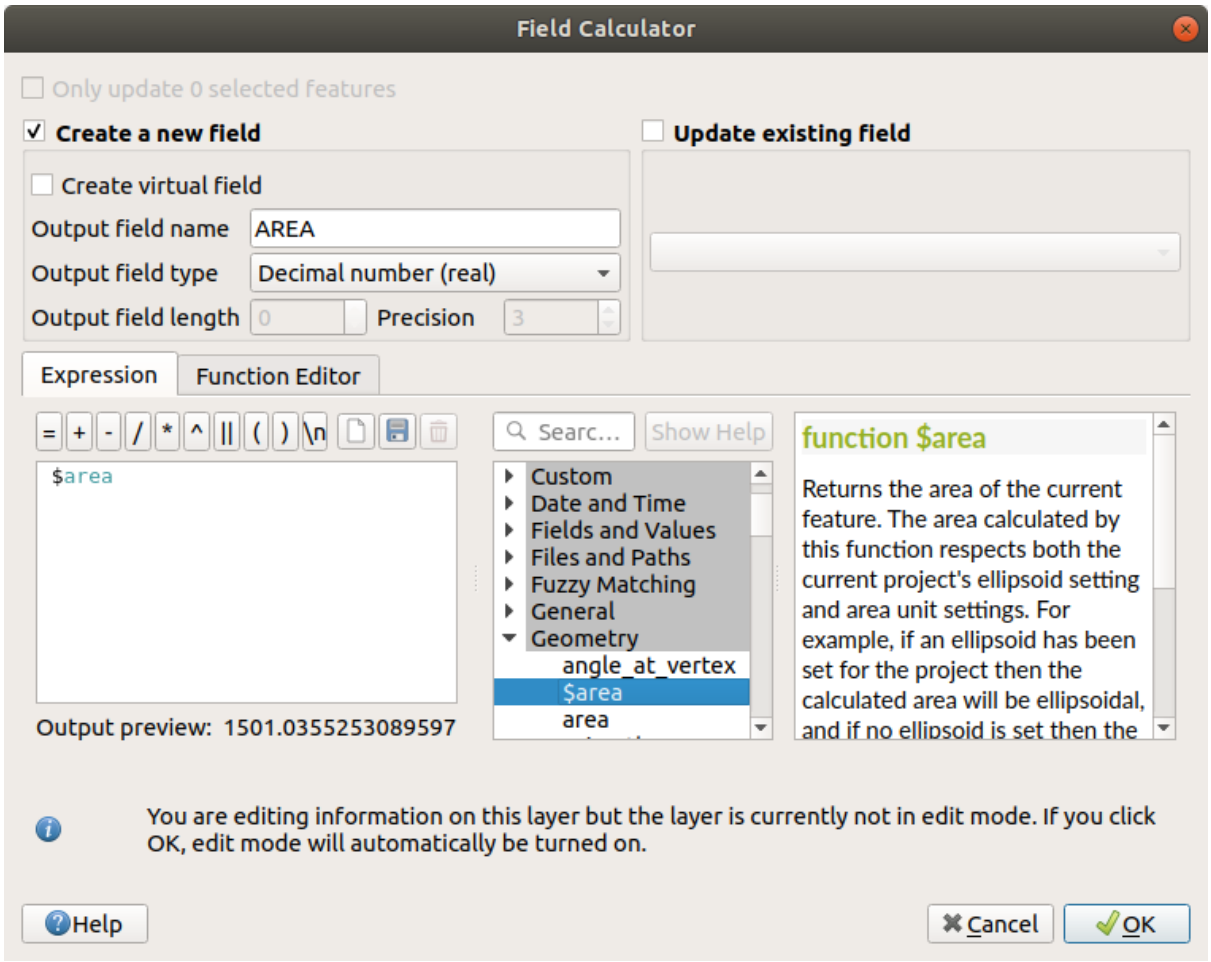
Your map should now show only those buildings which are within 50m of a road, 1km of a school and 500m of a restaurant:



6.2.11 Follow Along: Select Buildings of the Right Size

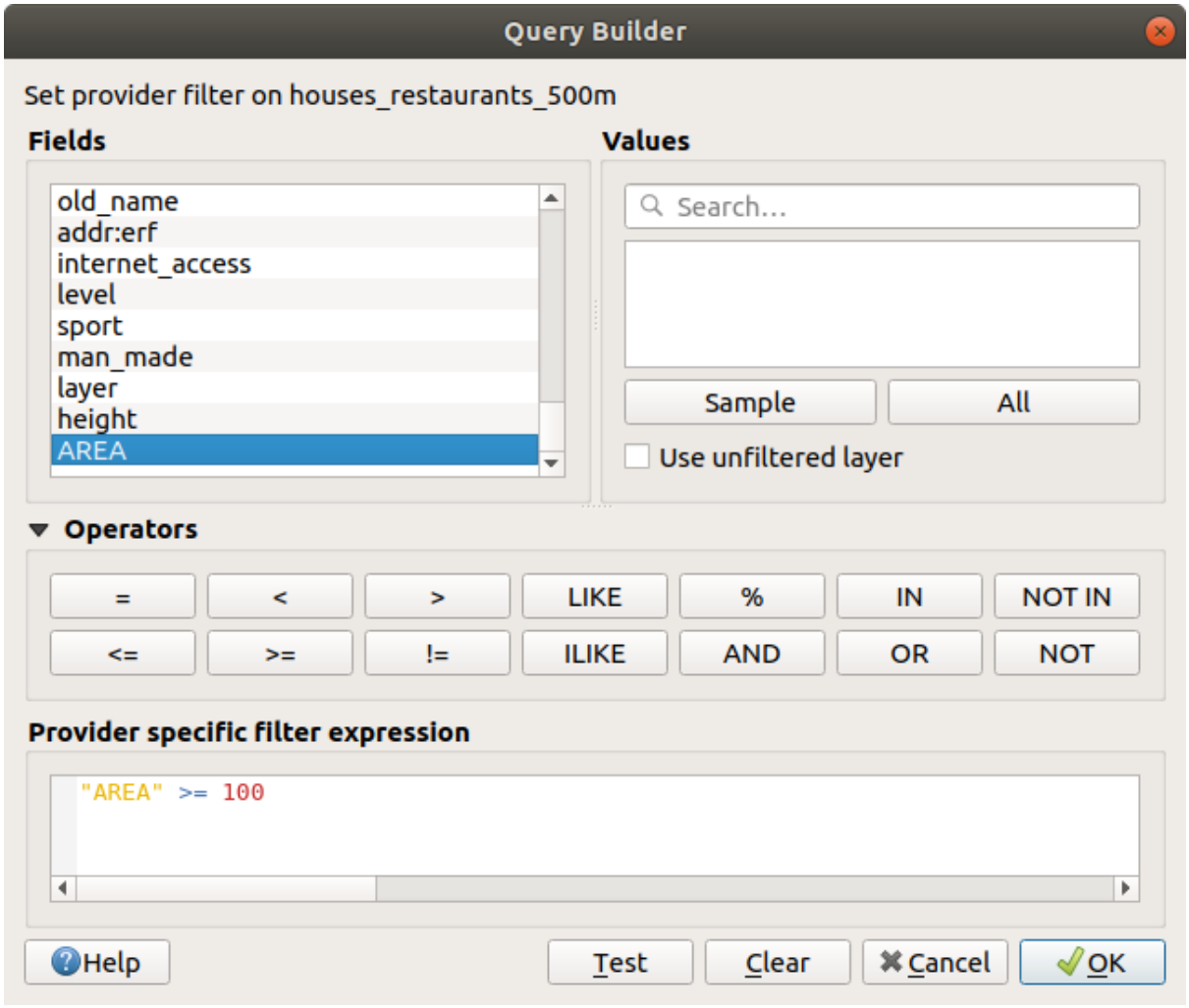
To see which buildings are of the correct size (more than 100 square meters), we need to calculate their size.

1. Select the *houses_restaurants_500m* layer and open the *Field Calculator* by clicking on the Open Field Calculator button in the main toolbar or in the attribute table window
2. Select *Create a new field*, set the *Output field name* to `AREA`, choose *Decimal number (real)* as *Output field type*, and choose `$area` from the *Geometry* group.



   The new field `AREA` will contain the area of each building in square meters.

3. Click *OK*. The `AREA` field has been added at the end of the attribute table.
4. Click the Toggle Editing button to finish editing, and save your edits when prompted.
5. In the *Source* tab of the layer properties, set the *Provider Feature Filter* to `"AREA >= 100`.



6. Click *OK*.

Your map should now only show you those buildings which match our starting criteria and which are more than 100 square meters in size.

6.2.12 Try Yourself

Save your solution as a new layer, using the approach you learned above for doing so. The file should be saved within the same GeoPackage database, with the name `solution`.

6.2.13 In Conclusion

Using the GIS problem solving approach together with QGIS vector analysis tools, you were able to solve a problem with multiple criteria quickly and easily.

6.2.14 What's Next?

In the next lesson, we will look at how to calculate the shortest distance along roads from one point to another.

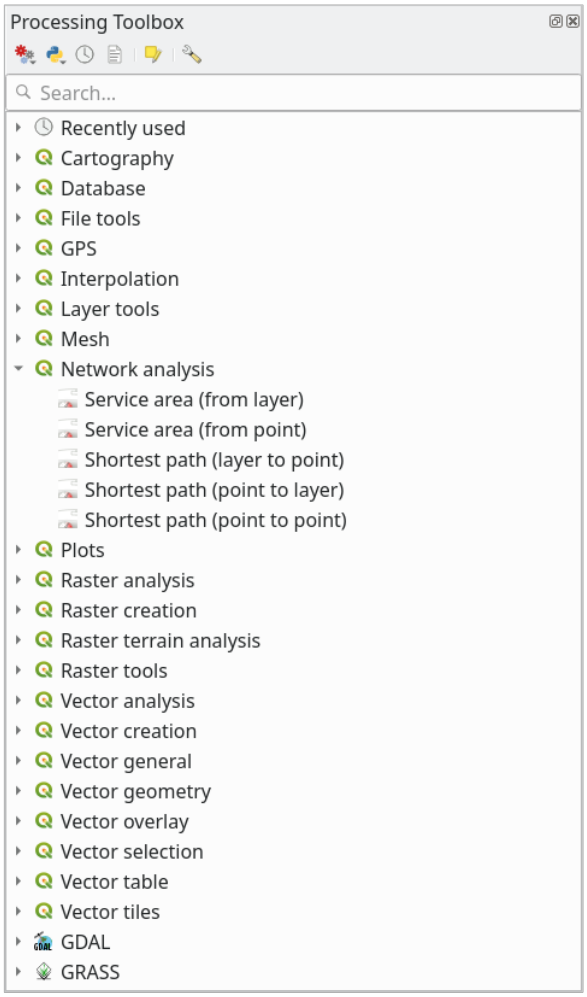
6.3 Lesson: Netzwerkanalyse

Calculating the shortest distance between two points is a common GIS task. Tools for this can be found in the *Processing Toolbox*.

The goal for this lesson: learn to use *Network analysis* algorithms.

6.3.1 Follow Along: The Tools and the Data

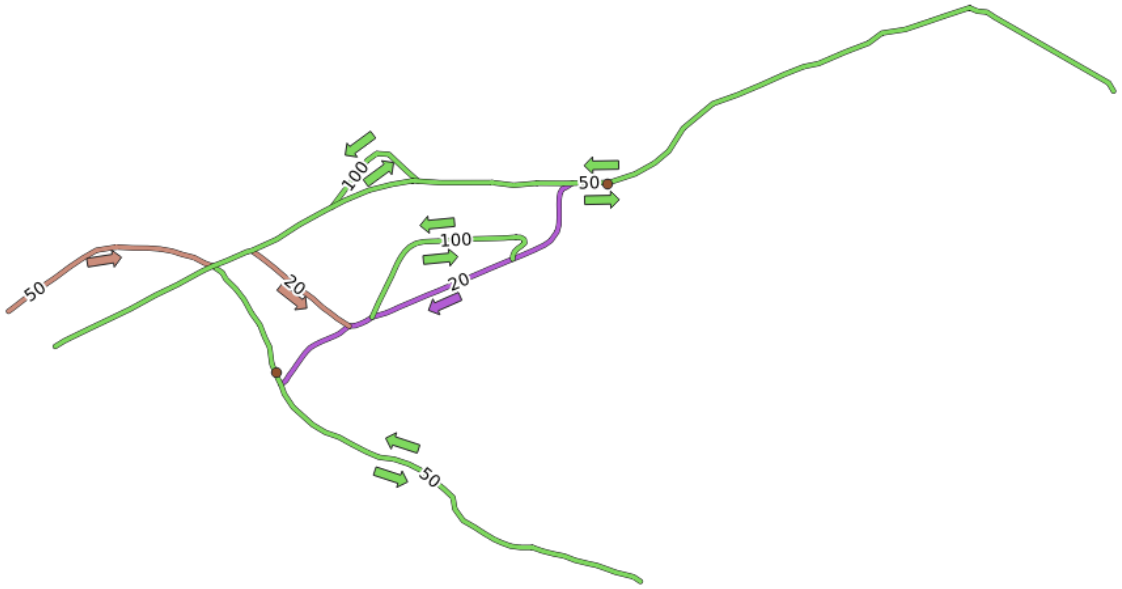
You can find all the network analysis algorithms in the *Processing* ► *Network Analysis* menu. You can see that there are many tools available:



Open the project `exercise_data/network_analysis/network.qgz`. It contains two layers:

- `network_points`
- `network_lines`

The *network_lines* layer has already a style that helps to understand the road network.



The shortest path tools provide ways to calculate either the shortest or the fastest path between two points of a network, given:

- start and end points selected on the map
- start point selected on the map and end points taken from a point layer
- start points taken from a point layer and end point selected on the map

Let's start.

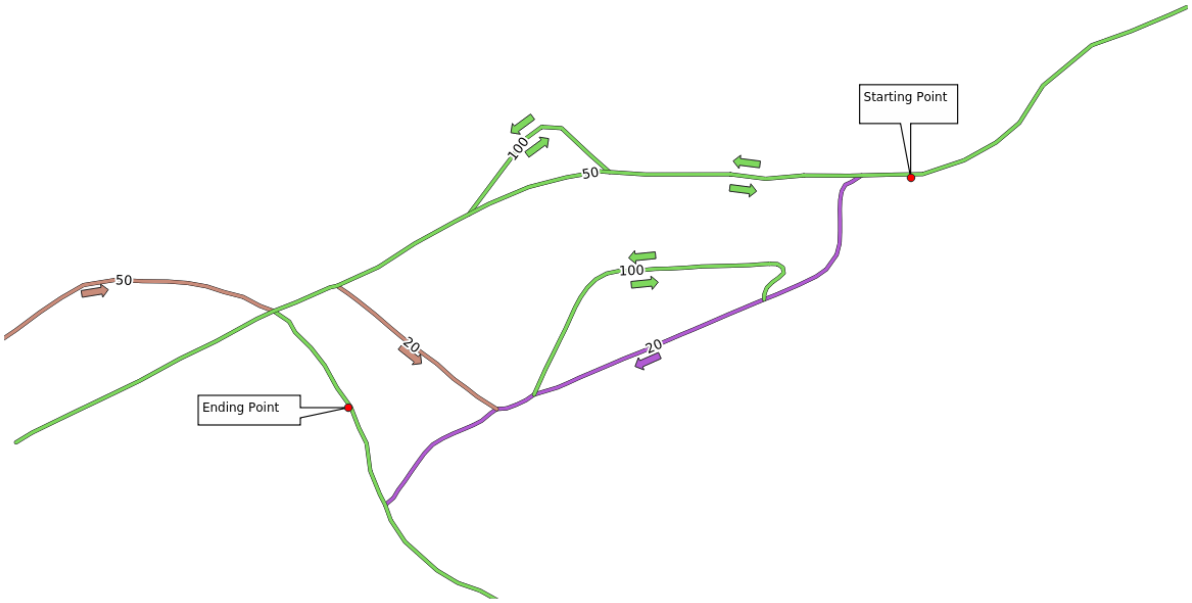
6.3.2 Calculate the shortest path (point to point)

The *Network analysis* ► *Shortest path (point to point)* allows you to calculate the shortest distance between two manually selected points on the map.

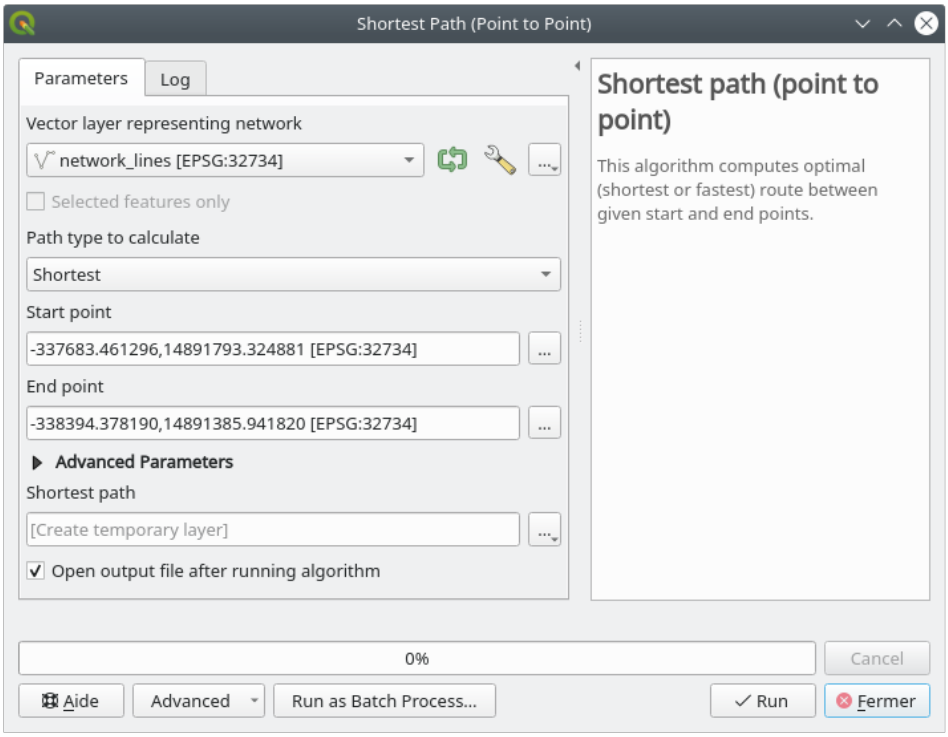
In this example we will calculate the **shortest** (not fastest) path between two points.

1. Open the *Shortest path (point to point)* algorithm
2. Select *network_lines* for *Vector layer representing network*
3. Use `Shortest` for *Path type to calculate*

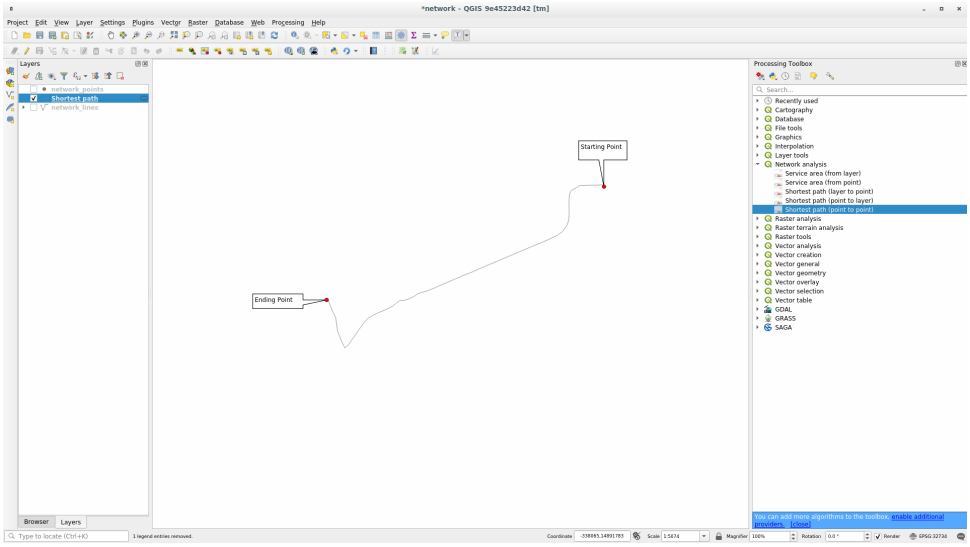
   Use these two points as starting and ending points for the analysis:



4. Click on the … button next to *Start point (x, y)* and choose the location tagged with `Starting Point` in the picture. Enable the snapping options for an accurate selection. The coordinates of the clicked point are added.
5. Do the same thing, but choosing the location tagged with `Ending point` for *End point (x, y)*
6. Click on the *Run* button:



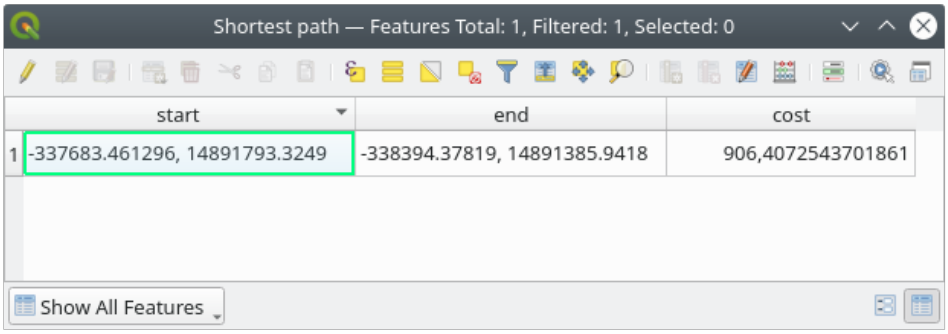
7. A new line layer is created representing the shortest path between the chosen points. Uncheck the `network_lines` layer to see the result better:



8. Open the attribute table of the output layer. It contains three fields, representing the coordinates of the start and end points and the **cost**.

   We chose `Shortest` as *Path type to calculate*, so the **cost** represent the **distance**, in layer units, between the two locations.

   In our case, the *shortest* distance between the chosen points is around `1000` meters:



Now that you know how to use the tool, feel free to test other locations.

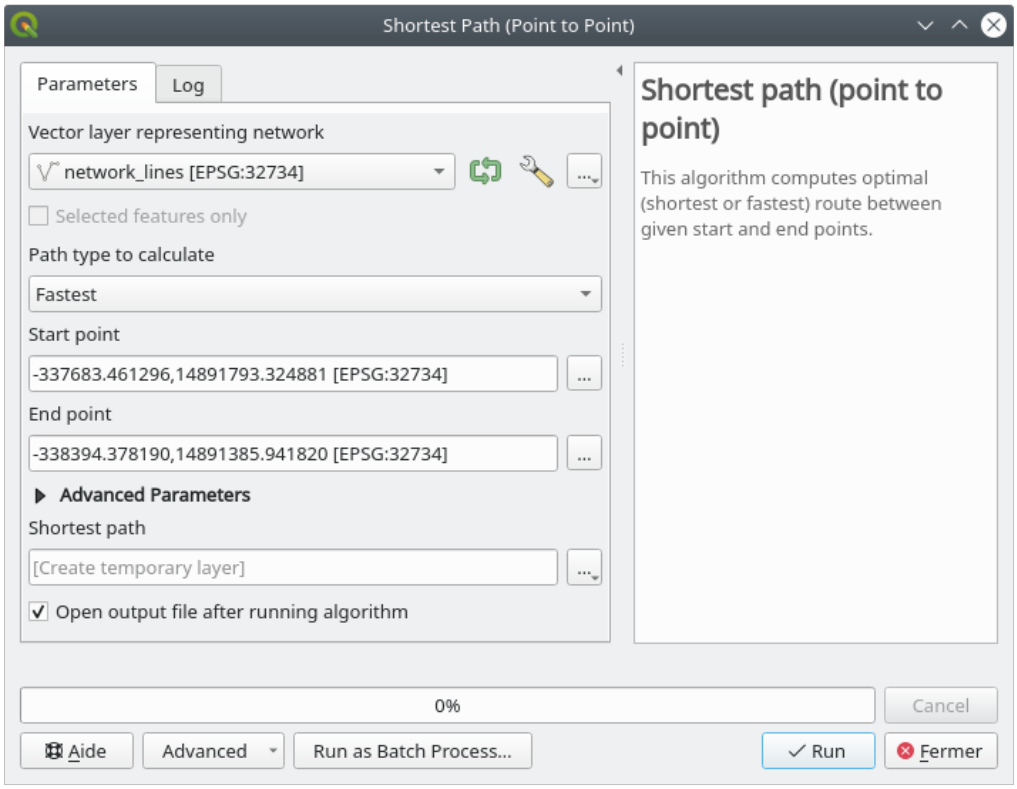
6.3.3 Try Yourself Fastest path

With the same data of the previous exercise, try to calculate the fastest path between the two points.

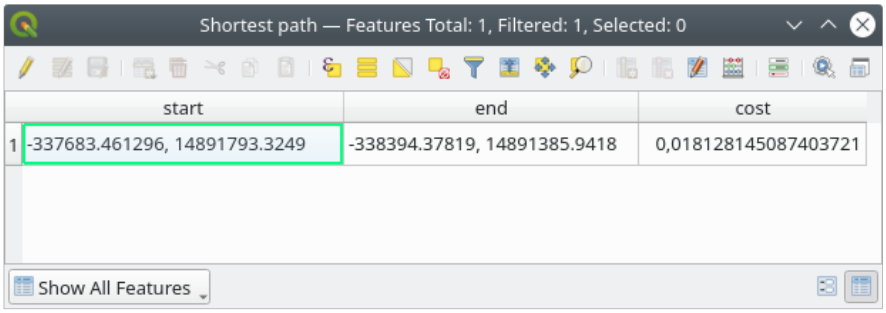
How much time do you need to go from the start to the end point?

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):



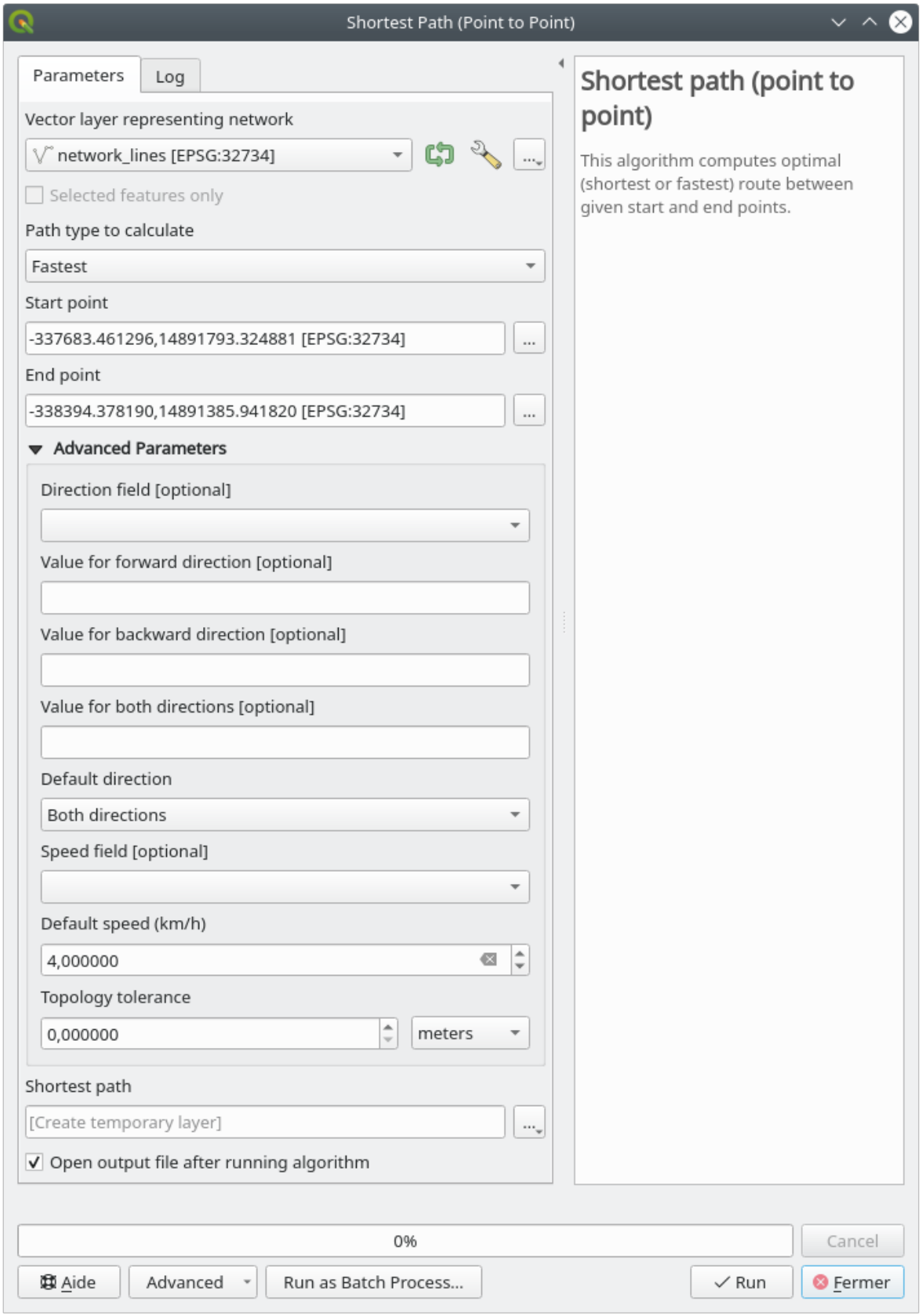
6.3.4 Follow Along: Advanced options

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**.

We will use the same layers and starting and ending points of the previous exercises.

1. Open the *Shortest path (point to point)* algorithm
2. Fill the *Input layer*, *Start point (x, y)* and *End point (x, y)* as we did before

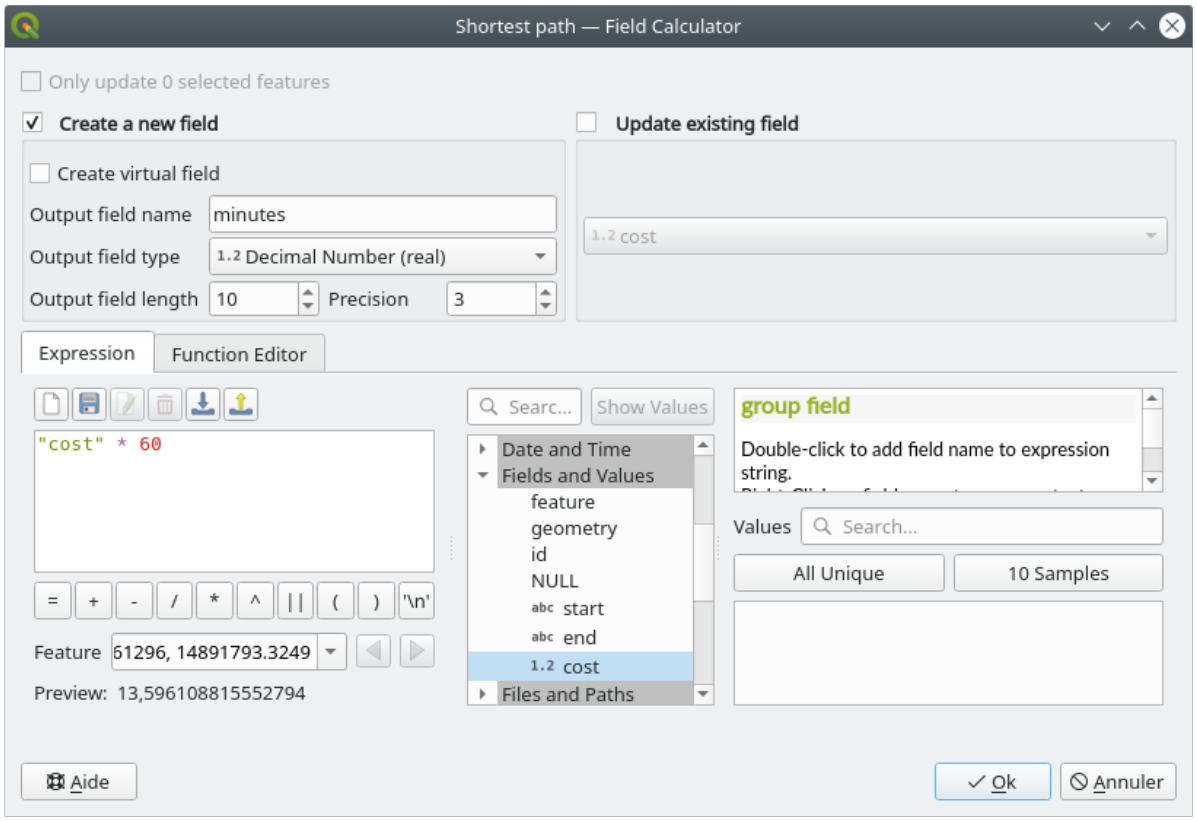
3. Choose `Fastest` as the *Path type to calculate*
4. Open the *Advanced parameter* menu
5. Change the *Default speed (km/h)* from the default `50` value to `4`



6. Click on *Run*
7. Once the algorithm is finished, close the dialog and open the attribute table of the output layer.

The *cost* field contains the value according to the speed parameter you have chosen. We can convert the *cost* field from hours with fractions to the more readable *minutes* values.

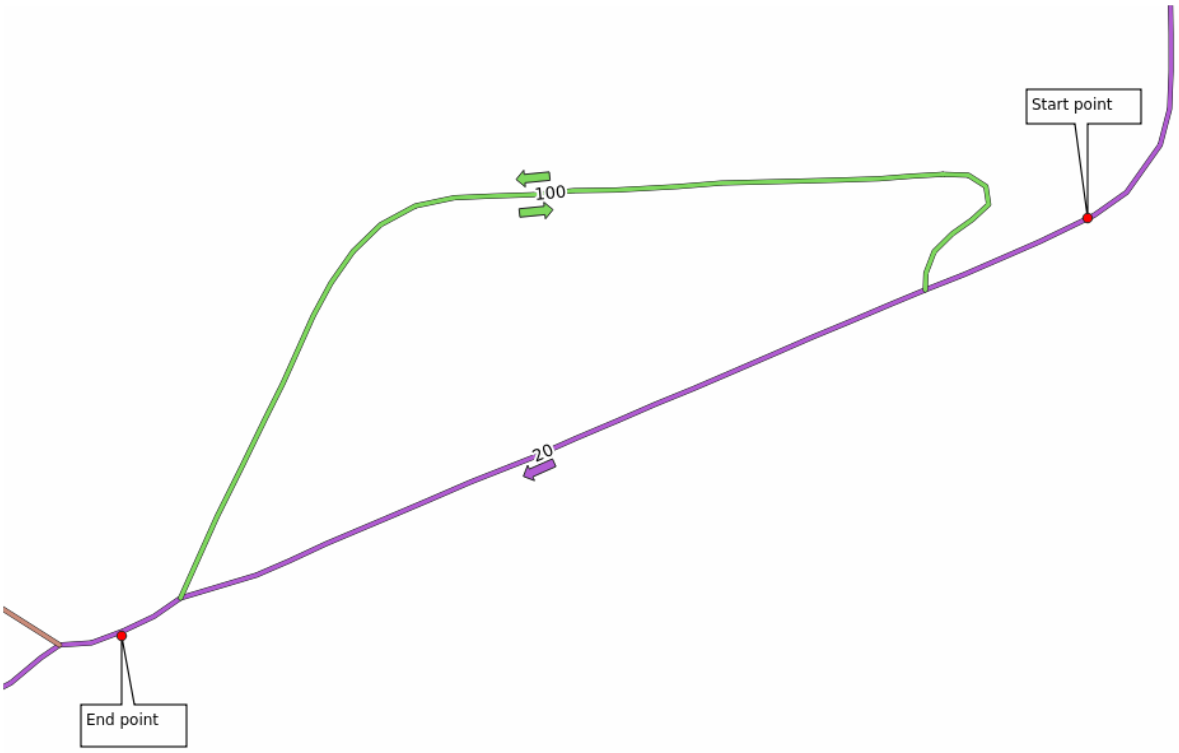
8. Open the field calculator by clicking on the icon
9. Add a new field to store the path cost in *minutes*.



That's it! Now you know how many minutes it will take to get from one point to the other one if the whole network speed is at `4 km/h`.

6.3.5 Shortest path with speed limit

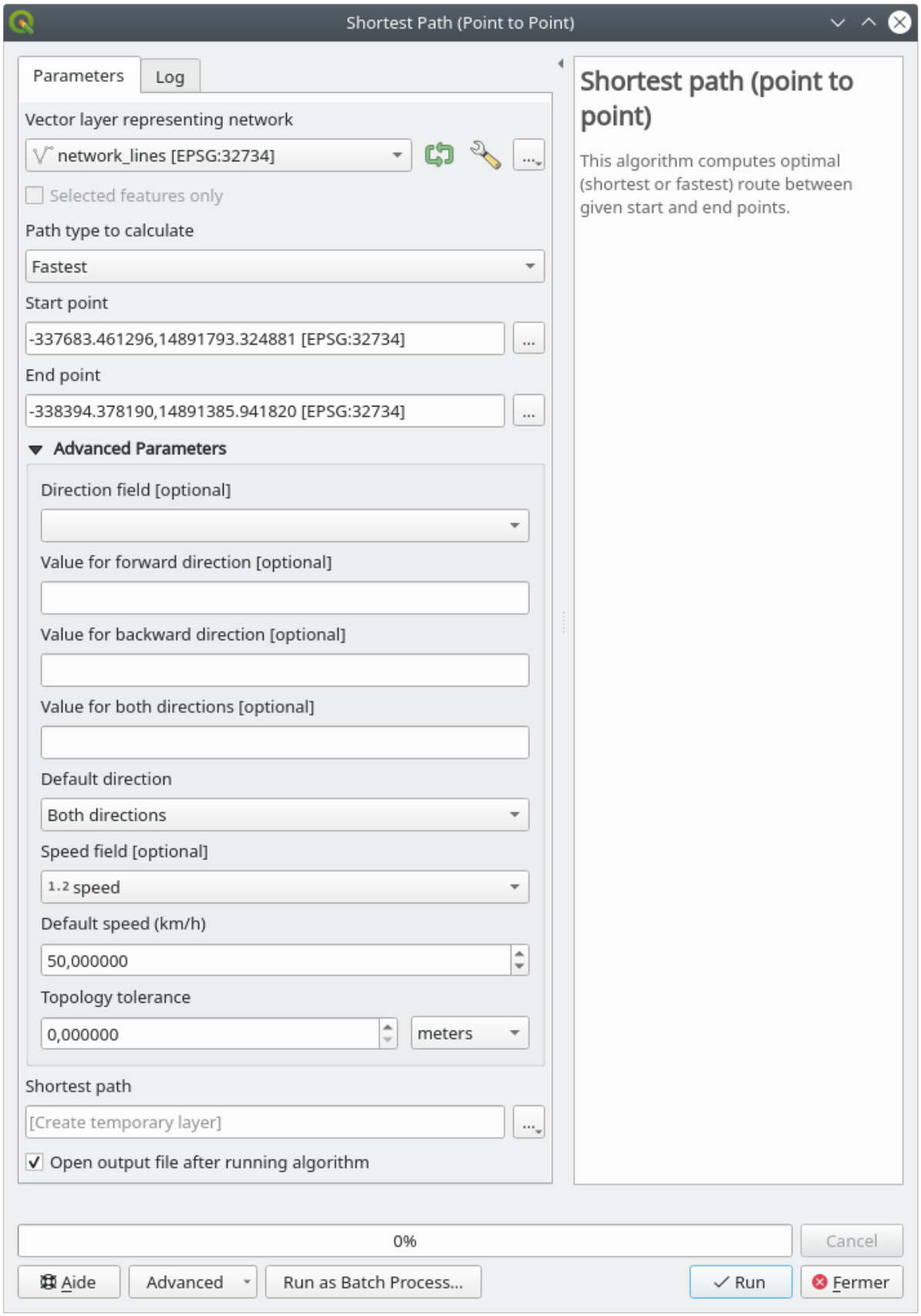
The Network analysis toolbox has other interesting options. Looking at the following map:



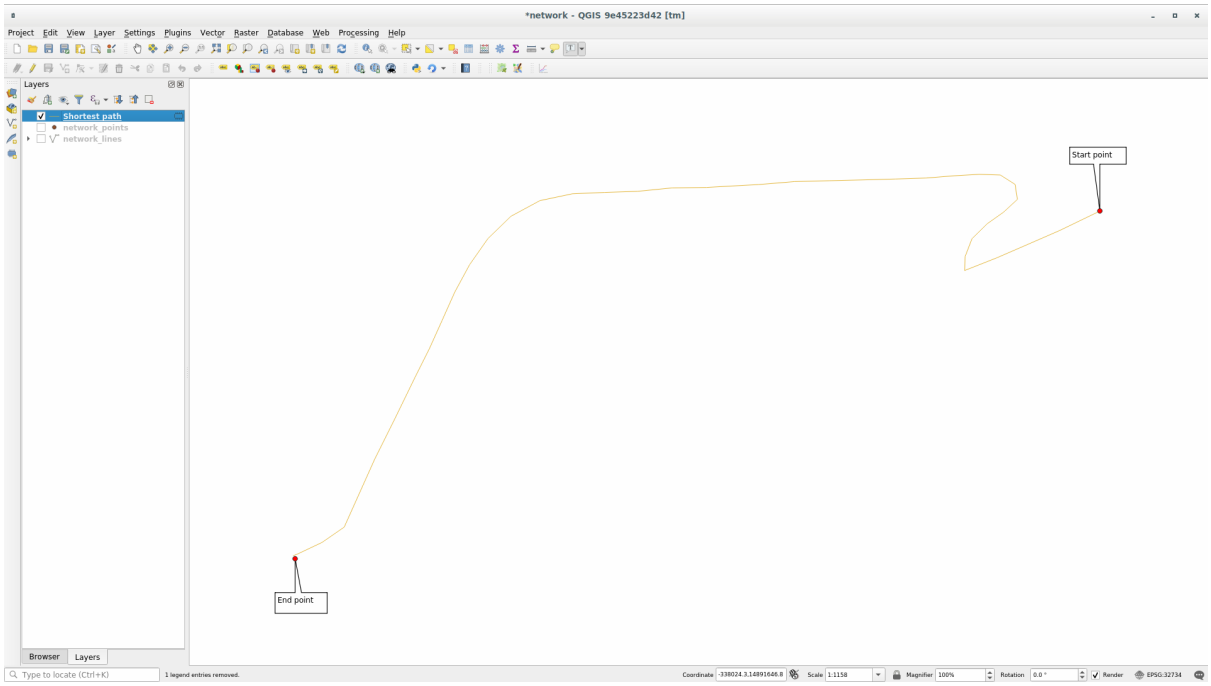
we would like to know the **fastest** route considering the **speed limits** of each road (the labels represent the speed limits in km/h). The shortest path without considering speed limits would of course be the purple path. But in that road the speed limit is 20 km/h, while in the green road you can go at 100 km/h!

As we did in the first exercise, we will use the *Network analysis* ► *Shortest path (point to point)* and we will manually choose the start and end points.

1. Open the *Network analysis* ► *Shortest path (point to point)* algorithm
2. Select *network_lines* for the *Vector layer representing network* parameter
3. Choose `Fastest` as the *Path type to calculate*
4. Select the *Start point (x, y)* and *End point (x, y)* as we did before
5. Open the *Advanced parameters* menu
6. Choose the *speed* field as the *Speed Field* parameter. With this option the algorithm will take into account the speed limits for each road.



7. Click on the *Run* button
8. Turn off the `network_lines` layer to better see the result



As you can see the fastest route does not correspond to the shortest one.

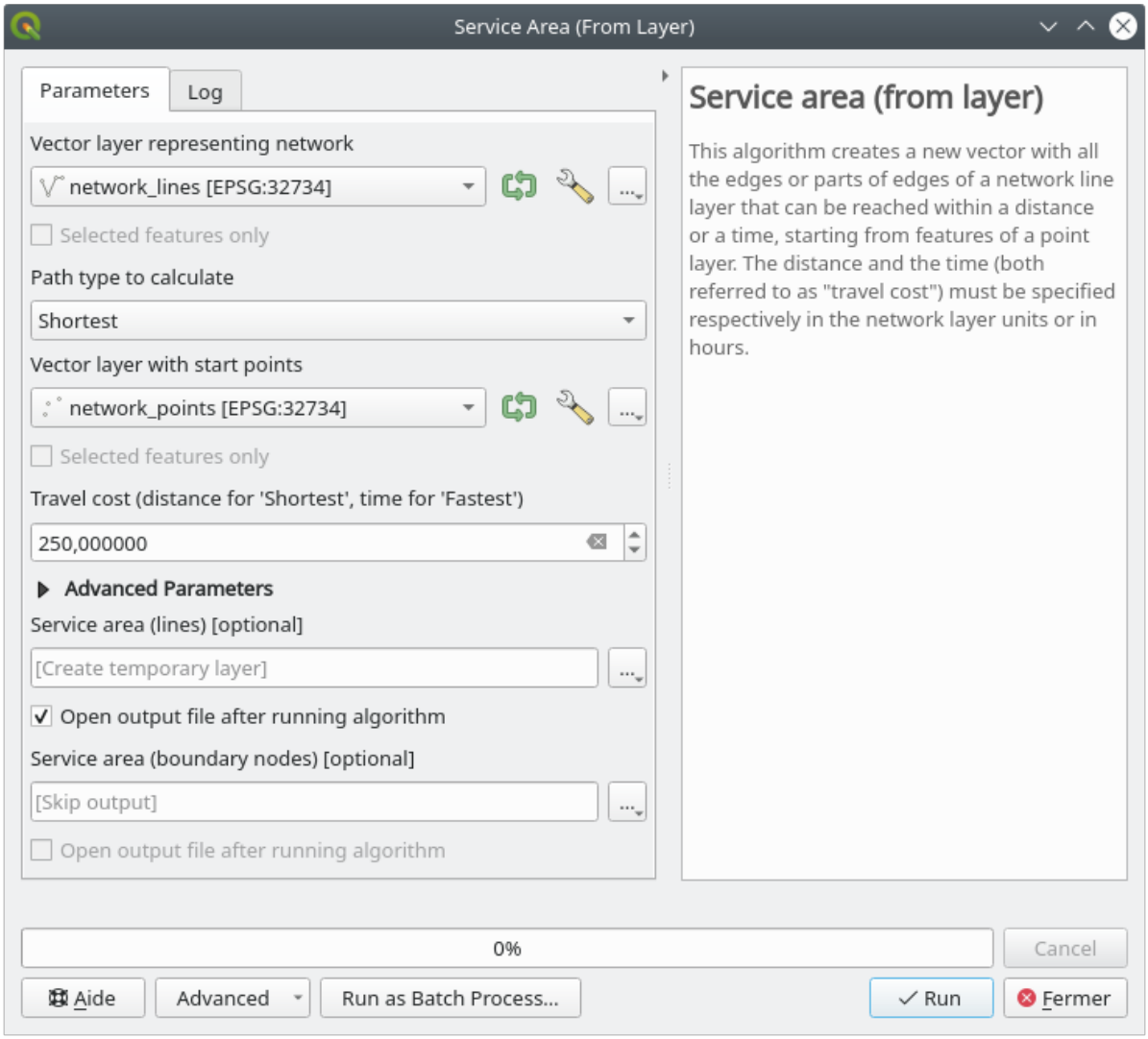
6.3.6 Service area (from layer)

The *Network Analysis* ► *Service area (from layer)* algorithm can answer the question: given a point layer, what are all the reachable areas given a distance or a time value?

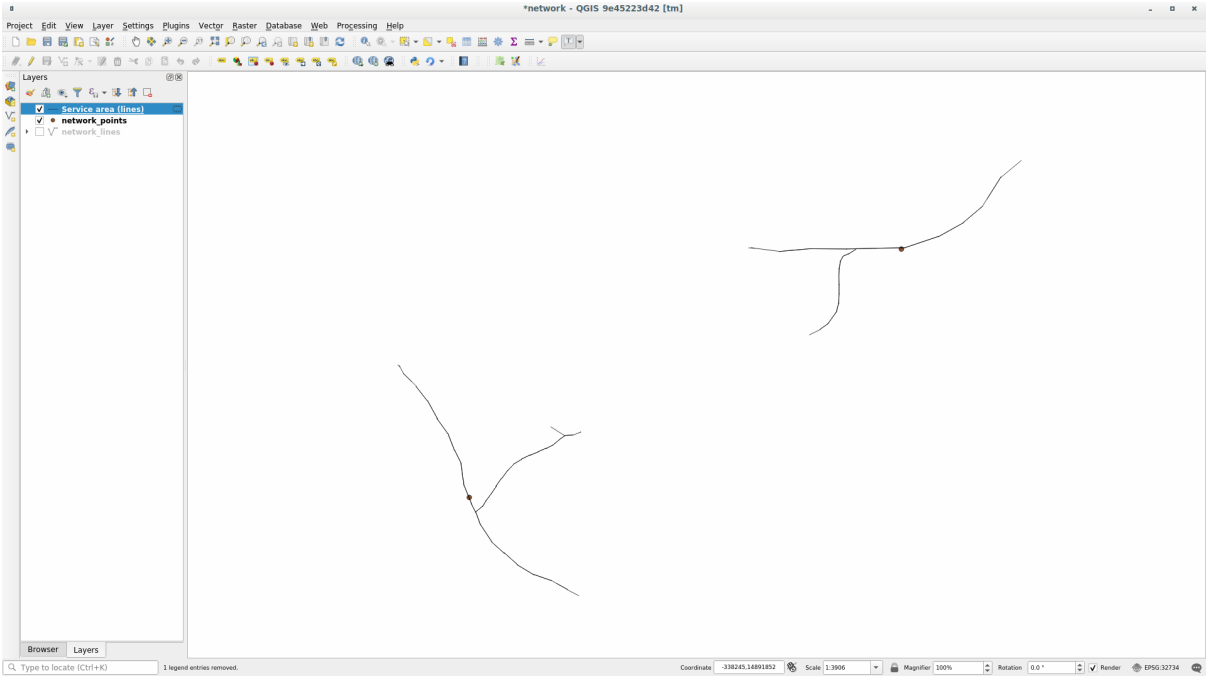
Bemerkung: The *Network Analysis* ► *Service area (from point)* is the same algorithm, but it allows you to manually choose the point on the map.

Given a distance of `250` meters we want to know how far we can go on the network from each point of the *network_points* layer.

1. Uncheck all the layers except `network_points`
2. Open the *Network Analysis* ► *Service area (from layer)* algorithm
3. Choose `network_lines` for *Vector layer representing network*
4. Choose `network_points` for *Vector layer with start points*
5. Choose `Shortest` in *Path type to calculate*
6. Enter `250` for the *Travel cost* parameter
7. Click on *Run* and close the dialog



The output layer represents the maximum path you can reach from the point features given a distance of 250 meters:



Cool isn't it?

6.3.7 In Conclusion

Now you know how to use *Network analysis* algorithm to solve shortest and fastest path problems.

We are now ready to perform some spatial statistic on vector layer data. Let's go!

6.3.8 What's Next?

Next you'll see how to run spatial statistics algorithms on vector datasets.

6.4 Lesson: Spatial Statistics

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

Spatial statistics allows you to analyze and understand what is going on in a given vector dataset. QGIS includes many useful tools for statistical analysis.

The goal for this lesson: To know how to use QGIS' spatial statistics tools within the *Processing Toolbox*.

6.4.1 Follow Along: Create a Test Dataset

We will create a random set of points, to get a dataset to work with.

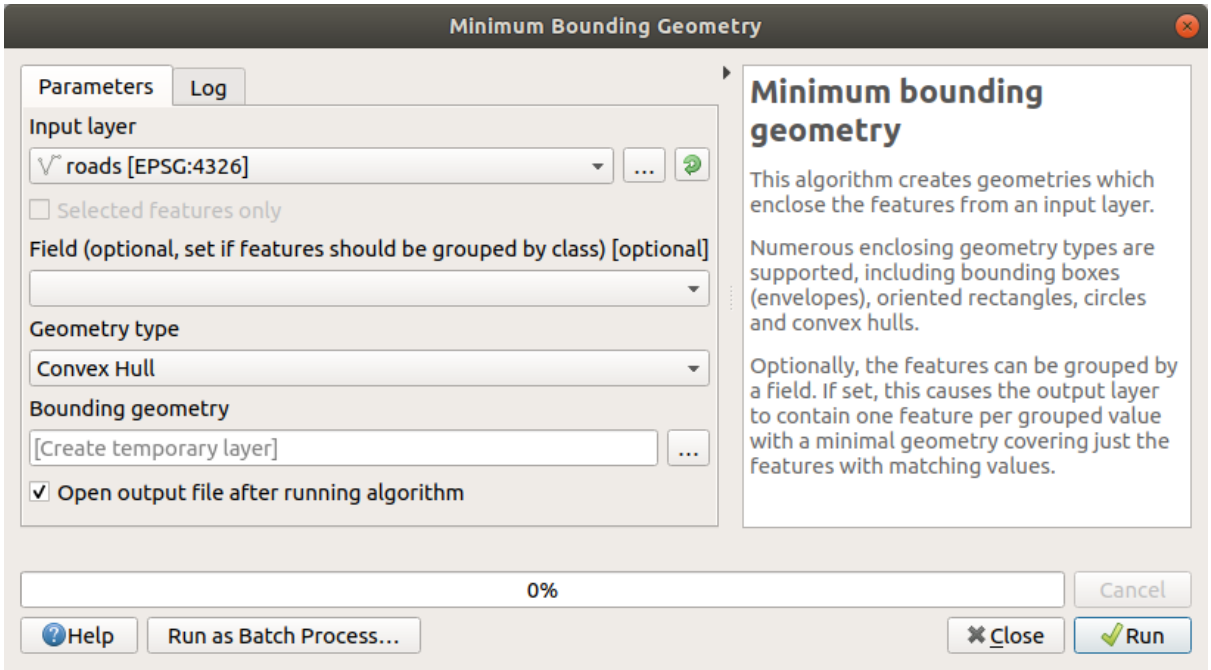
To do so, you will need a polygon dataset to define the area you want to create the points in.

We will use the area covered by streets.

1. Start a new project
2. Add your `roads` dataset, as well as `srtm_41_19` (elevation data) found in `exercise_data/raster/SRTM/`.

   Bemerkung: You might find that the SRTM DEM layer has a different CRS to that of the roads layer. QGIS is reprojecting both layers in a single CRS. For the following exercises this difference does not matter, but feel free to reproject (as shown earlier in this module).

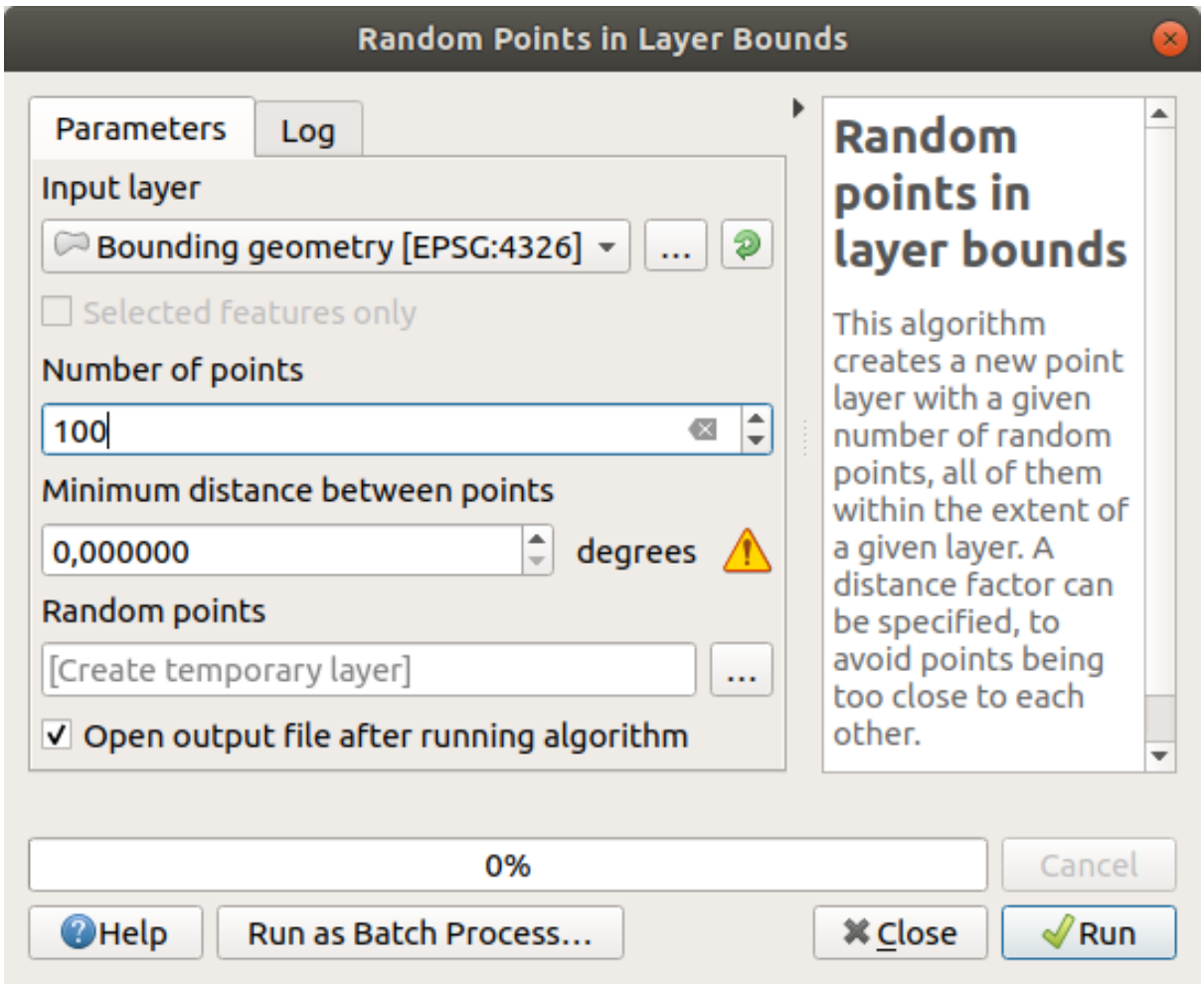
3. Open *Processing* toolbox
4. Use the *Vector Geometry* ► *Minimum bounding geometry* tool to generate an area enclosing all the roads by selecting `Convex Hull` as the *Geometry Type*:



As you know, if you don't specify the output, *Processing* creates temporary layers. It is up to you to save the layers immediately or at a later stage.

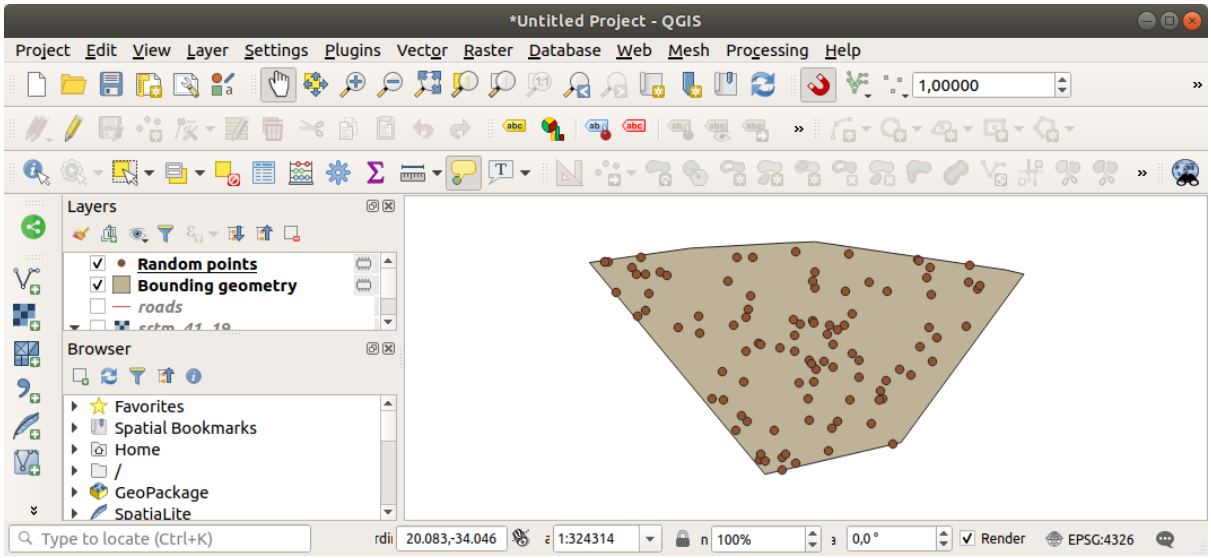
Creating random points

- Create 100 random points in this area using the tool at *Vector Creation* ► *Random points in layer bounds*, with a minimum distance of `0.0`:



> **Bemerkung:** The yellow warning sign tells you that that parameter concerns distances. The *Bounding geometry* layer is in a Geographical Coordinate System and the algorithm is just reminding you this. For this example we won't use this parameter so you can ignore it.

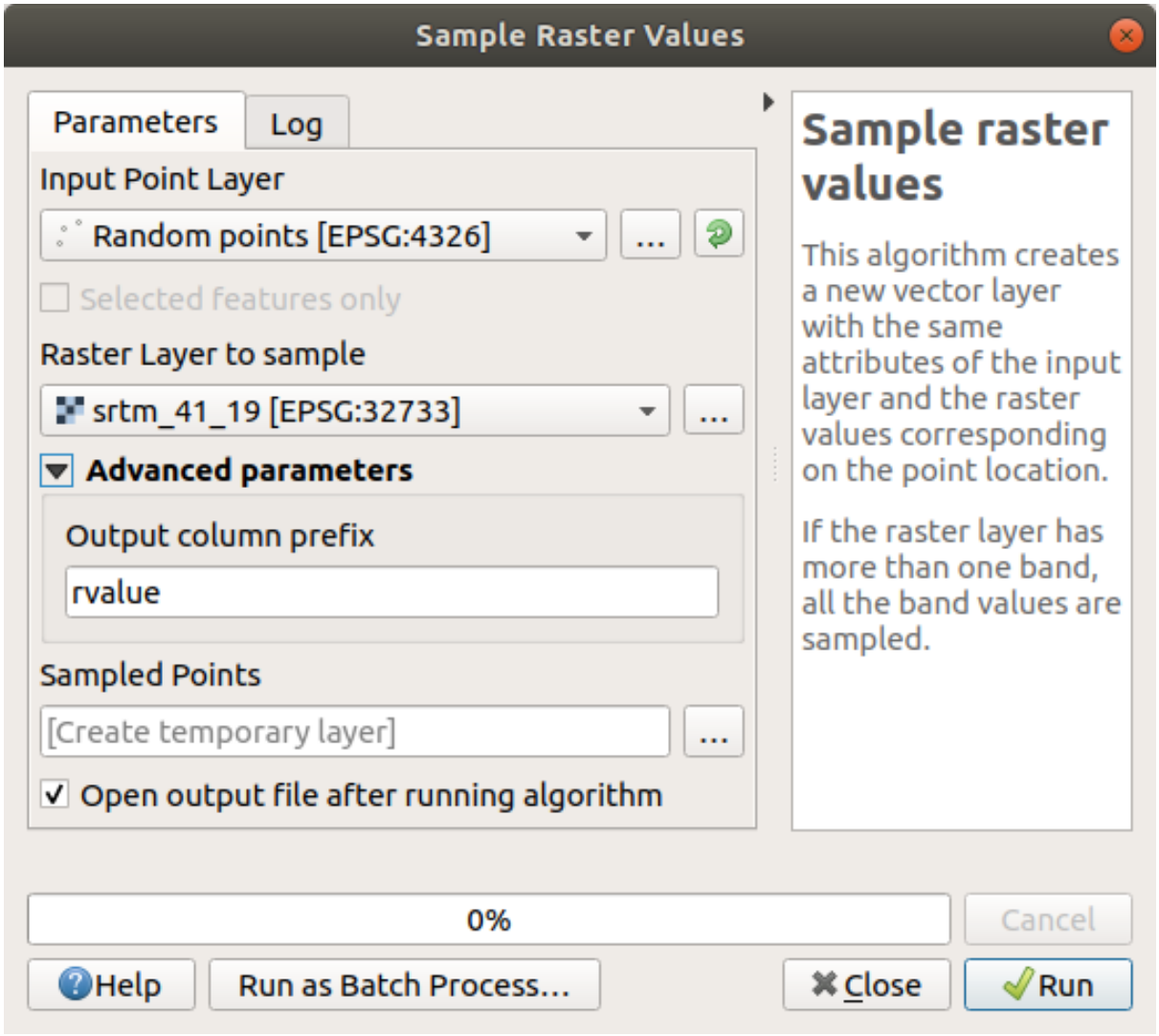
If needed, move the generated random point to the top of the legend to see them better:



Sampling the data

To create a sample dataset from the raster, you'll need to use the *Raster Analysis* ► *Sample raster values* algorithm. This tool samples the raster at the locations of the points and adds the raster values in new field(s) depending on the number of bands in the raster.

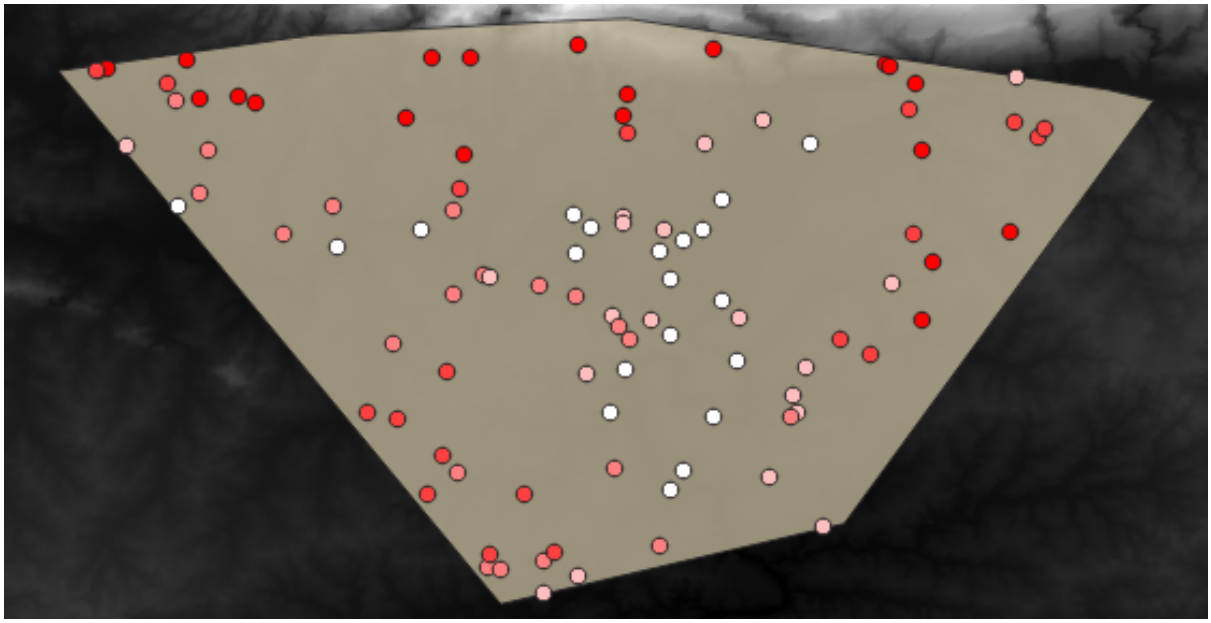
1. Open the *Sample raster values* algorithm dialog
2. Select `Random_points` as the layer containing sampling points, and the SRTM raster as the band to get values from. The default name of the new field is `rvalue_N`, where `N` is the number of the raster band. You can change the name of the prefix if you want.



3. Press *Run*

Now you can check the sampled data from the raster file in the attribute table of the `Sampled Points` layer. They will be in a new field with the name you have chosen.

A possible sample layer is shown here:



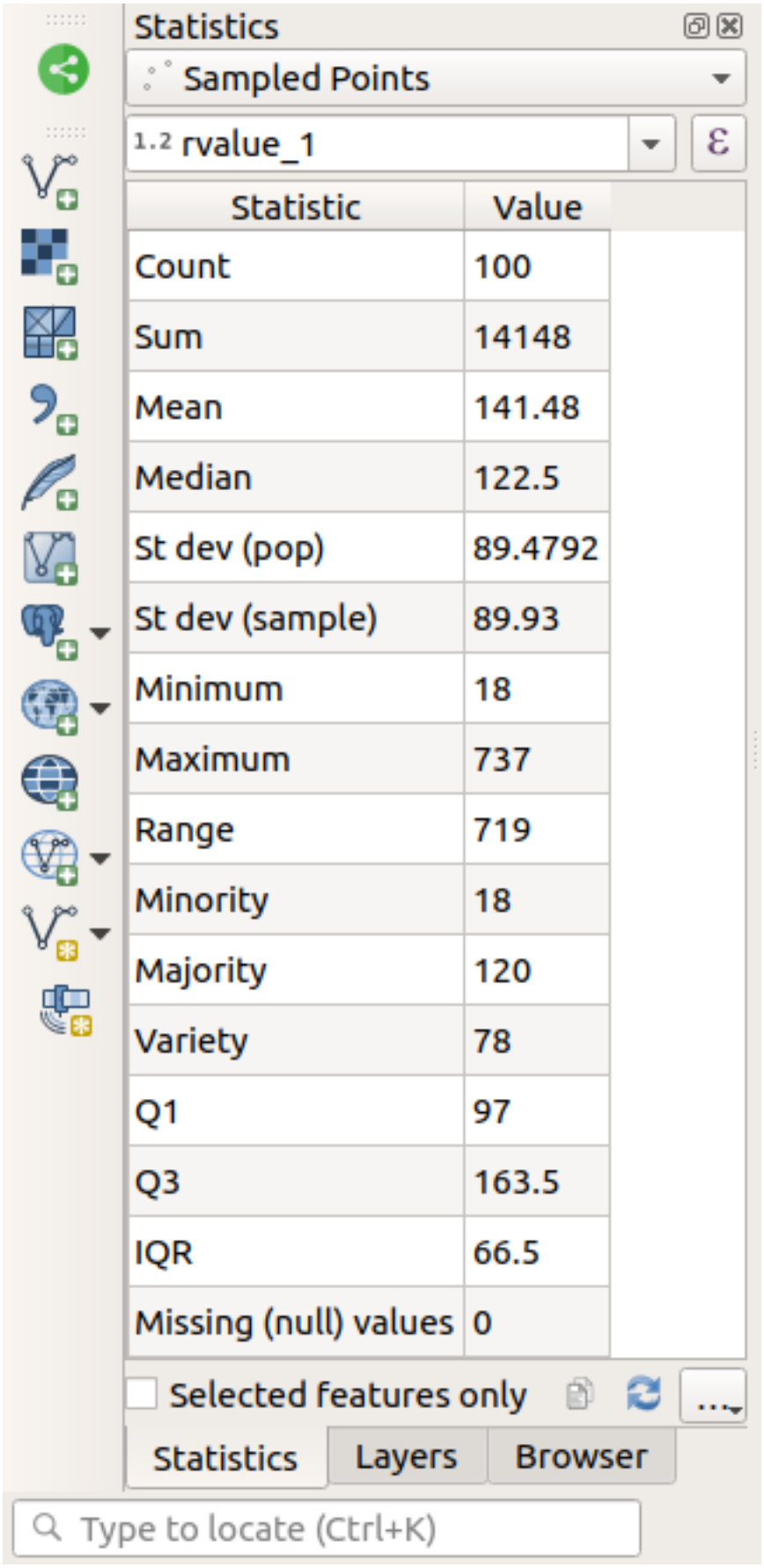
The sample points are classified using the `rvalue_1` field such that red points are at a higher altitude.

You will be using this sample layer for the rest of the statistical exercises.

6.4.2 Follow Along: Basic Statistics

Now get the basic statistics for this layer.

1. Click on the Show statistical summary icon in the *Attributes Toolbar*. A new panel will pop up.
2. In the dialog that appears, specify the `Sampled Points` layer as the source.
3. Select the *rvalue_1* field in the field combo box. This is the field you will calculate statistics for.
4. The *Statistics* Panel will be automatically updated with the calculated statistics:


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)

Bemerkung: You can copy the values by clicking on the Copy Statistics To Clipboard button and paste the results into a spreadsheet.

5. Close the *Statistics* Panel when done

Many different statistics are available:

Count
The number of samples/values.

Sum
The values added together.

Mean
The mean (average) value is simply the sum of the values divided by the number of values.

Median
If you arrange all the values from smallest to greatest, the middle value (or the average of the two middle values, if N is an even number) is the median of the values.

St Dev (pop)
The standard deviation. Gives an indication of how closely the values are clustered around the mean. The smaller the standard deviation, the closer values tend to be to the mean.

Minimum
The minimum value.

Maximum
The maximum value.

Bereich
The difference between the minimum and maximum values.

Q1
First quartile of the data.

Q3
Third quartile of the data.

Missing (null) values
The number of missing values.

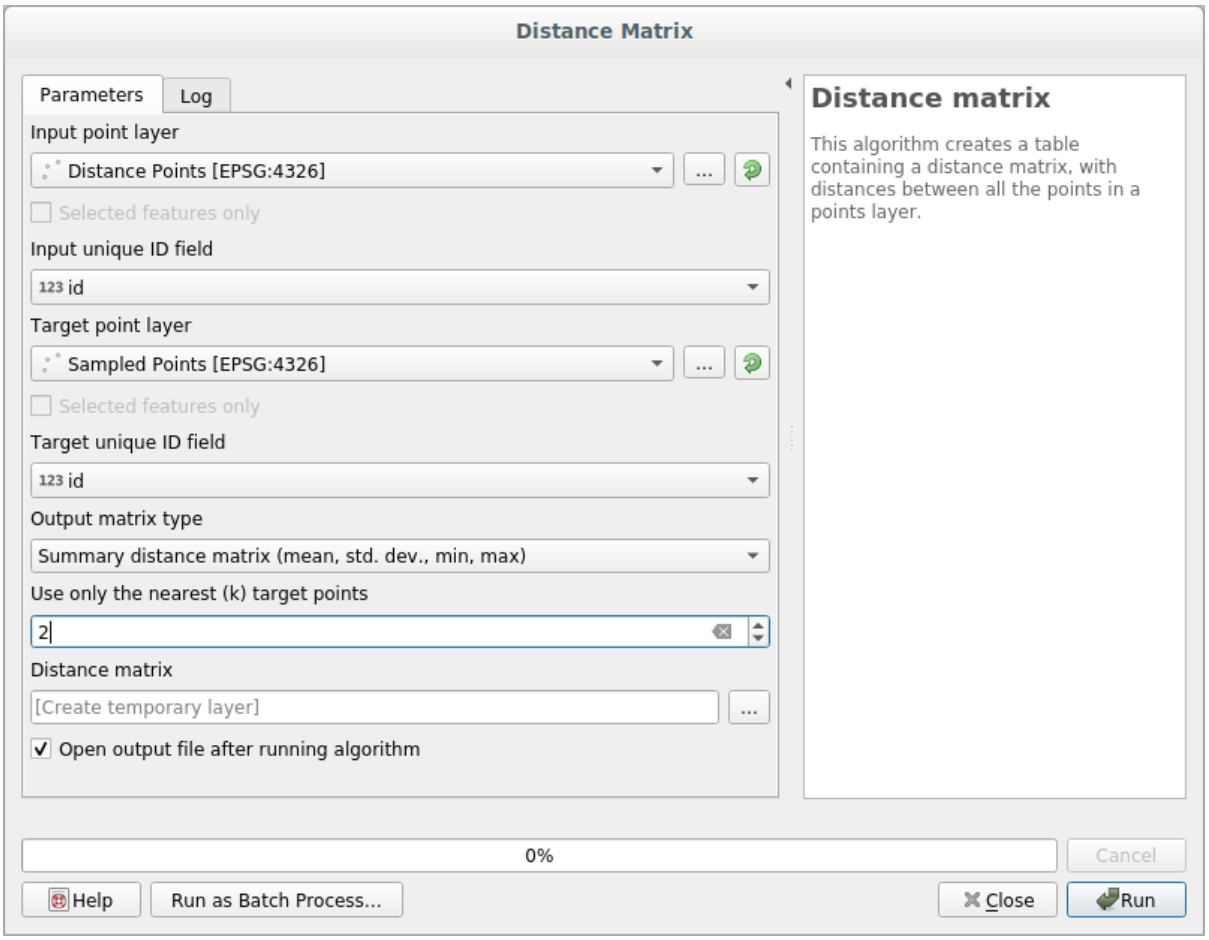
6.4.3 Follow Along: Compute statistics on distances between points

1. Create a new temporary point layer.
2. Enter edit mode, and digitize three points somewhere among the other points.

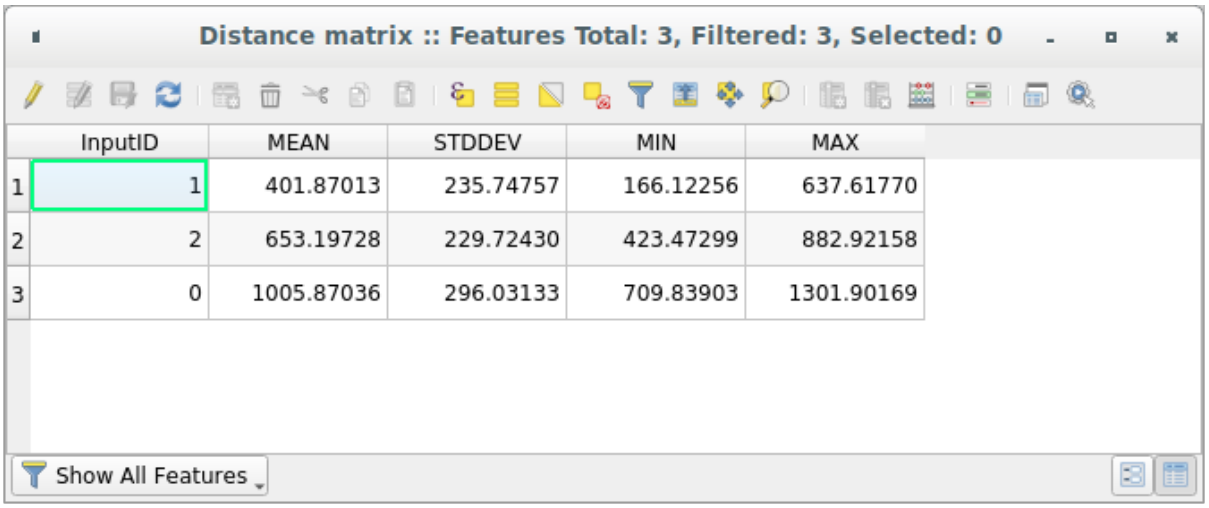
   Alternatively, use the same random point generation method as before, but specify only **three** points.
3. Save your new layer as *distance_points* in the format you prefer.

To generate statistics on the distances between points in the two layers:

1. Open the *Vector Analysis* ► *Distance matrix* tool.
2. Select the `distance_points` layer as the input layer, and the `Sampled Points` layer as the target layer.
3. Set it up like this:



4. If you want you can save the output layer as a file or just run the algorithm and save the temporary output layer later.
5. Click *Run* to generate the distance matrix layer.
6. Open the attribute table of the generated layer: values refer to the distances between the *distance_points* features and their two nearest points in the *Sampled Points* layer:

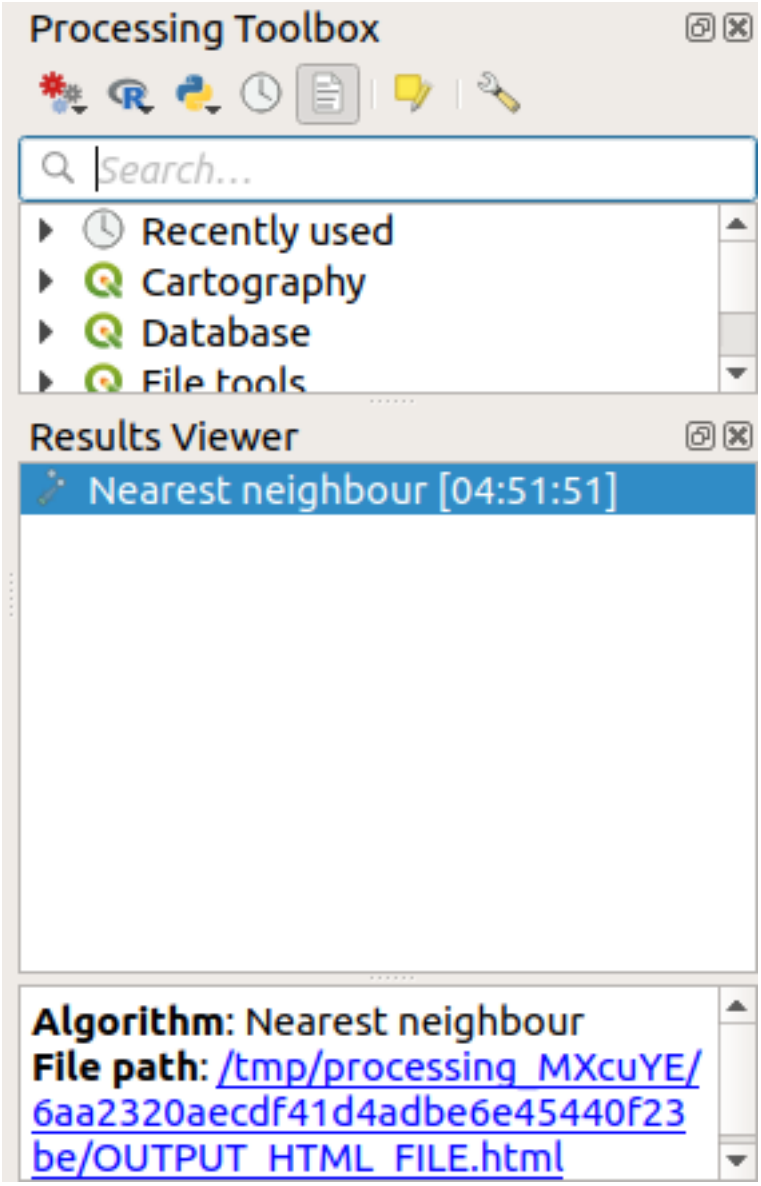


With these parameters, the *Distance Matrix* tool calculates distance statistics for each point of the input layer with respect to the nearest points of the target layer. The fields of the output layer contain the mean, standard deviation, minimum and maximum for the distances to the nearest neighbors of the points in the input layer.

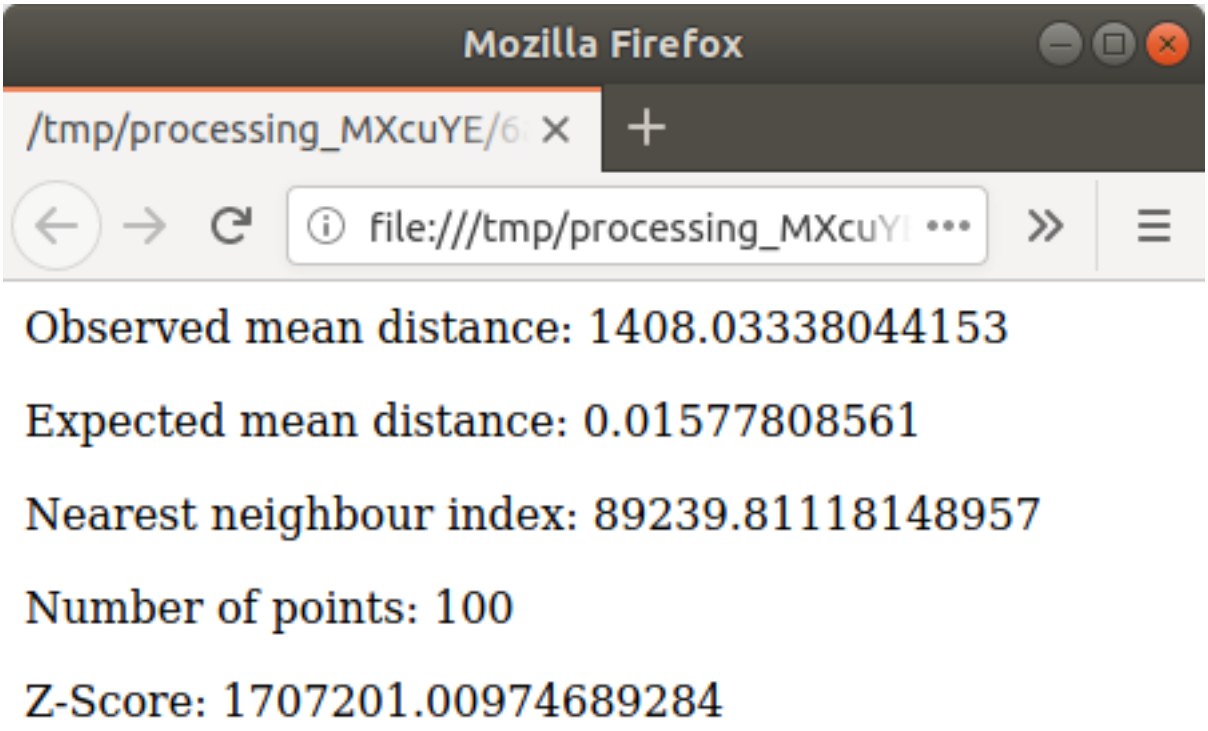
6.4.4 Follow Along: Nearest Neighbor Analysis (within layer)

To do a nearest neighbor analysis of a point layer:

1. Choose *Vector analysis* ► *Nearest neighbor analysis*.
2. In the dialog that appears, select the *Random points* layer and click *Run*.
3. The results will appear in the Processing *Result Viewer* Panel.



4. Click on the blue link to open the `html` page with the results:



6.4.5 Follow Along: Mean Coordinates

To get the mean coordinates of a dataset:

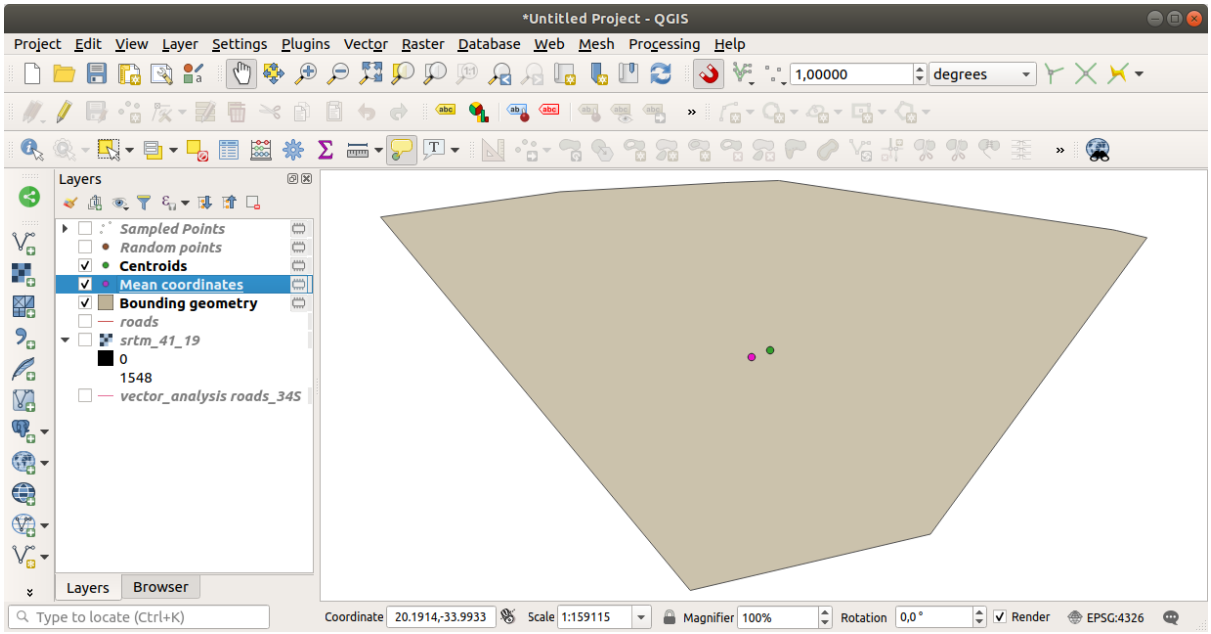
1. Start *Vector analysis* ► *Mean coordinate(s)*
2. In the dialog that appears, specify *Random points* as *Input layer*, and leave the optional choices unchanged.
3. Klicken Sie auf *Starten*.

Let us compare this to the central coordinate of the polygon that was used to create the random sample.

1. Start *Vector geometry* ► *Centroids*
2. In the dialog that appears, select `Bounding geometry` as the input layer.

As you can see, the mean coordinates (pink point) and the center of the study area (in green) don't necessarily coincide.

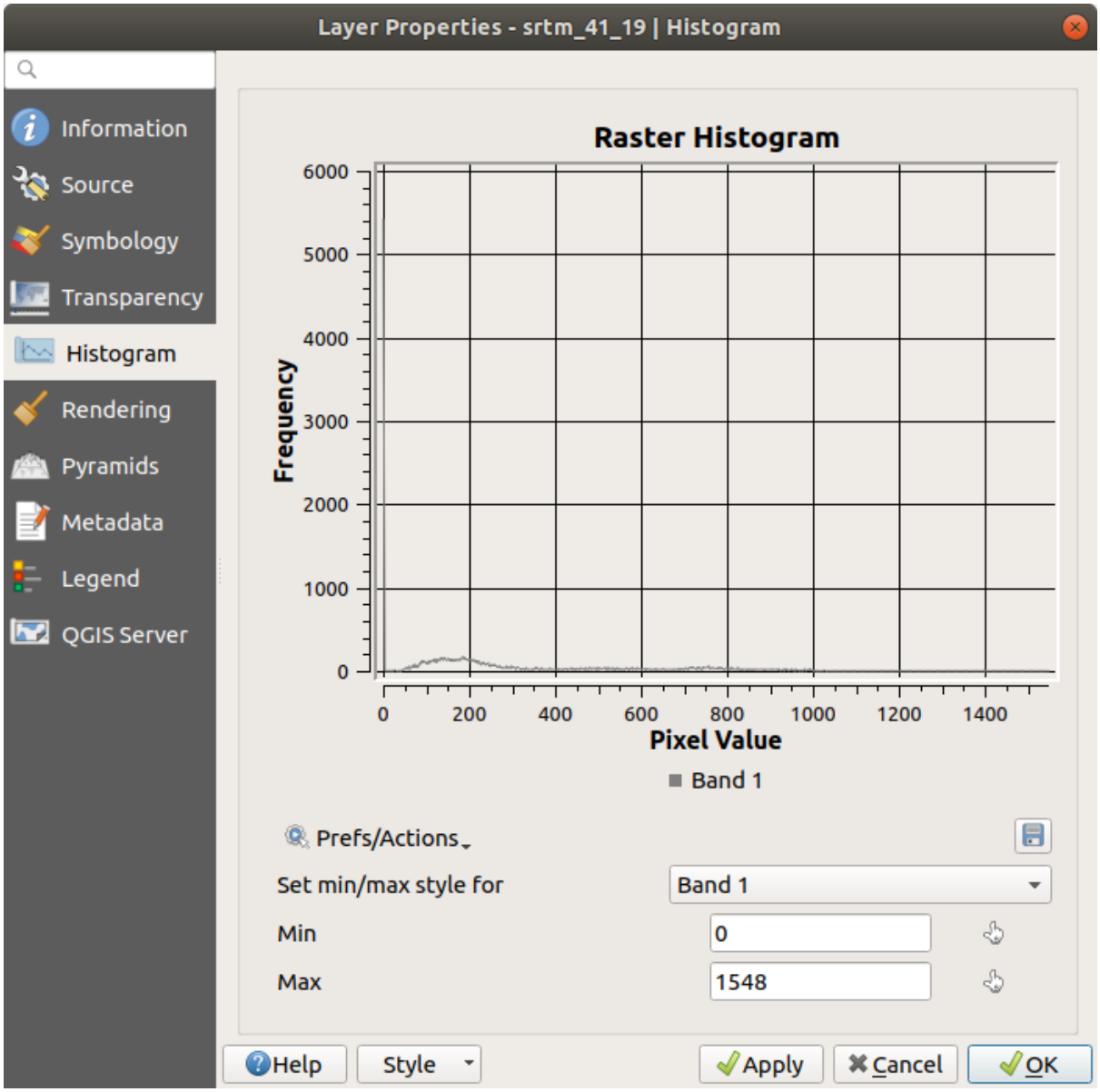
The centroid is the barycenter of the layer (the barycenter of a square is the center of the square) while the mean coordinates represent the average of all node coordinates.



6.4.6 Follow Along: Image Histograms

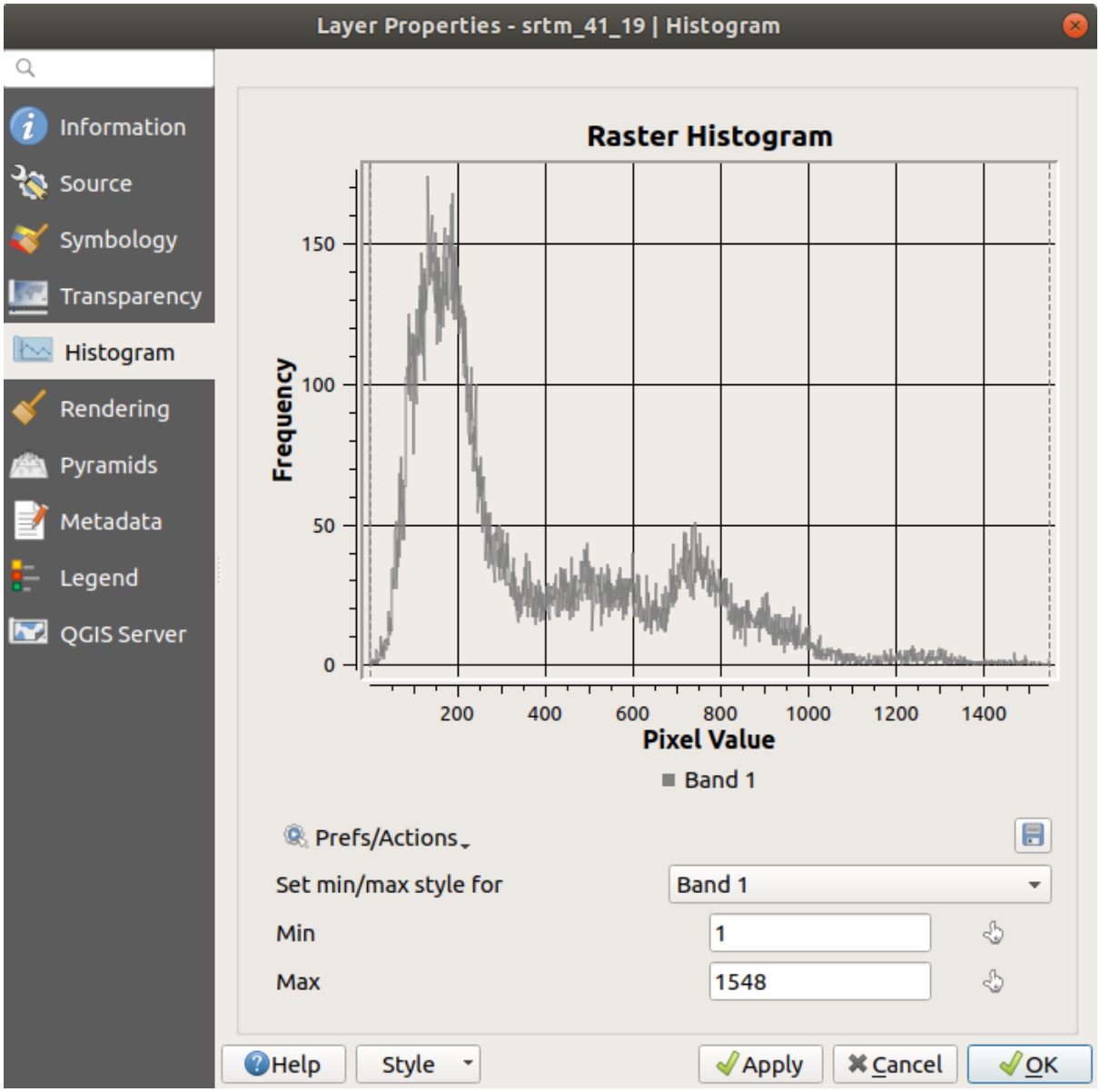
The histogram of a dataset shows the distribution of its values. The simplest way to demonstrate this in QGIS is via the image histogram, available in the *Layer Properties* dialog of any image layer (raster dataset).

1. In your *Layers* panel, right-click on the `srtm_41_19` layer
2. Select *Properties*
3. Choose the *Histogram* tab. You may need to click on the *Compute Histogram* button to generate the graphic. You will see a graph that shows the frequency distribution for the raster values.



4. The graph can be exported as an image with the Save plot button
5. You can see more detailed information about the layer in the *Information* tab (the mean and max values are estimated, and may not be exact).

The mean value is `332.8` (estimated to 324.3), and the maximum value is `1699` (estimated to 1548)! You can zoom in the histogram. Since there are a lot of pixels with value `0`, the histogram looks compressed vertically. By zooming in to cover everything but the peak at `0`, you will see more details:



Bemerkung: If the mean and maximum values are not the same as above, it can be due to the min/max value calculation. Open the *Symbology* tab and expand the *Min / Max Value Settings* menu. Choose *Min / max* and click on *Apply*.

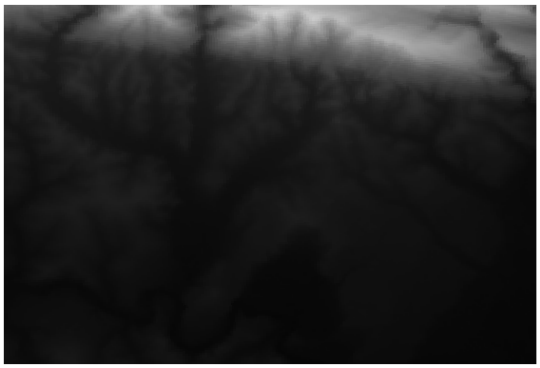
Keep in mind that a histogram shows you the distribution of values, and not all values are necessarily visible on the graph.

6.4.7 Follow Along: Spatial Interpolation

Let's say you have a collection of sample points from which you would like to extrapolate data. For example, you might have access to the *Sampled points* dataset we created earlier, and would like to have some idea of what the terrain looks like.

1. To start, launch the *GDAL* ► *Raster analysis* ► *Grid (IDW with nearest neighbor searching)* tool in the *Processing Toolbox*.
2. For *Point layer* select `Sampled points`
3. Set *Weighting power* to `5.0`
4. In *Advanced parameters*, set *Z value from field* to `rvalue_1`
5. Finally click on *Run* and wait until the processing ends
6. Close the dialog

Here is a comparison of the original dataset (left) to the one constructed from our sample points (right). Yours may look different due to the random nature of the location of the sample points.



As you can see, 100 sample points aren't really enough to get a detailed impression of the terrain. It gives a very general idea, but it can be misleading as well.

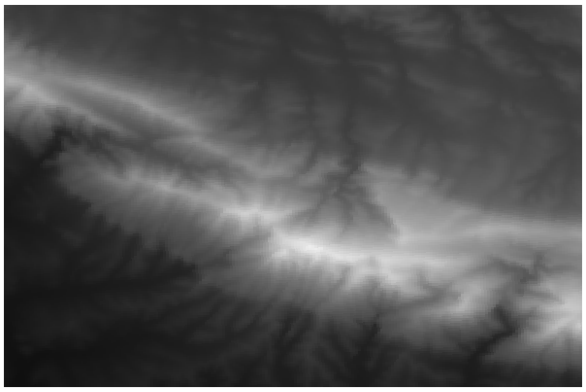
6.4.8 Try Yourself Different interpolation methods

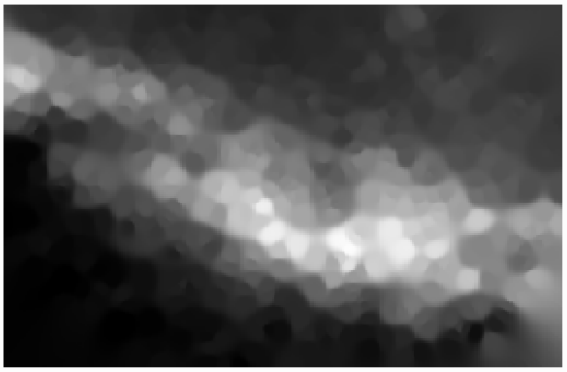
1. Use the processes shown above to create a set of 10 000 random points

   Bemerkung: If the number of points is really big, the processing time can take a long time.

2. Use these points to sample the original DEM
3. Use the *Grid (IDW with nearest neighbor searching)* tool on this dataset.
4. Set *Power* and *Smoothing* to `5.0` and `2.0`, respectively.

The results (depending on the positioning of your random points) will look more or less like this:





This is a better representation of the terrain, due to the greater density of sample points. Remember, larger samples give better results.

6.4.9 In Conclusion

QGIS has a number of tools for analyzing the spatial statistical properties of datasets.

6.4.10 What's Next?

Now that we have covered vector analysis, why not see what can be done with rasters? That is what we will do in the next module!

KAPITEL 7

Module: Raster

Wir haben schon früher Raster für die Digitalisierung verwendet, aber Rasterdaten können auch direkt verwendet werden. In diesem Modul sehen Sie, wie es in QGIS gemacht wird.

7.1 Lesson: Arbeiten mit Rasterdaten

Rasterdaten unterscheiden sich deutlich von Vektordaten. Vektordaten haben individuelle Eigenschaften und bestehen aus geometrischen Formen, die aus verbundenen Punkten zusammengesetzt sind. Ggf. sind sie mit anderen Linien und/oder Flächen verbunden sind. Rasterdaten hingegen sind mit einem Bild vergleichbar. Obwohl Rasterdaten verschiedene Eigenschaften von Objekten in der realen Welt abbilden können, existieren sie nicht als separate Objekte. Stattdessen werden sie durch Pixel mit unterschiedlichen Werten dargestellt.

In diesem Modul werden Sie Rasterdaten verwenden, um Ihre bestehenden GIS-Analysen zu erweitern.

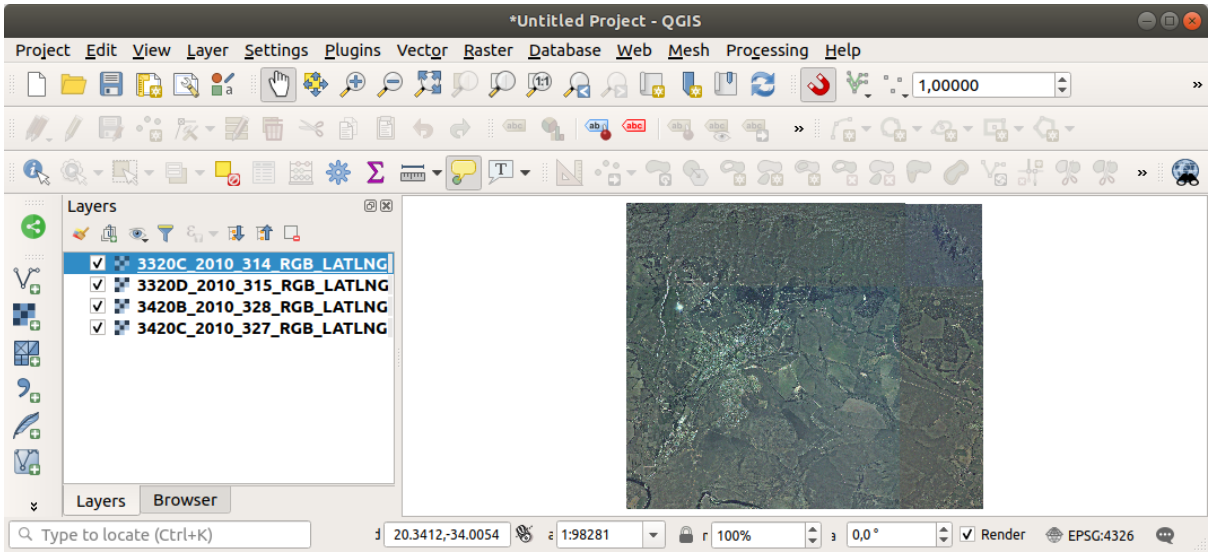
Das Ziel dieser Lektion: Sie lernen, wie Sie mit Rasterdaten in QGIS arbeiten.

7.1.1 Follow Along: Rasterdaten laden

Raster data can be loaded with the same methods we used for vector data. However we suggest to use the *Browser* Panel.

1. Open the *Browser* Panel and expand the `exercise_data/raster` folder.
2. Load all the data in this folder:
    - `3320C_2010_314_RGB_LATLNG.tif`
    - `3320D_2010_315_RGB_LATLNG.tif`
    - `3420B_2010_328_RGB_LATLNG.tif`
    - `3420C_2010_327_RGB_LATLNG.tif`

You should see the following map:



There we have it - four aerial images covering our study area.

7.1.2 Follow Along: Create a Virtual Raster

Now as you can see from this, your solution layer lies across all four images. What this means is that you are going to have to work with four rasters all the time. That's not ideal. It would be better to have one file to work with.

Luckily, QGIS allows you to do exactly this, and without needing to actually create a new raster file. You can create a **Virtual Raster**. This is also often called a *Catalog*, which explains its function. It's not really a new raster. Rather, it is a way to organize your existing rasters into one catalog: one file for easy access.

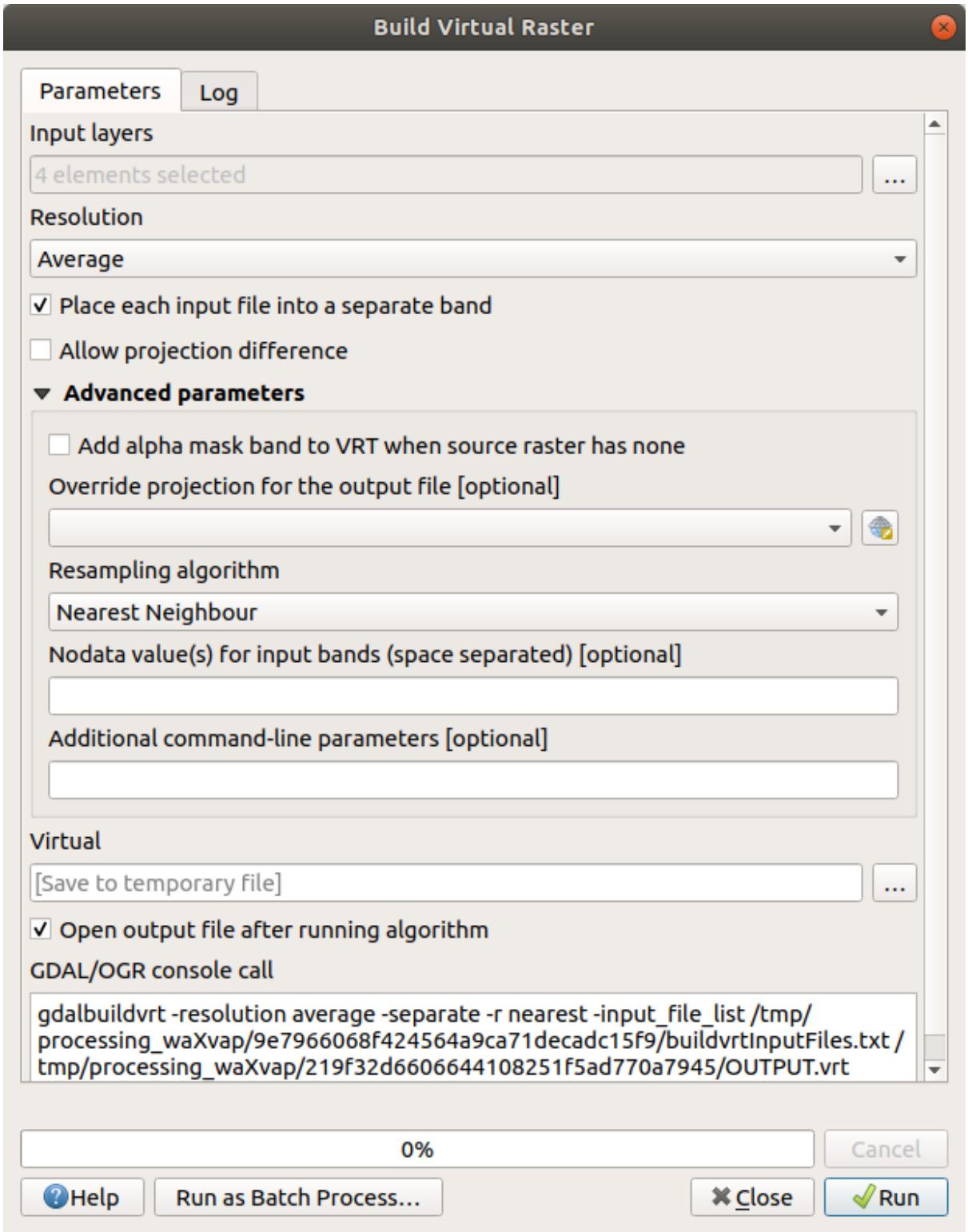
To make a catalog we will use the *Processing* ► *Toolbox*.

1. Open the *Build virtual raster* algorithm from the *GDAL* ► *Raster miscellaneous*;
2. In the dialog that appears, click on the … button next to the *Input layers* parameter and check all the layers or use the *Select All* button;
3. Uncheck *Place each input file into a separate band*. Notice the text field below. What this dialog is actually doing is that it is writing that text for you. It is a long command that QGIS is going to run.

   Bemerkung: Keep in mind that you can copy and paste the text in the `OSGeo Shell` (Windows user) or `Terminal` (Linux and OSX users) to run the command. You can also create a script for each GDAL command. This is very handy when the procedure is taking a long time or when you want to schedule specific tasks. Use the *Help* button to get more help on the syntax of the command.

4. Finally click on *Run*.

Bemerkung: As you know from the previous modules, *Processing* creates temporary layers by default. To save the file click on the … button.



You can now remove the original four rasters from the *Layers* Panel and leave only the output virtual catalog raster.

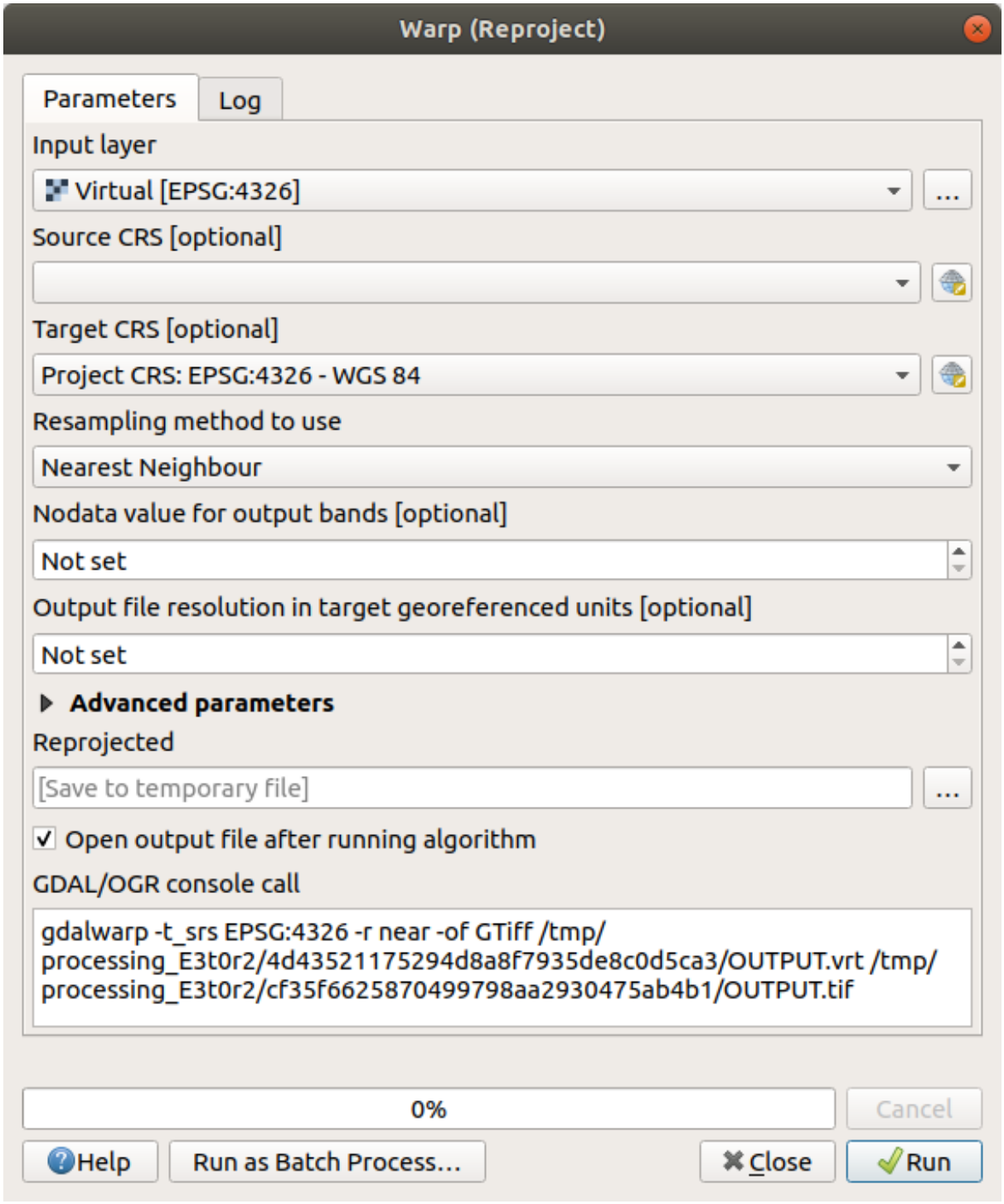
7.1.3 Transforming Raster Data

The above methods allow you to virtually merge datasets using a catalog, and to reproject them „on the fly“. However, if you are setting up data that you'll be using for quite a while, it may be more efficient to create new rasters that are already merged and reprojected. This improves performance while using the rasters in a map, but it may take some time to set up initially.

Reprojecting rasters

Open *Warp (reproject)* from *GDAL* ► *Raster projections*.

You can also reproject virtual rasters (catalogs), enable multithreaded processing, and more.



Merging rasters

If you need to create a new raster layer and save it to disk you can use the merge algorithm.

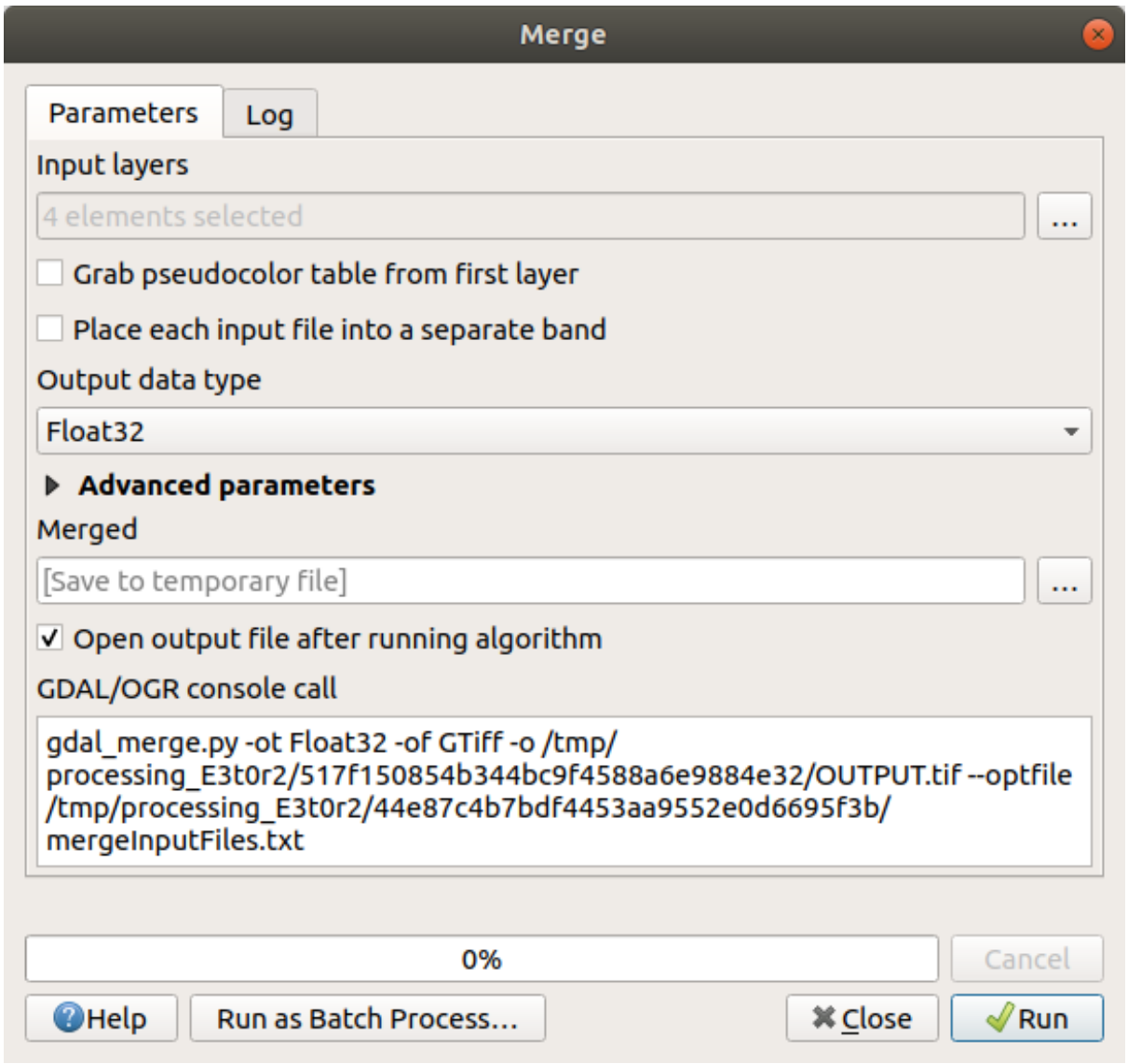
Bemerkung: Depending on how many raster files you are merging and their resolution, the new raster file created can be really big. Consider instead to create a raster catalog as described in the *Create a Virtual Raster* section.

---

1. Click on the *Merge* algorithm from the *GDAL* ► *Raster miscellaneous* menu.
2. As we did for the *Create a Virtual raster*, use the … button to choose which layers you want to merge.

You can also specify a Virtual raster as input, and then all of the rasters that it consists of will be processed.

3. If you know the GDAL library, you can also add your own options by opening the *Advanced parameters* menu.



7.1.4 In Conclusion

QGIS makes it easy to include raster data into your existing projects.

7.1.5 What's Next?

Next, we'll use raster data that isn't aerial imagery, and see how symbolization is useful in the case of rasters as well.

7.2 Lesson: Changing Raster Symbology

Not all raster data are aerial photos. There are many other forms of raster data, and in many of those cases, it is essential to symbolize the them so that they becomes properly visible and useful.

The goal for this lesson: To change the symbology for a raster layer.

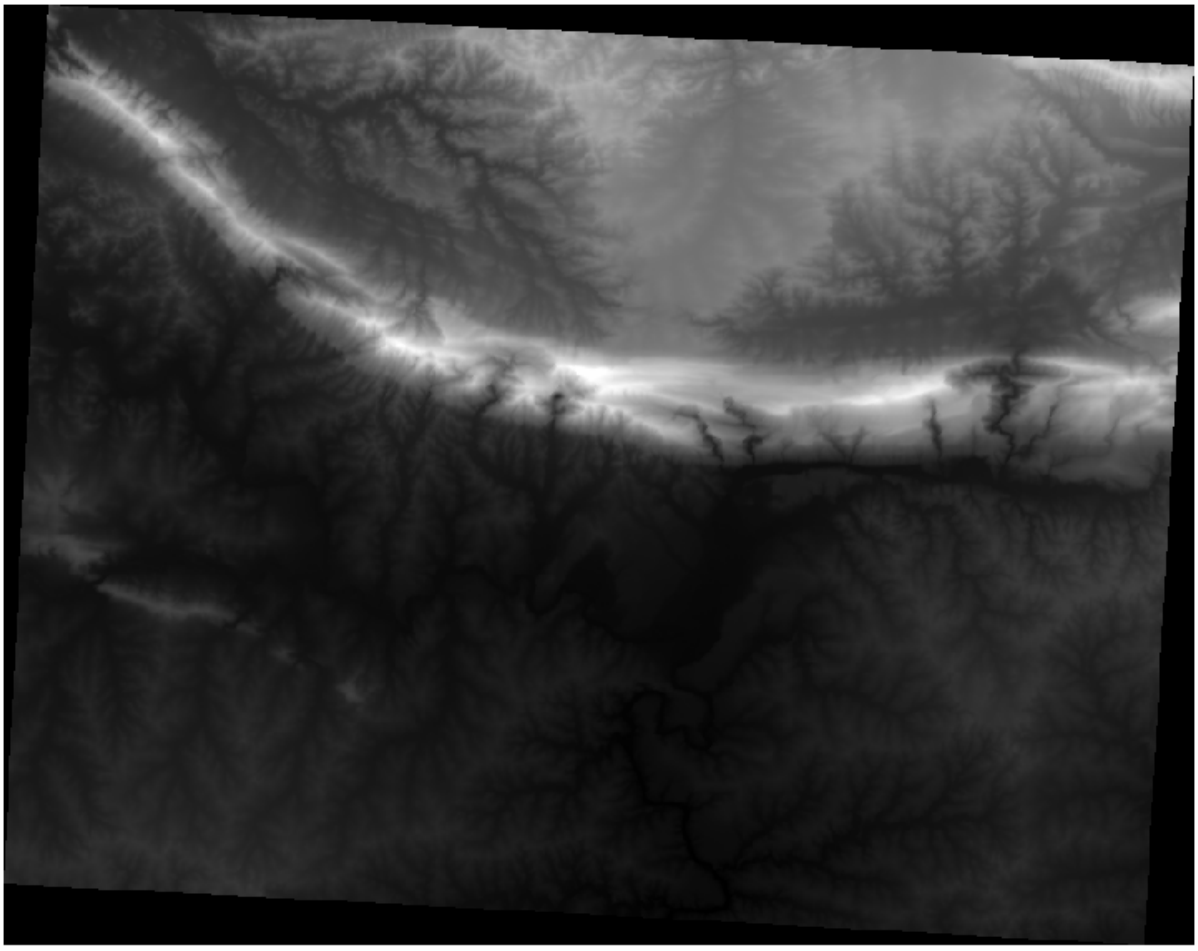
7.2.1 Try Yourself

1. Use the *Browser* Panel to load `srtm_41_19.tif`, found under `exercise_data/raster/SRTM/`
2. Zoom to the extent of this layer by right-clicking on it in the *Layers panel* and selecting *Zoom to Layer*.

This dataset is a *Digital Elevation Model (DEM)*. It is a map of the elevation (altitude) of the terrain, allowing us to see where the mountains and valleys are, for example.

While each pixel of the dataset of the previous section contained color information, in a *DEM*, each pixel contains elevation values.

Once the DEM is loaded, you will notice that it is a grayscale representation:



QGIS has automatically applied a stretch to the pixel values of the image for visualization purposes, and we will learn more about how this works as we continue.

7.2.2 Follow Along: Changing Raster Layer Symbology

You have two different options to change the raster symbology:

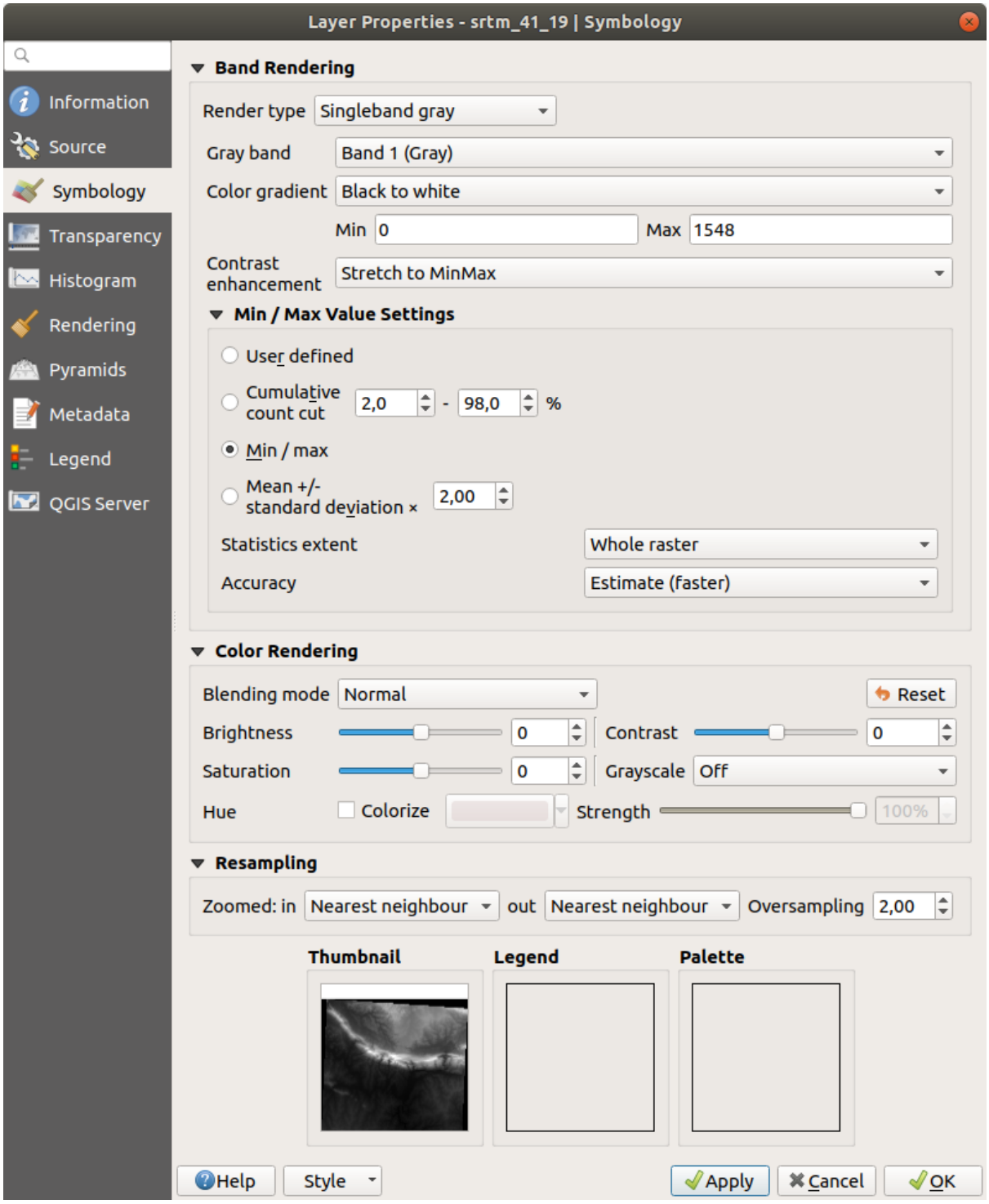
1. Within the *Layer Properties* dialog, by right-clicking on the layer in the Layer tree and selecting the *Properties* option. Then switch to the *Symbology* tab
2. By clicking on the Open the Layer Styling panel button right above the *Layers* panel (shortcut `F7`). This will open the *Layer Styling* panel, where you can switch to the Symbology tab.

Choose the method you prefer to work with.

7.2.3 Follow Along: Singleband gray

When you load a raster file, if it is not a photo image like the ones of the previous section, the default style is set to a grayscale gradient.

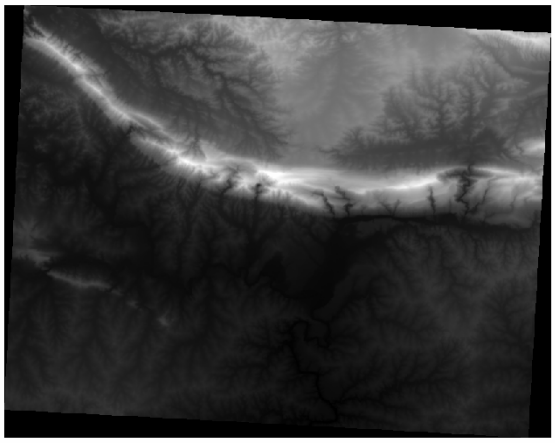
Let's explore some of the features of this renderer.



The default *Color gradient* is set to `Black to white`, meaning that low pixel values are black and while high values are white. Try to invert this setting to `White to black` and see the results.

Very important is the *Contrast enhancement* parameter: by default it is set to `Stretch to MinMax` meaning that the pixel values are stretched to the minimum and maximum values.

Look at the difference with the enhancement (left) and without (right):



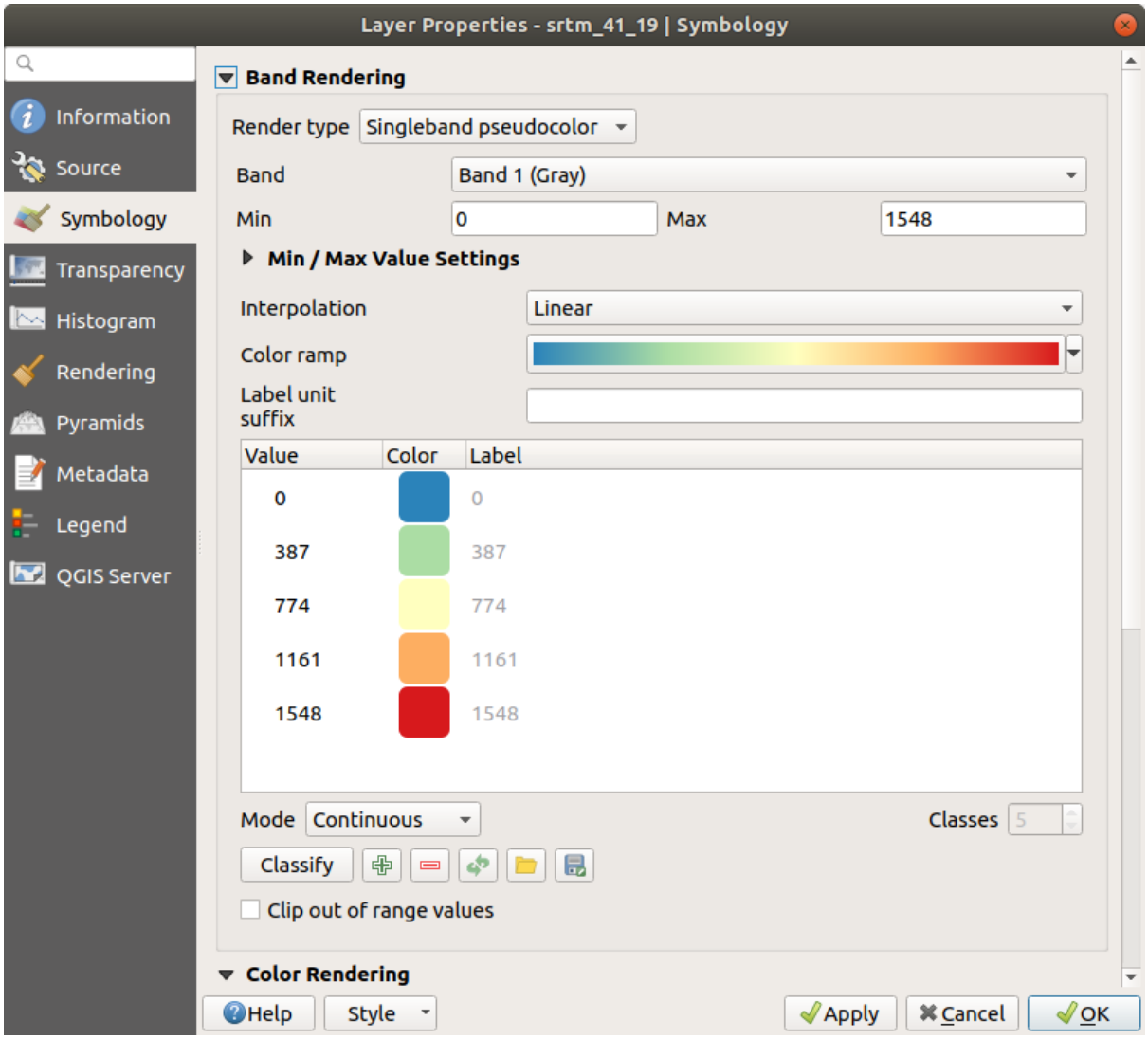
But what are the minimum and maximum values that should be used for the stretch? The ones that are currently under *Min / Max Value Settings*. There are many ways to calculate the minimum and maximum values and use them for the stretch:

1. **User Defined**: you enter the *Min* and *Max* values manually
2. **Cumulative count cut**: this is useful when you have some extreme low or high values. It *cuts* the `2%` (or the value you choose) of these values
3. **Min / max**: the *Real* or *Estimated* minimum and maximum values of the raster
4. **Mean +/- standard deviation**: the values will be calculated according to the mean value and the standard deviation

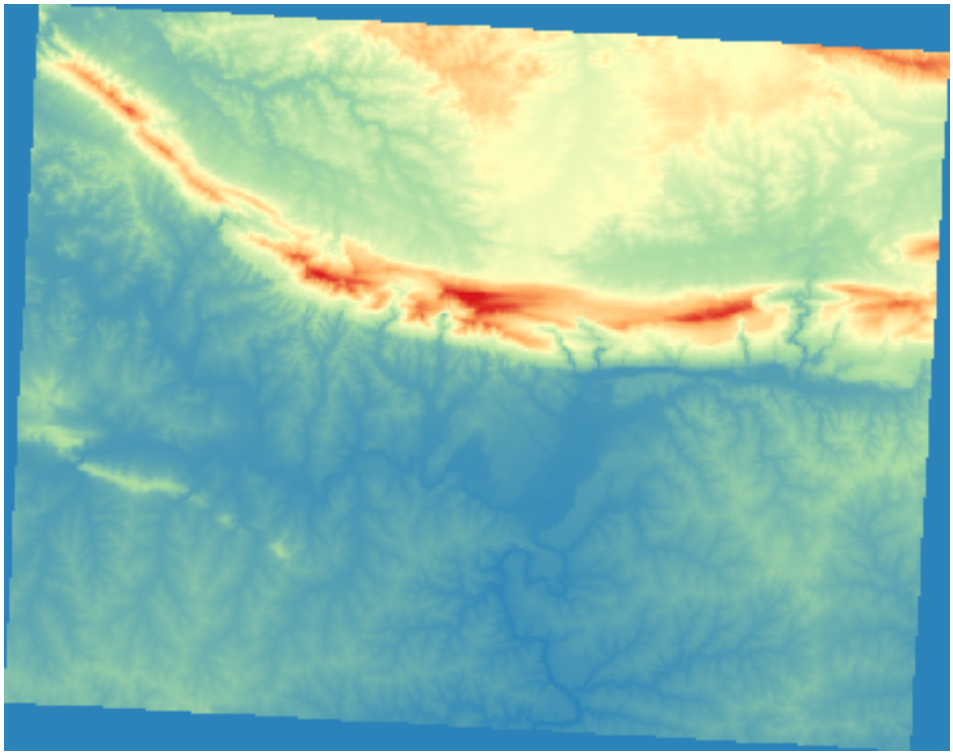
7.2.4 Follow Along: Singleband pseudocolor

Grayscales are not always great styles for raster layers. Let's try to make the DEM more colorful.

- Change the *Render type* to *Singleband pseudocolor*. If you don't like the default colors loaded, select another *Color ramp*
- Click the *Classify* button to generate a new color classification
- If it is not generated automatically click on the *OK* button to apply this classification to the DEM



You'll see the raster looking like this:

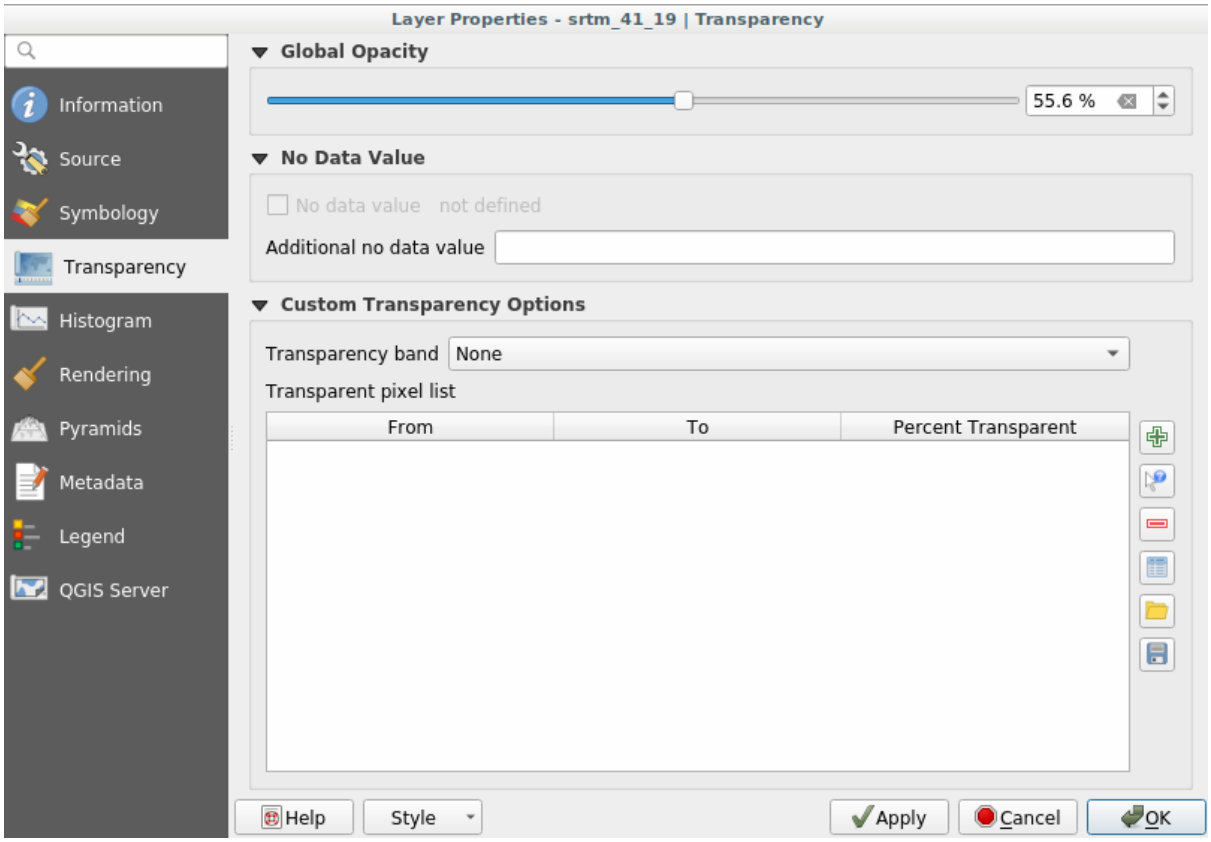


This is an interesting way of looking at the DEM. You will now see that the values of the raster are again properly displayed, going from blue for the lower areas to red for the higher ones.

7.2.5 Follow Along: Changing the transparency

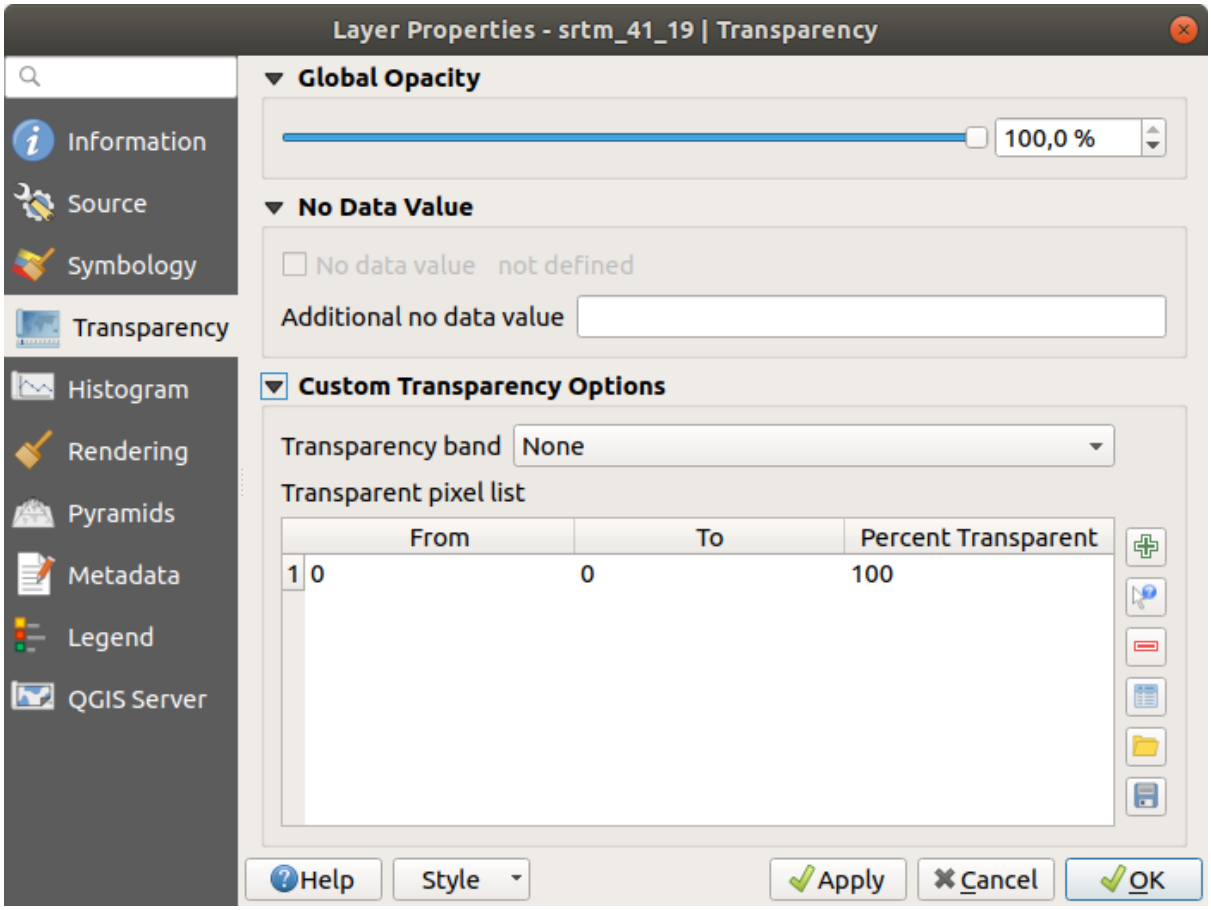
Sometimes changing the transparency of the whole raster layer can help you to see other layers covered by the raster itself and better understand the study area.

To change the transparency of the whole raster switch to the *Transparency* tab and use the slider of the *Global Opacity* to lower the opacity:

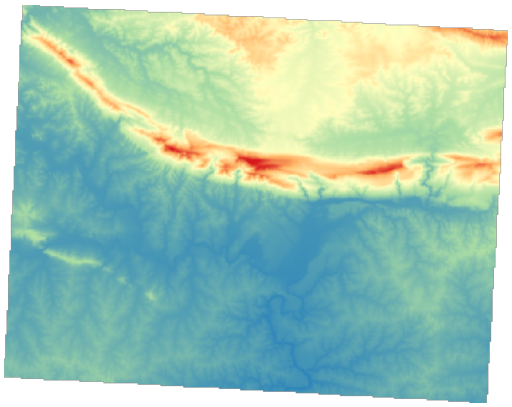


More interesting is changing the transparency for some pixel values. For example in the raster we used you can see a homogeneous color at the corners. To set these pixels as transparent, go to *Custom Transparency Options* in the *Transparency* tab.

- By clicking on the Add values manually button, you can add a range of values and set their transparency percentage
- For single values the Add values from display button is more useful
- Click on the Add values from display button. The dialog disappears, and you can interact with the map.
- Click on the homogeneous color in a corner of the DEM
- You will see that the transparency table will be filled with the clicked values:



- Click on *OK* to close the dialog and see the changes.



See? The corners are now 100% transparent.

7.2.6 In Conclusion

These are some the basic functions to get you started with raster symbology. QGIS also gives you many other options, such as symbolizing a layer using paletted/unique values, representing different bands with different colors in a multispectral image, or making an automatic hillshade effect (useful only with DEM raster files).

7.2.7 Referenz

The SRTM dataset was obtained from http://srtm.csi.cgiar.org/

7.2.8 What's Next?

Now that we can see our data displayed properly, let's investigate how we can analyze it further.

7.3 Lesson: Terrain Analysis

Certain types of rasters allow you to gain more insight into the terrain that they represent. Digital Elevation Models (DEMs) are particularly useful in this regard. In this lesson you will use terrain analysis tools to find out more about the study area for the proposed residential development from earlier.

The goal for this lesson: To use terrain analysis tools to derive more information about the terrain.

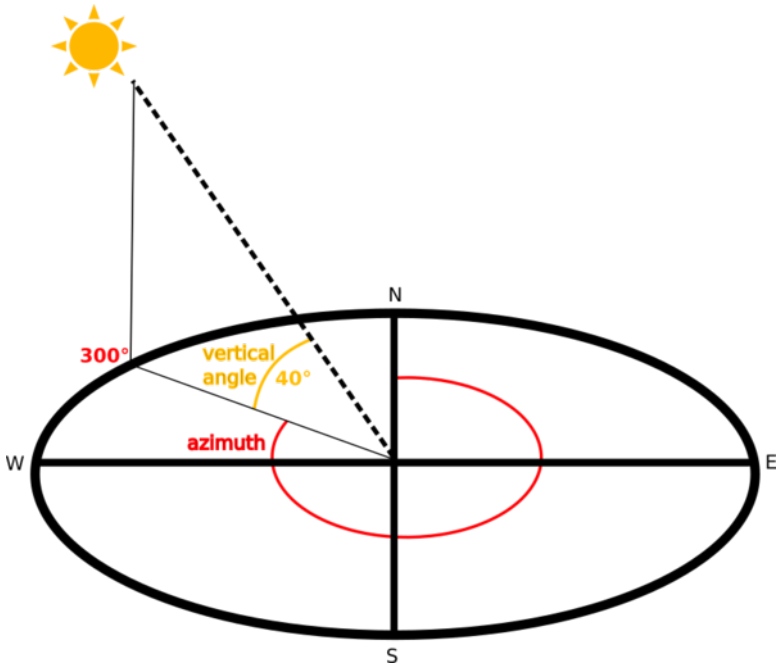
7.3.1 Follow Along: Calculating a Hillshade

We are going to use the same DEM layer as in the previous lesson. If you are starting this chapter from scratch, use the *Browser* panel and load the `raster/SRTM/srtm_41_19.tif`.

The DEM layer shows you the elevation of the terrain, but it can sometimes seem a little abstract. It contains all the 3D information about the terrain that you need, but it doesn't look like a 3D object. To get a better impression of the terrain, it is possible to calculate a *hillshade*, which is a raster that maps the terrain using light and shadow to create a 3D-looking image.

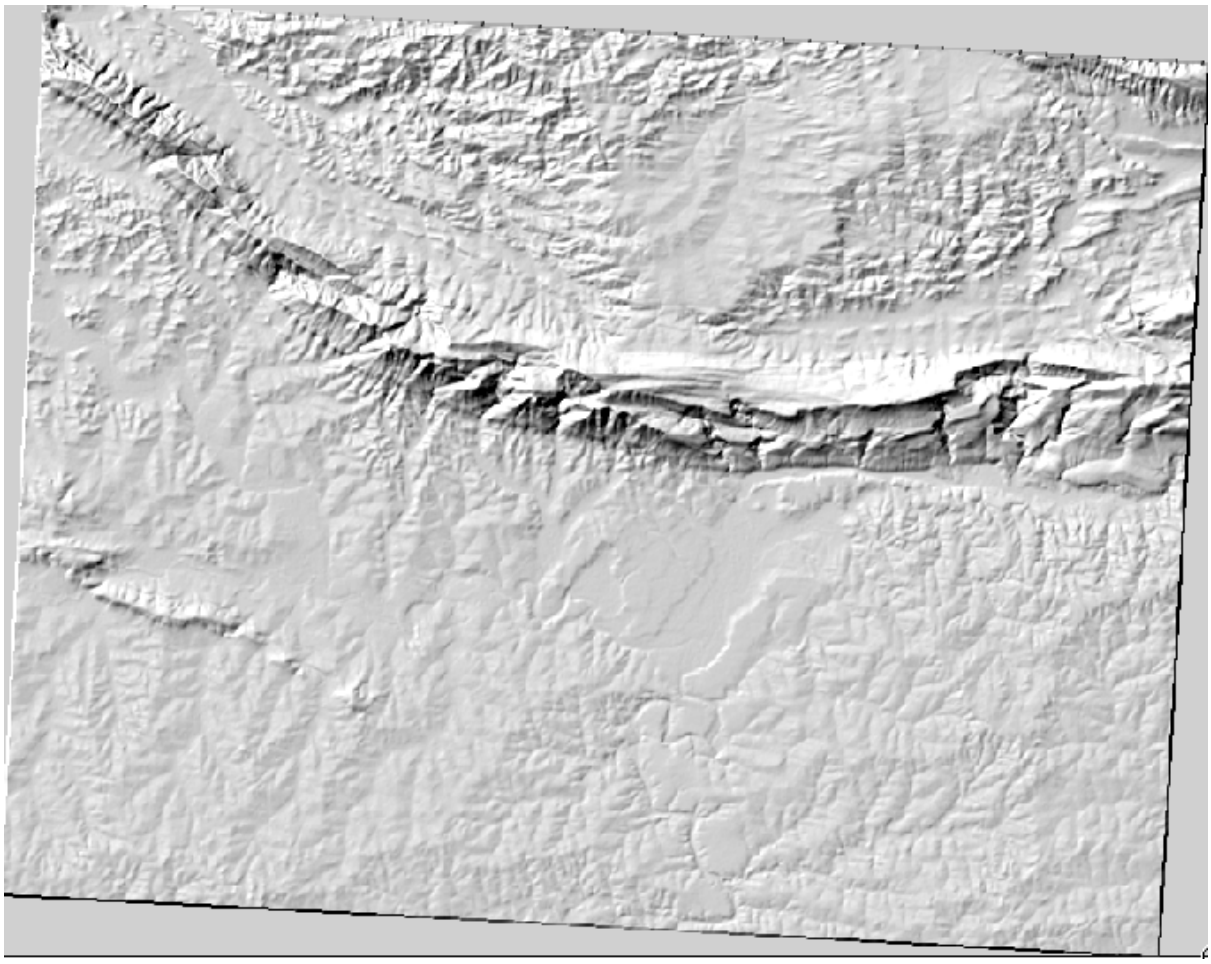
We are going to use algorithms in the *Raster* ► *Raster terrain analysis* menu.

1. Click on the *Hillshade* menu
2. The algorithm allows you to specify the position of the light source: *Azimuth* has values from 0 (North) through 90 (East), 180 (South) and 270 (West), while the *Vertical angle* sets how high the light source is (0 to 90 degrees).
3. We will use the following values:
   - *Z factor* at `1.0`
   - *Azimuth (horizontal angle)* at `300.0°`
   - *Vertical angle* at `40.0°`



4. Save the file in a new folder `exercise_data/raster_analysis/` with the name `hillshade`
5. Finally click on *Run*

You will now have a new layer called *hillshade* that looks like this:



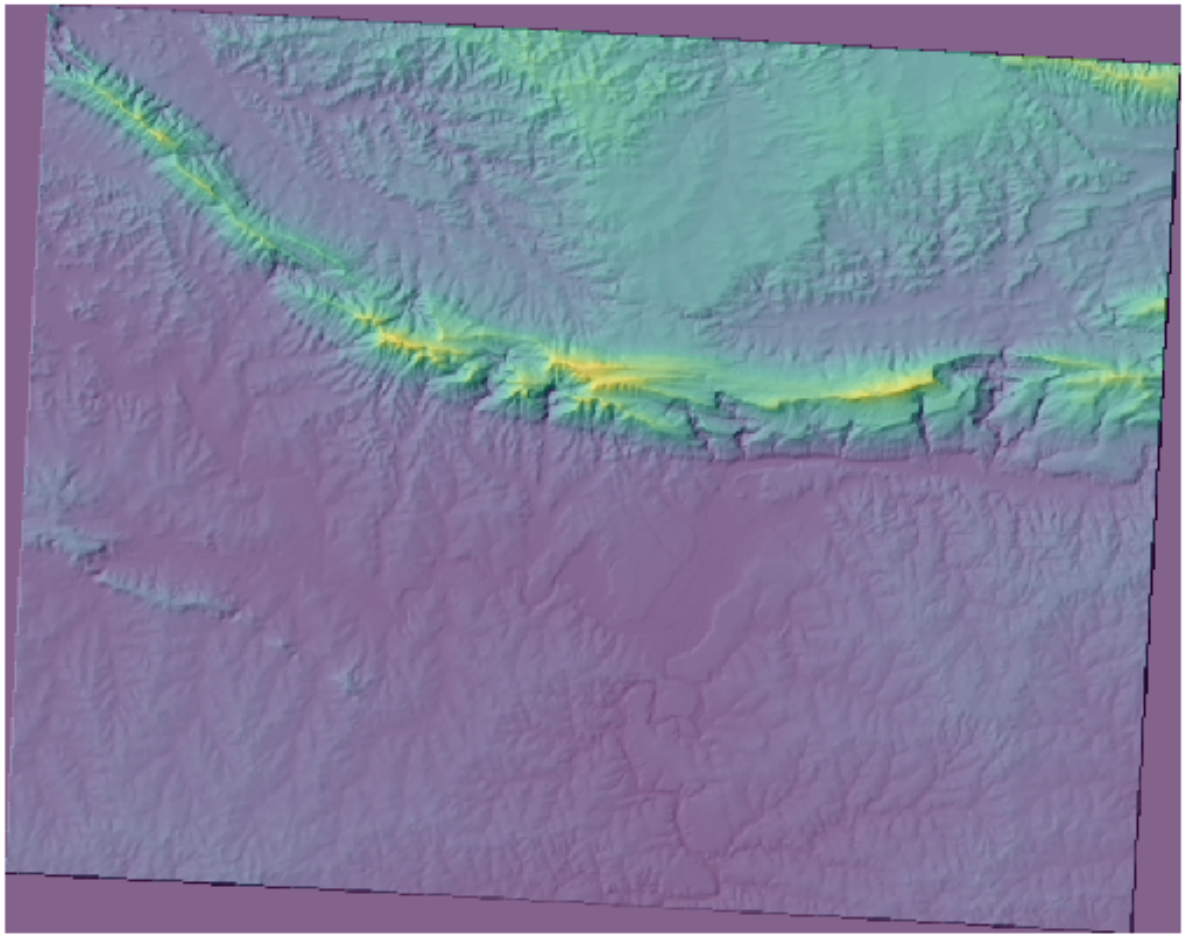
That looks nice and 3D, but can we improve on this? On its own, the hillshade looks like a plaster cast. Can't we use it together with our other, more colorful rasters somehow? Of course we can, by using the hillshade as an overlay.

7.3.2 Follow Along: Using a Hillshade as an Overlay

A hillshade can provide very useful information about the sunlight at a given time of day. But it can also be used for aesthetic purposes, to make the map look better. The key to this is setting the hillshade to being mostly transparent.

1. Change the symbology of the original *srtm_41_19* layer to use the *Pseudocolor* scheme as in the previous exercise
2. Hide all the layers except the *srtm_41_19* and *hillshade* layers
3. Click and drag the *srtm_41_19* to be beneath the *hillshade* layer in the *Layers* panel
4. Set the *hillshade* layer to be transparent by clicking on the *Transparency* tab in the layer properties
5. Set the *Global opacity* to `50%`.

   You'll get a result like this:



6. Switch the *hillshade* layer off and back on in the *Layers* panel to see the difference it makes.

Using a hillshade in this way, it's possible to enhance the topography of the landscape. If the effect doesn't seem strong enough to you, you can change the transparency of the *hillshade* layer; but of course, the brighter the hillshade becomes, the dimmer the colors behind it will be. You will need to find a balance that works for you.

Remember to save the project when you are done.

7.3.3 Follow Along: Finding the best areas

Think back to the estate agent problem, which we last addressed in the *Vector Analysis* lesson. Let us imagine that the buyers now wish to purchase a building and build a smaller cottage on the property. In the Southern Hemisphere, we know that an ideal plot for development needs to have areas on it that:

- are north-facing
- with a slope of less than 5 degrees
- But if the slope is less than 2 degrees, then the aspect doesn't matter.

Let's find the best areas for them.

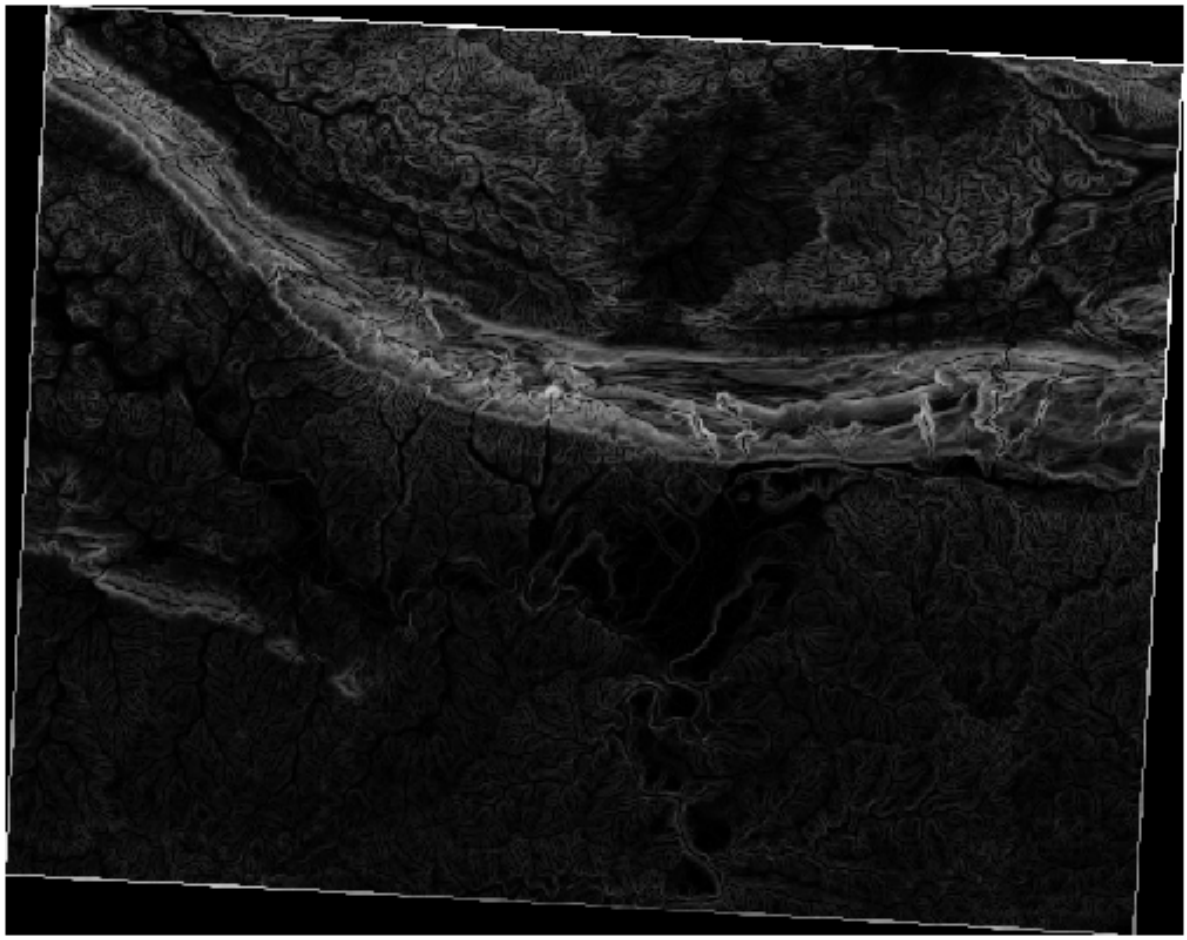
7.3.4 Follow Along: Calculating the Slope

Slope informs about how steep the terrain is. If, for example, you want to build houses on the land there, then you need land that is relatively flat.

To calculate the slope, you need to use the *Slope* algorithm of the *Processing* ► *Raster terrain analysis*.

1. Open the algorithm
2. Choose *srtm_41_19* as the *Elevation layer*
3. Keep the *Z factor* at `1.0`
4. Save the output as a file with the name `slope` in the same folder as the `hillshade`
5. Click on *Run*

Now you'll see the slope of the terrain, each pixel holding the corresponding slope value. Black pixels show flat terrain and white pixels, steep terrain:



7.3.5 Try Yourself Calculating the aspect

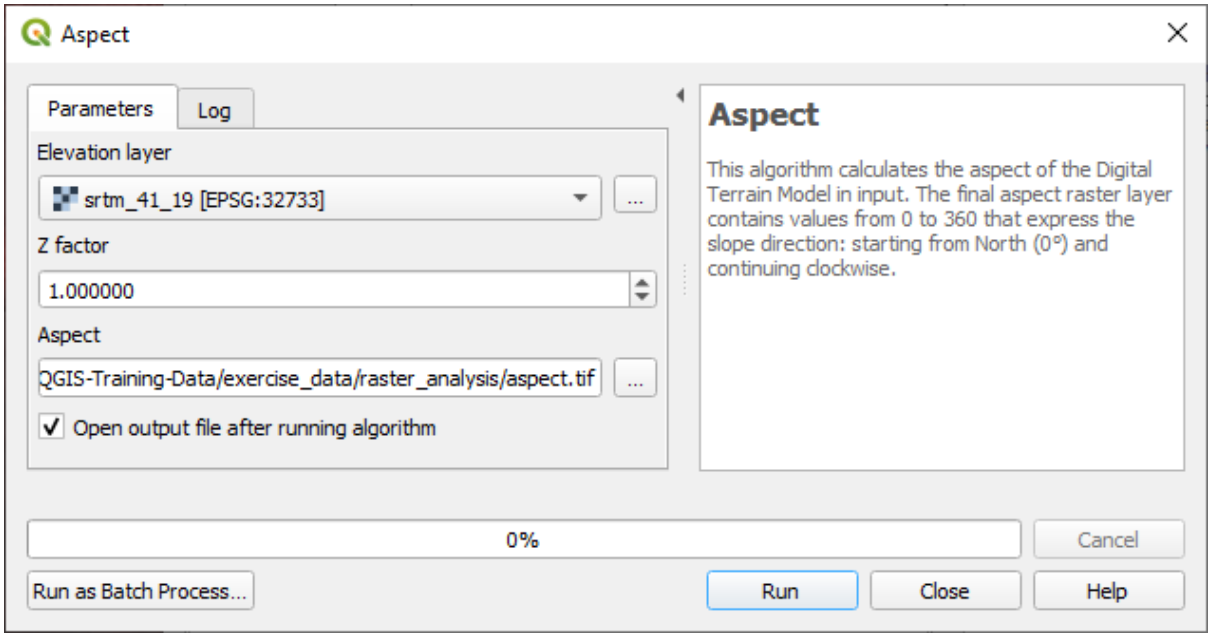
Aspect is the compass direction that the slope of the terrain faces. An aspect of 0 means that the slope is North-facing, 90 East-facing, 180 South-facing, and 270 West-facing.

Since this study is taking place in the Southern Hemisphere, properties should ideally be built on a north-facing slope so that they can remain in the sunlight.

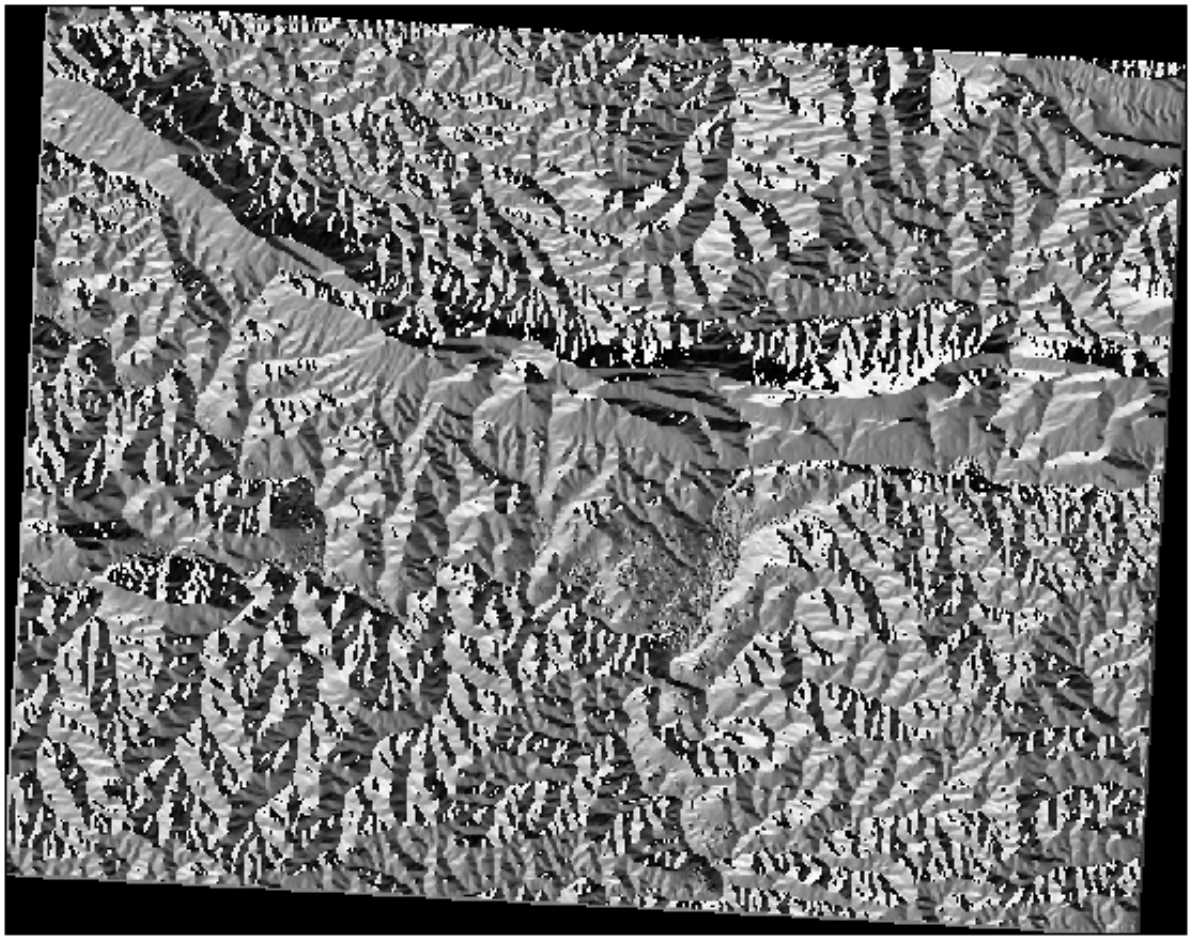
Use the *Aspect* algorithm of the *Processing* ► *Raster terrain analysis* to get the `aspect` layer saved along with the `slope`.

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 Follow Along: Finding the north-facing aspect

Now, you have rasters showing you the slope as well as the aspect, but you have no way of knowing where ideal conditions are satisfied at once. How could this analysis be done?

The answer lies with the *Raster calculator*.

QGIS has different raster calculators available:

- *Raster* ► *Raster Calculator*
- In processing:
  - *Raster Analysis* ► *Raster calculator*
  - *GDAL* ► *Raster miscellaneous* ► *Raster calculator*
  - *SAGA* ► *Raster calculus* ► *Raster calculator*

Each tool is leading to the same results, but the syntax may be slightly different and the availability of operators may vary.

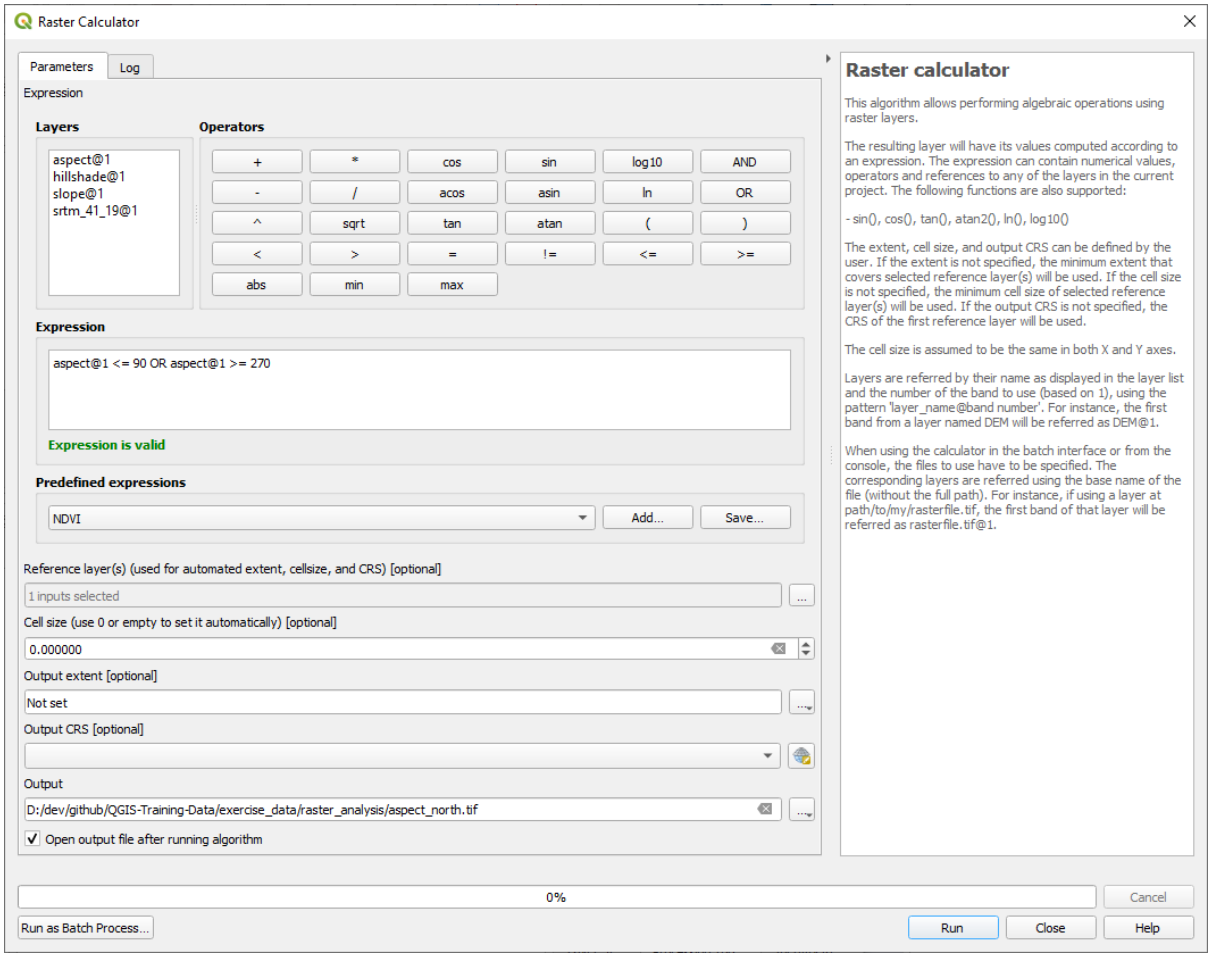
We will use *Raster Analysis* ► *Raster calculator* in the *Processing Toolbox*

1. Open the tool by double clicking on it.
   - The upper left part of the dialog lists all the loaded raster layers as `name@N`, where `name` is the name of the layer and `N` is the band.
   - In the upper right part you will see a lot of different operators. Stop for a moment to think that a raster is an image. You should see it as a 2D matrix filled with numbers.
2. North is at 0 (zero) degrees, so for the terrain to face north, its aspect needs to be greater than 270 degrees or less than 90 degrees. Therefore the formula is:

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

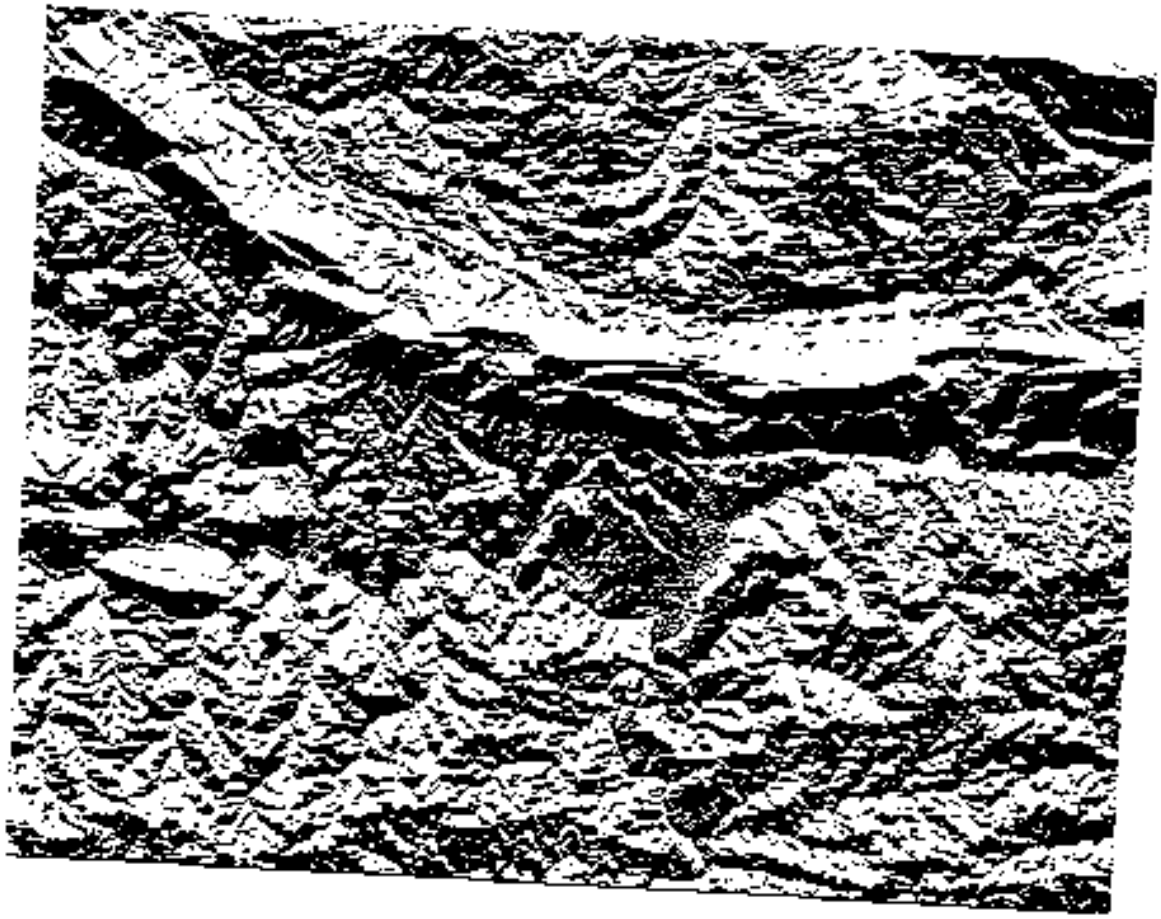
3. Now you have to set up the raster details, like the cell size, extent and CRS. This can be done manually or it can be automatically set by choosing a `Reference layer`. Choose this last option by clicking on the … button next to the *Reference layer(s)* parameter.
4. In the dialog, choose the *aspect* layer, because we want to obtain a layer with the same resolution.
5. Save the layer as `aspect_north`.

   The dialog should look like:



6. Finally click on *Run*.

Your result will be this:



The output values are `0` or `1`. What does it mean? For each pixel in the raster, the formula we wrote returns whether it matches the conditions or not. Therefore the final result will be **False** (0) and **True** (1).

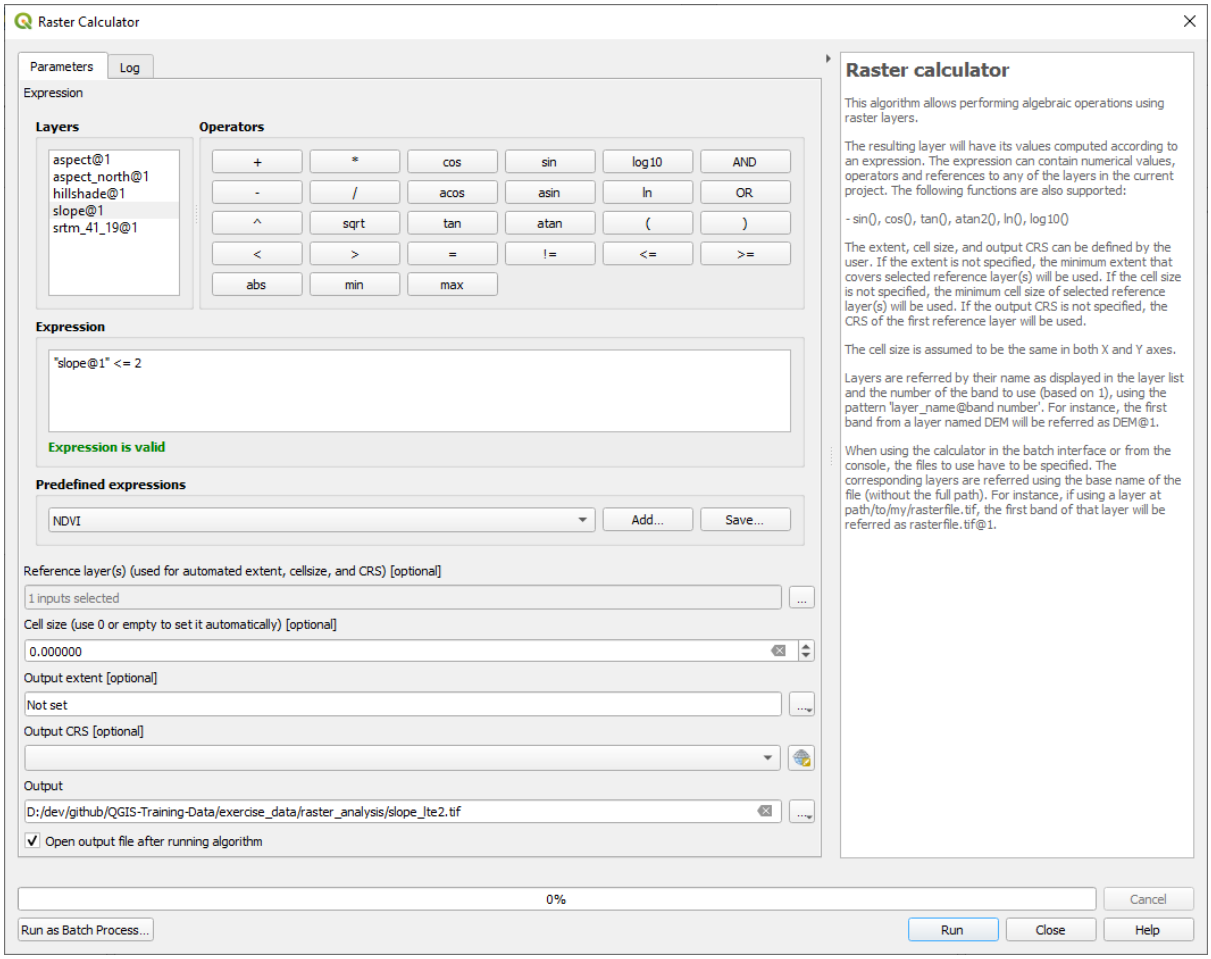
7.3.7 Try Yourself More criteria

Now that you have done the aspect, create two new layers from the DEM.

- The first shall identify areas where the slope is less than or equal to `2` degrees
- The second is similar, but the slope should be less than or equal to `5` degrees.
- Save them under `exercise_data/raster_analysis` as `slope_lte2.tif` and `slope_lte5.tif`.

Answer

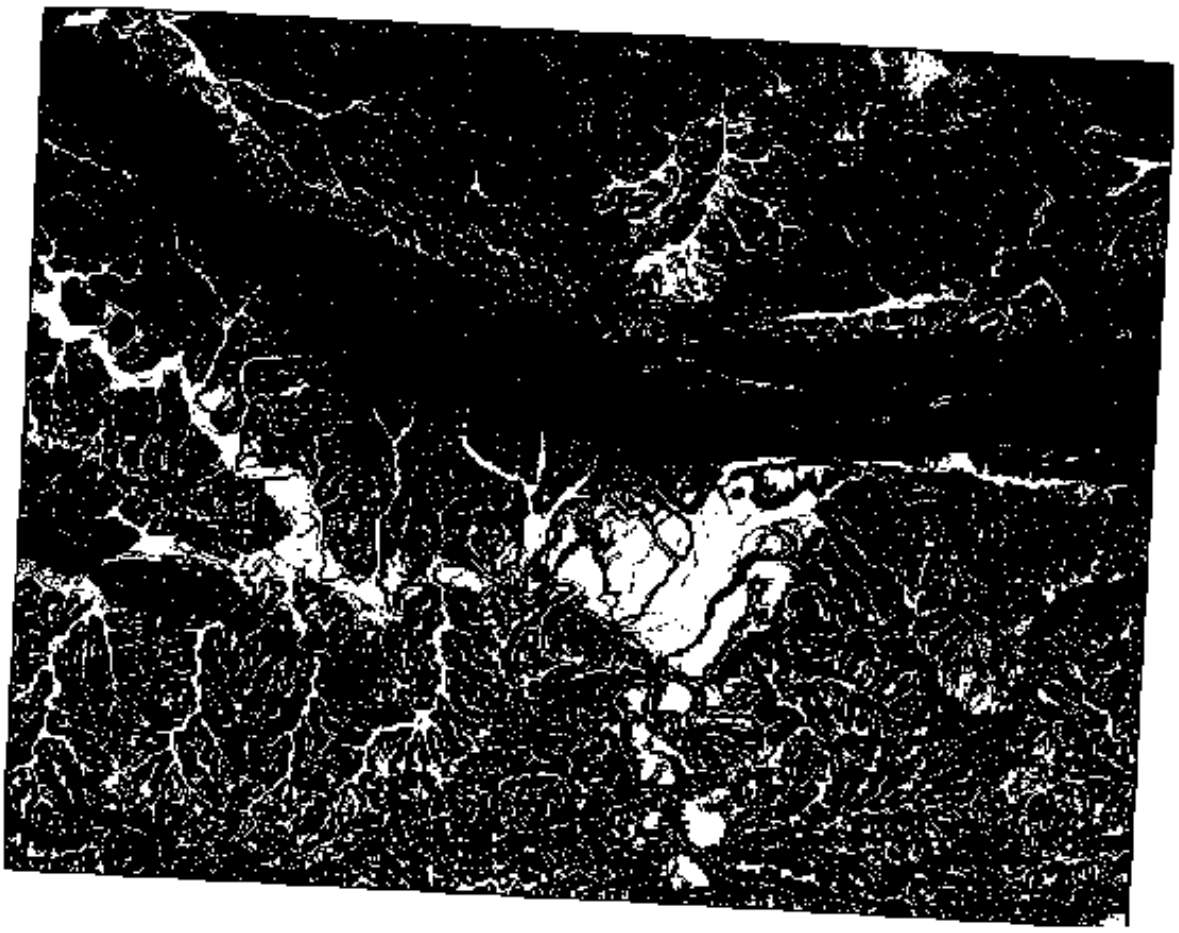
- Set your *Raster calculator* dialog up with:
- the following expression: `slope@1 <= 2`
- the `slope` layer as the *Reference layer(s)*



- 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 Follow Along: Combining Raster Analysis Results

Now you have generated three raster layers from the DEM:

- *aspect_north*: terrain facing north
- *slope_lte2*: slope equal to or below 2 degrees
- *slope_lte5*: slope equal to or below 5 degrees

Where the condition is met, the pixel value is `1`. Elsewhere, it is `0`. Therefore, if you multiply these rasters, the pixels that have a value of `1` for all of them will get a value of `1` (the rest will get `0`).

The conditions to be met are:

- at or below 5 degrees of slope, the terrain must face north
- at or below 2 degrees of slope, the direction that the terrain faces does not matter.

Therefore, you need to find areas where the slope is at or below five degrees `AND` the terrain is facing north, `OR` the slope is at or below 2 degrees. Such terrain would be suitable for development.

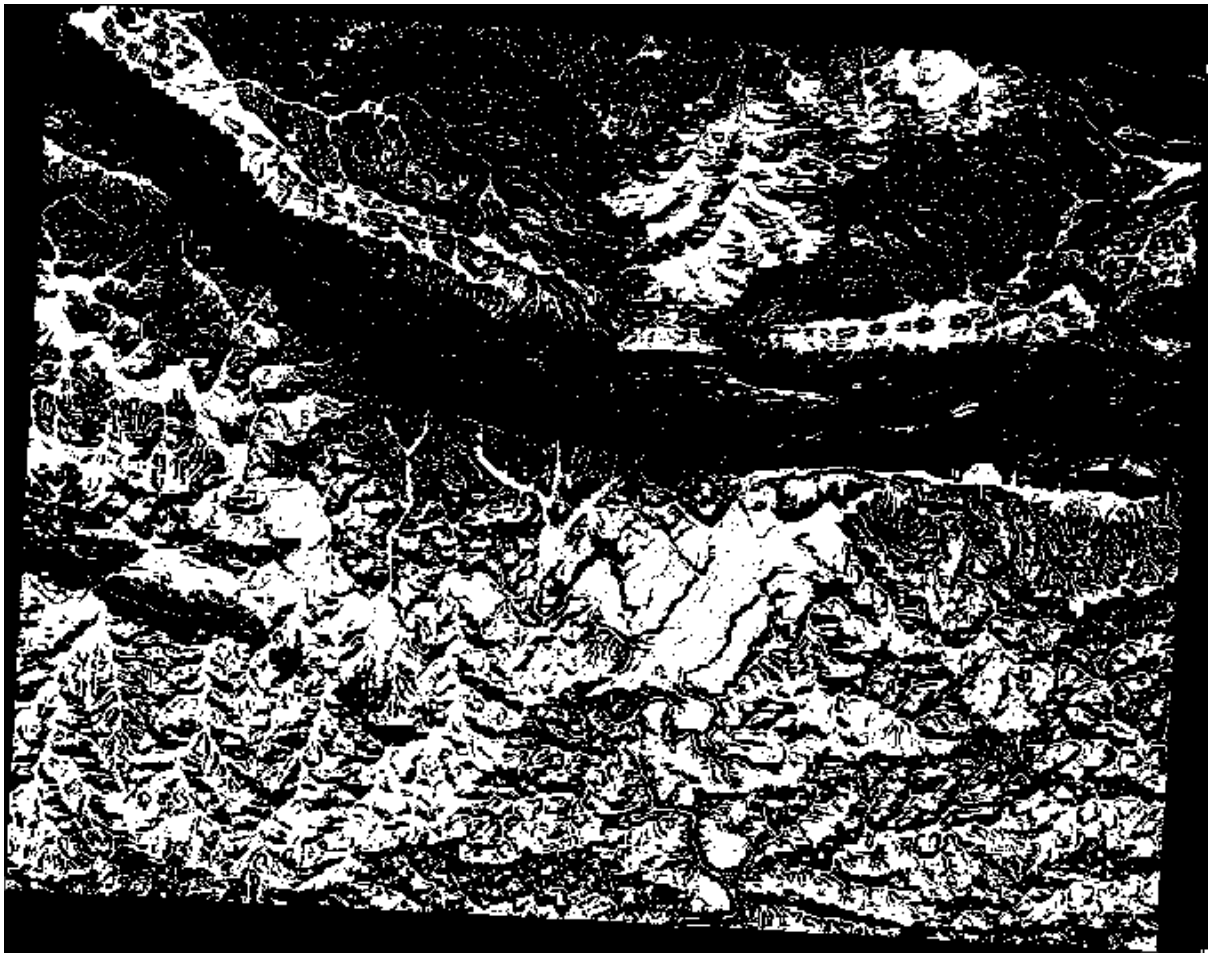
To calculate the areas that satisfy these criteria:

1. Open the *Raster calculator* again
2. Use this expression in *Expression*:

```
( aspect_north@1 = 1 AND slope_lte5@1 = 1 ) OR slope_lte2@1 = 1
```

3. Set the *Reference layer(s)* parameter to `aspect_north` (it does not matter if you choose another - they have all been calculated from `srtm_41_19`)
4. Save the output under `exercise_data/raster_analysis/` as `all_conditions.tif`
5. Click *Run*

Das Ergebnis:



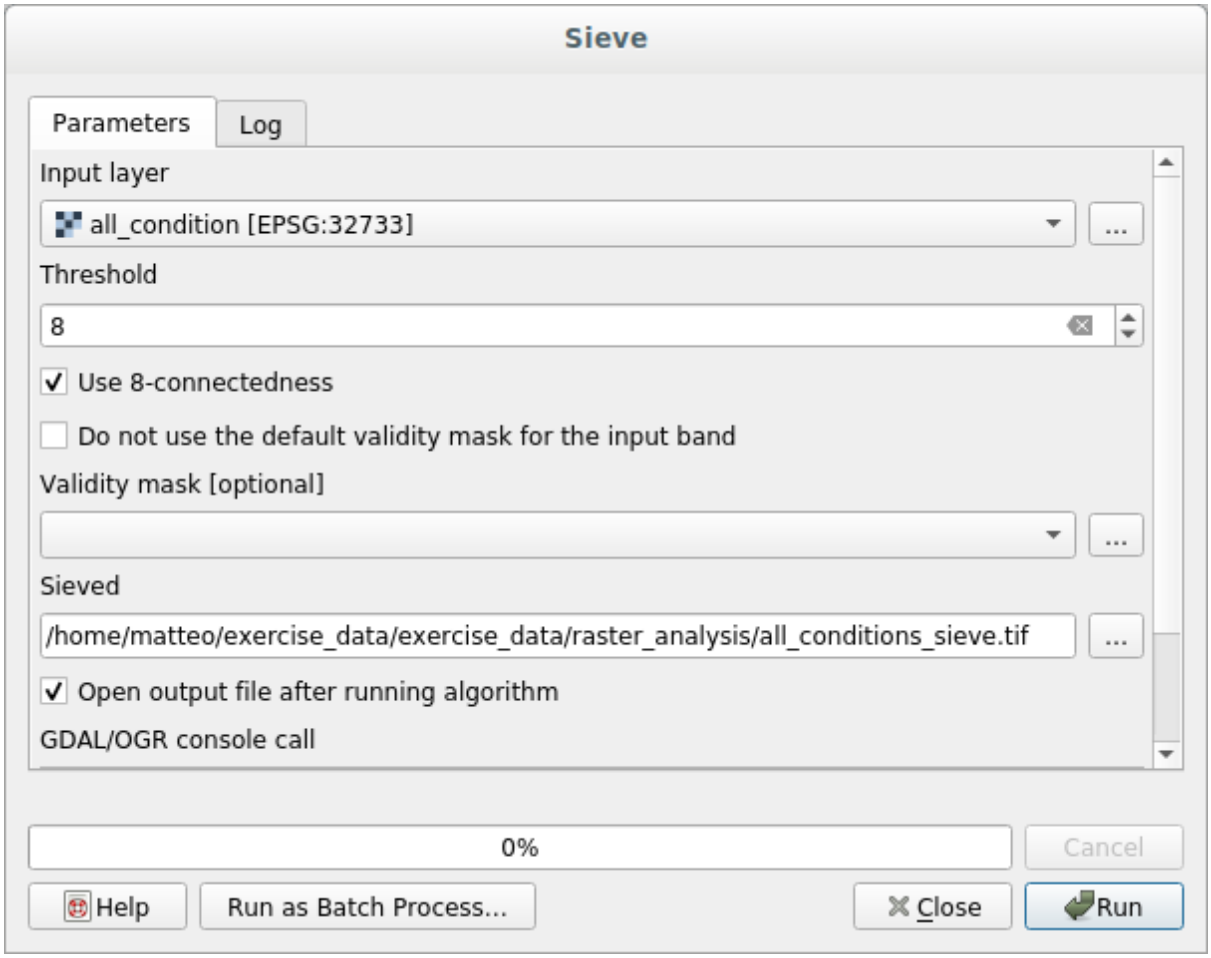
Hinweis: The previous steps could have been simplified using the following command:

```
((aspect@1 <= 90 OR  aspect@1 >= 270) AND slope@1 <= 5) OR slope@1 <= 2
```

7.3.9 Follow Along: Simplifying the Raster

As you can see from the image above, the combined analysis has left us with many, very small areas where the conditions are met (in white). But these aren't really useful for our analysis, since they are too small to build anything on. Let us get rid of all these tiny unusable areas.

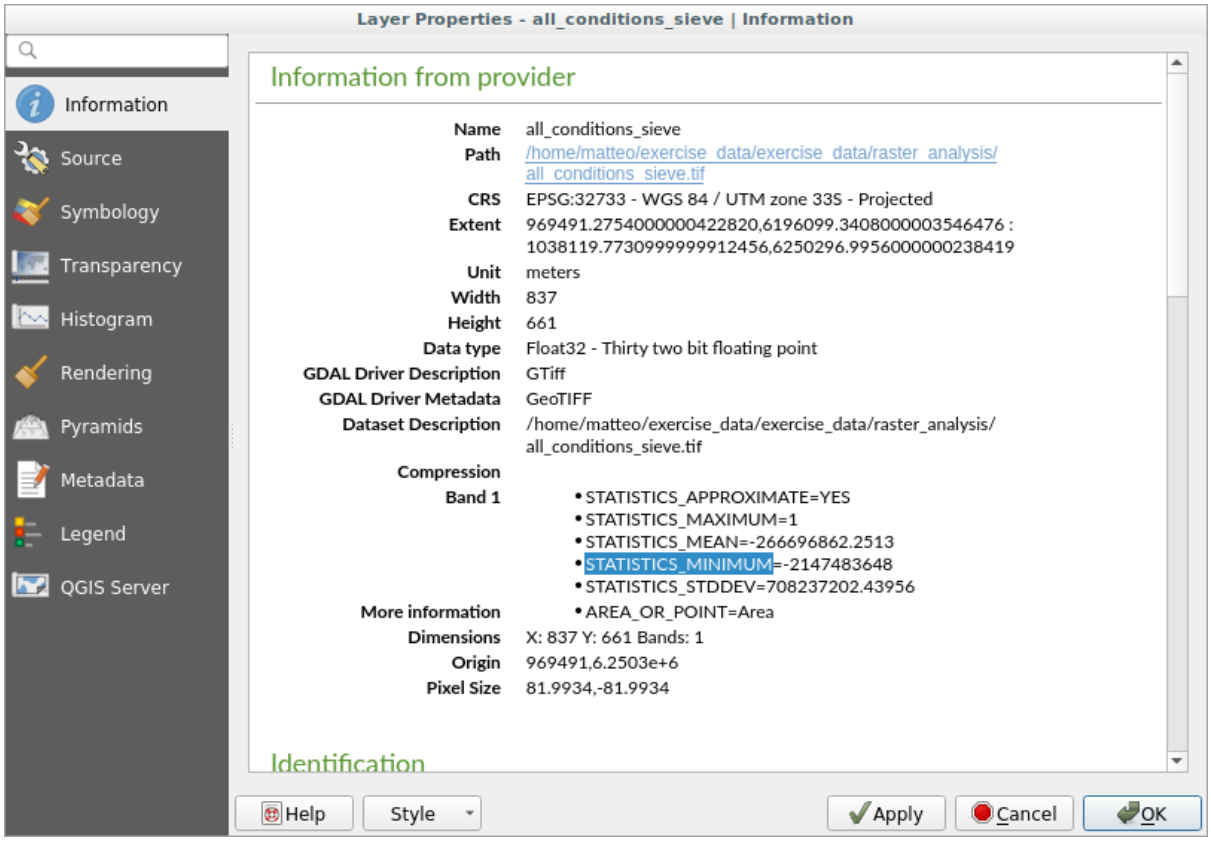
1. Open the *Sieve* tool (*GDAL* ► *Raster Analysis* in the *Processing Toolbox*)
2. Set the *Input file* to `all_conditions`, and the *Sieved* to `all_conditions_sieve.tif` (under `exercise_data/raster_analysis/`).
3. Set the *Threshold* to 8 (minimum eight contiguous pixels), and check *Use 8-connectedness*.



Once processing is done, the new layer will be loaded.

What is going on? The answer lies in the new raster file's metadata.

4. View the metadata under the *Information* tab of the *Layer Properties* dialog. Look the `STATISTICS_MINIMUM` value:



This raster, like the one it is derived from, should only feature the values `1` and `0`, but it has also a very large negative number. Investigation of the data shows that this number acts as a null value. Since we are only after areas that weren't filtered out, let us set these null values to zero.

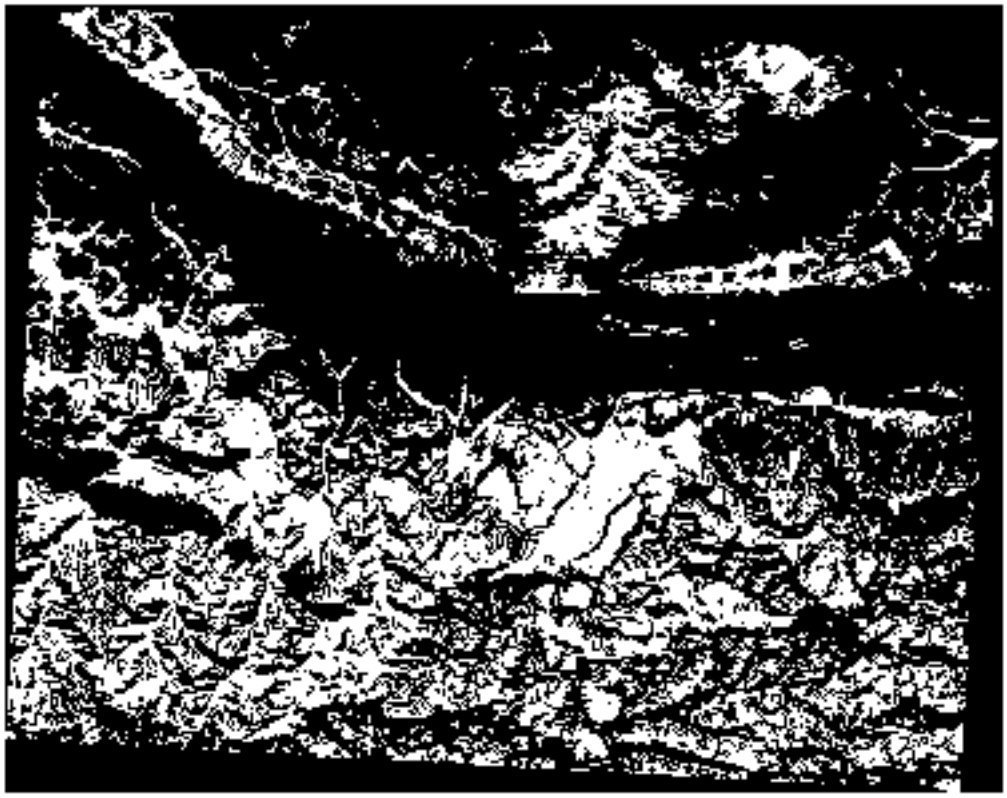
5. Open the *Raster Calculator*, and build this expression:

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

This will maintain all non-negative values, and set the negative numbers to zero, leaving all the areas with value `1` intact.

6. Save the output under `exercise_data/raster_analysis/` as `all_conditions_simple.tif`.

Your output looks like this:



This is what was expected: a simplified version of the earlier results. Remember that if the results you get from a tool aren't what you expected, viewing the metadata (and vector attributes, if applicable) can prove essential to solving the problem.

7.3.10 Follow Along: Reclassifying the Raster

We have used the *Raster calculator* to do calculations on raster layers. There is another powerful tool that we can use to extract information from existing layers.

Back to the `aspect` layer. We know now that it has numerical values within a range from 0 through 360. What we want to do is to *reclassify* this layer to other discrete values (from 1 to 4), depending on the aspect:

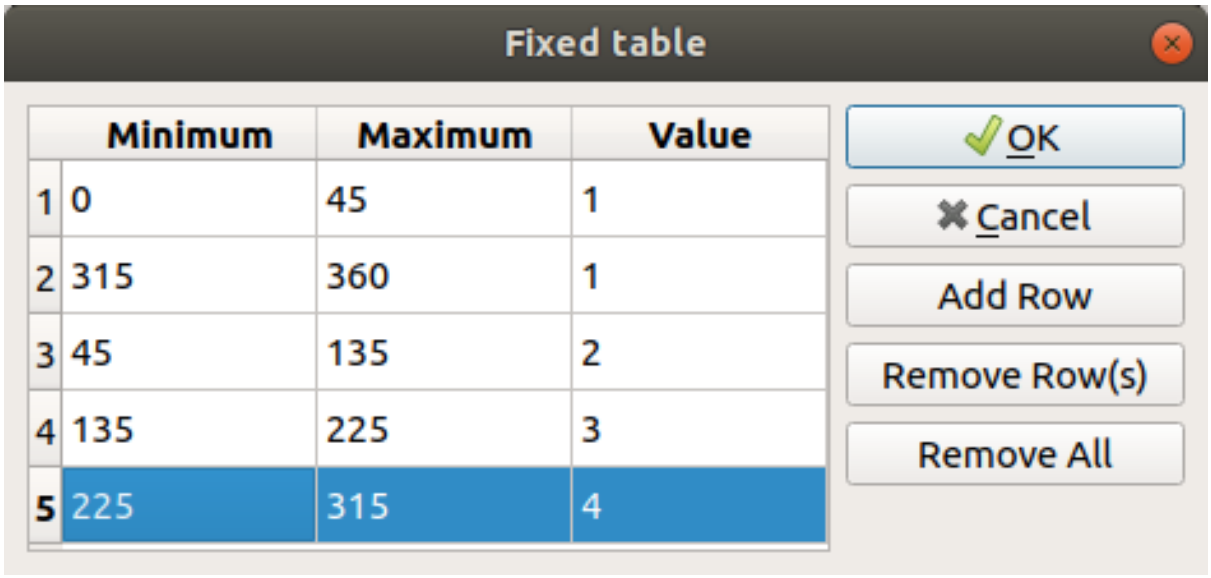
- 1 = North (from 0 to 45 and from 315 to 360);
- 2 = East (from 45 to 135)
- 3 = South (from 135 to 225)

- 4 = West (from 225 to 315)

This operation can be achieved with the raster calculator, but the formula would become very very large.

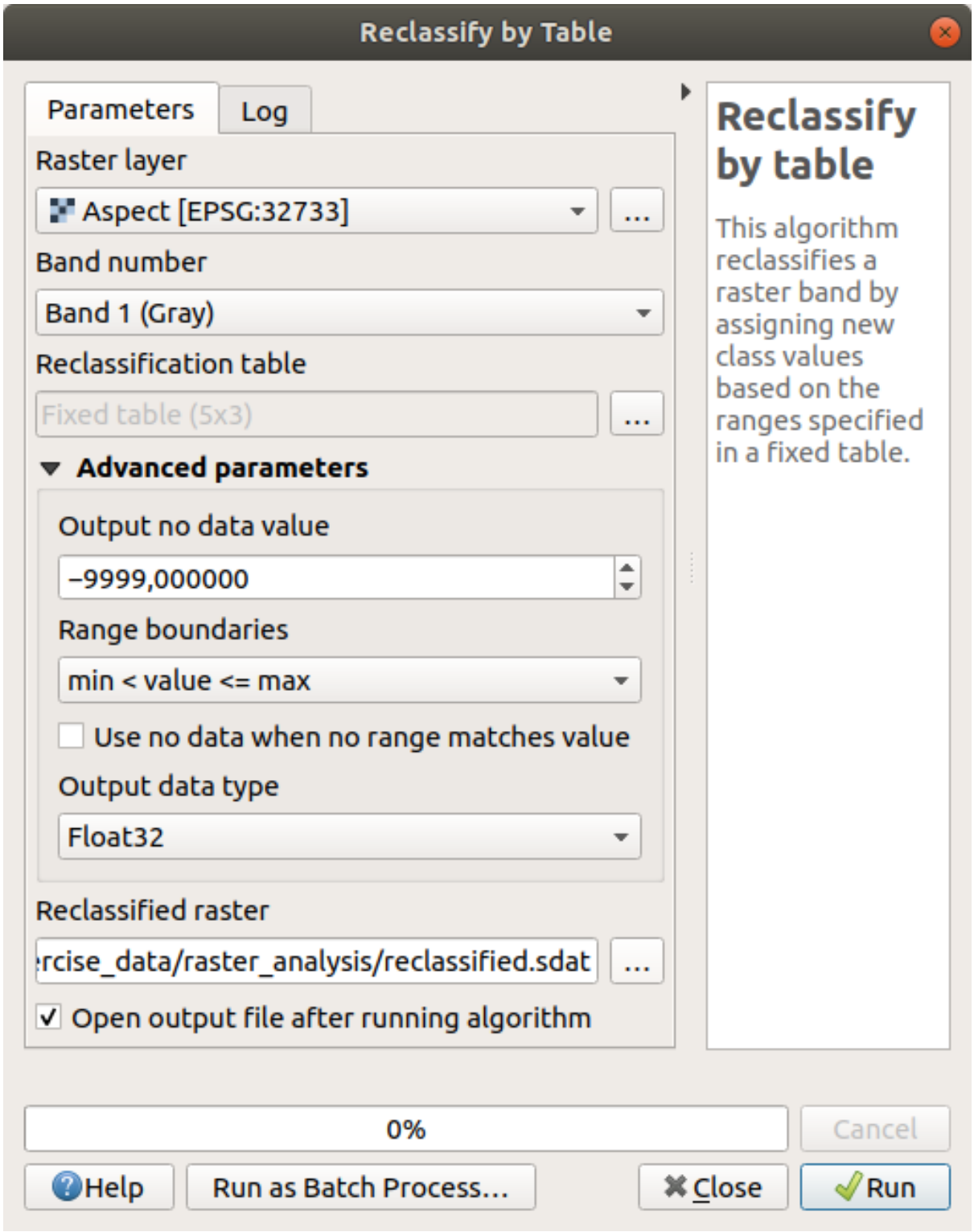
The alternative tool is the *Reclassify by table* tool in *Raster analysis* in the *Processing Toolbox*.

1. Open the tool
2. Choose *aspect* as the `Input raster layer`
3. Click on the … of *Reclassification table*. A table-like dialog will pop up, where you can choose the minimum, maximum and new values for each class.
4. Click on the *Add row* button and add 5 rows. Fill in each row as the following picture and click *OK*:



   The method used by the algorithm to treat the threshold values of each class is defined by the *Range boundaries*.

5. Save the layer as `reclassified.tif` in the `exercise_data/raster_analysis/` folder



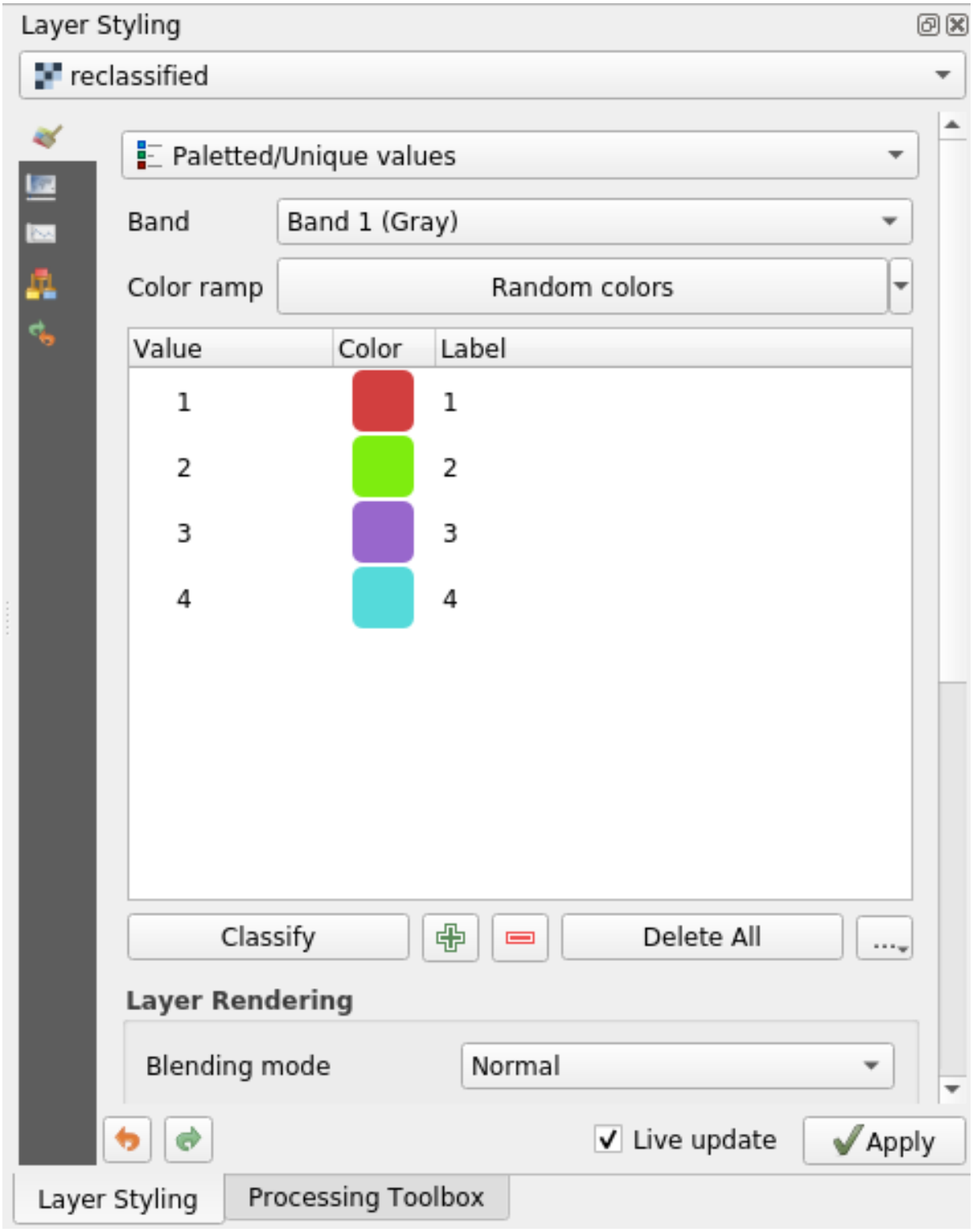
6. Click on *Run*

If you compare the native *aspect* layer with the *reclassified* one, there are not big differences. But by looking at the legend, you can see that the values go from `1` to `4`.

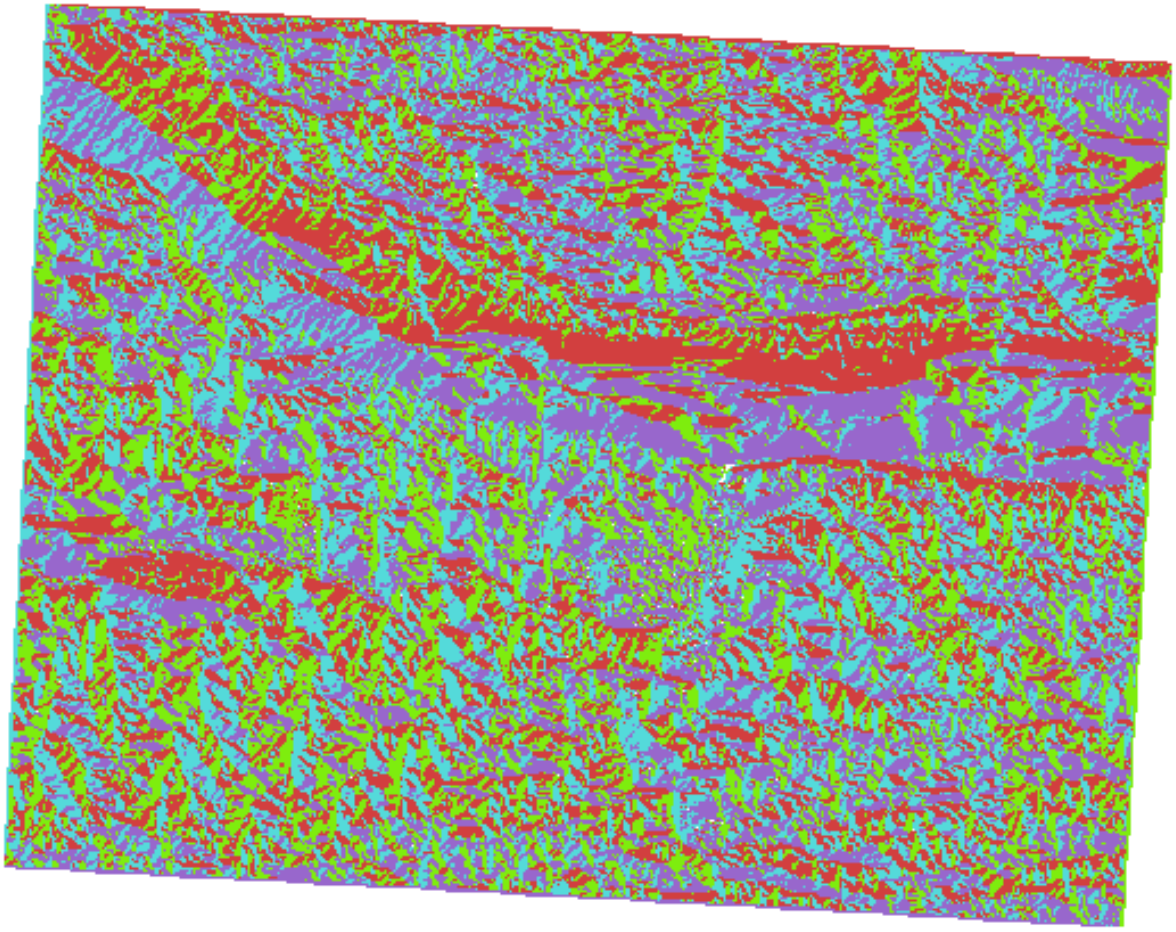
Let us give this layer a better style.

1. Open the *Layer Styling* panel
2. Choose *Paletted/Unique values*, instead of *Singleband gray*

3. Click on the *Classify* button to automatically fetch the values and assign them random colors:



The output should look like this (you can have different colors given that they have been randomly generated):



With this reclassification and the paletted style applied to the layer, you can immediately differentiate the aspect areas.

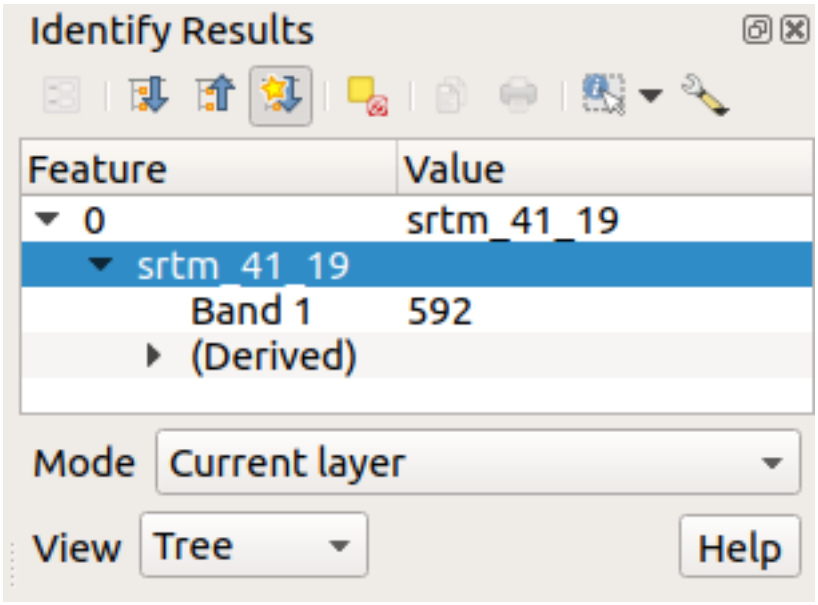
7.3.11 Follow Along: Querying the raster

Unlike vector layers, raster layers don't have an attribute table. Each pixel contains one or more numerical values (singleband or multiband rasters).

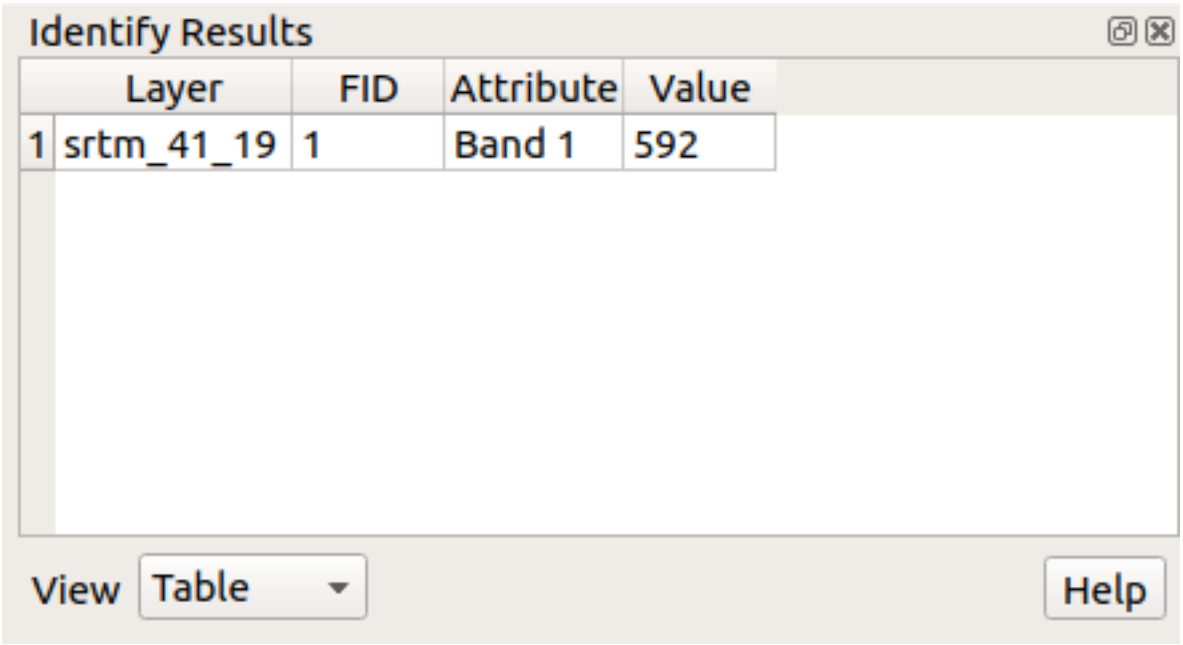
All the raster layers we used in this exercise consist of just one band. Depending on the layer, pixel values may represent elevation, aspect or slope values.

How can we query the raster layer to get the value of a pixel? We can use the Identify Features button!

1. Select the tool from the Attributes toolbar.
2. Click on a random location of the *srtm_41_19* layer. *Identify Results* will appear with the value of the band at the clicked location:

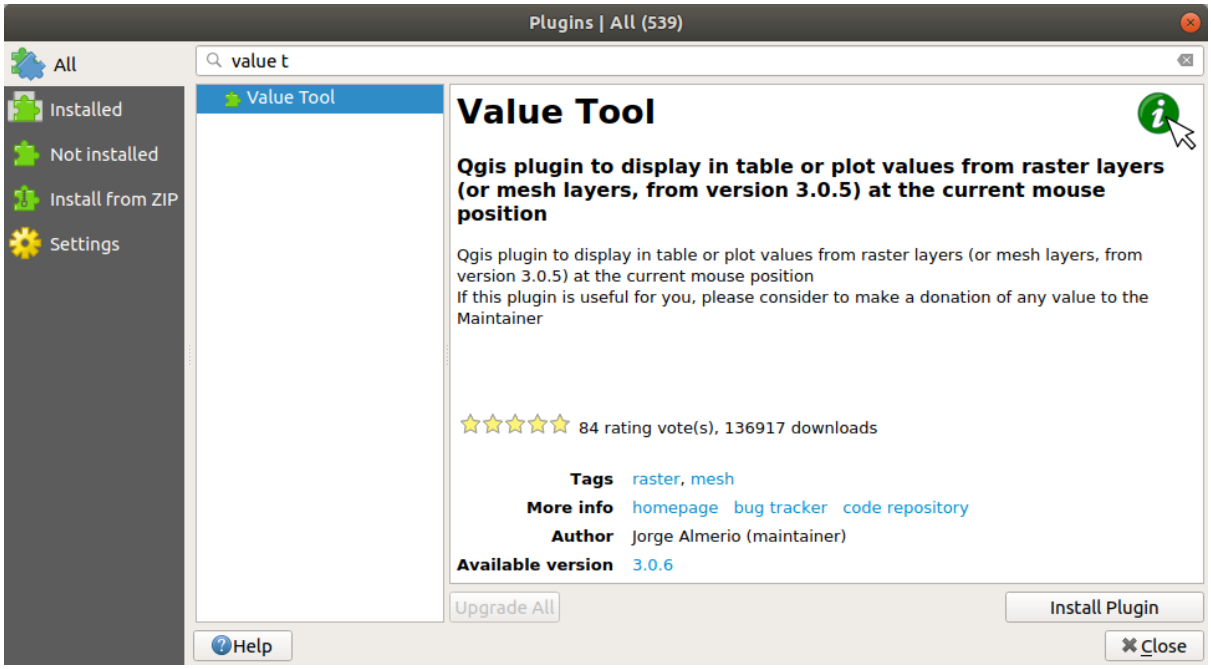


3. You can change the output of the *Identify Results* panel from the current `tree` mode to a `table` one by selecting *Table* in the *View* menu at the bottom of the panel:



Clicking each pixel to get the value of the raster could become annoying after a while. We can use the *Value Tool* plugin to solve this problem.

1. Go to *Plugins* ► *Manage/Install Plugins...*
2. In the *All* tab, type `value t` in the search box
3. Select the *Value Tool* plugin, press *Install Plugin* and then *Close* the dialog.

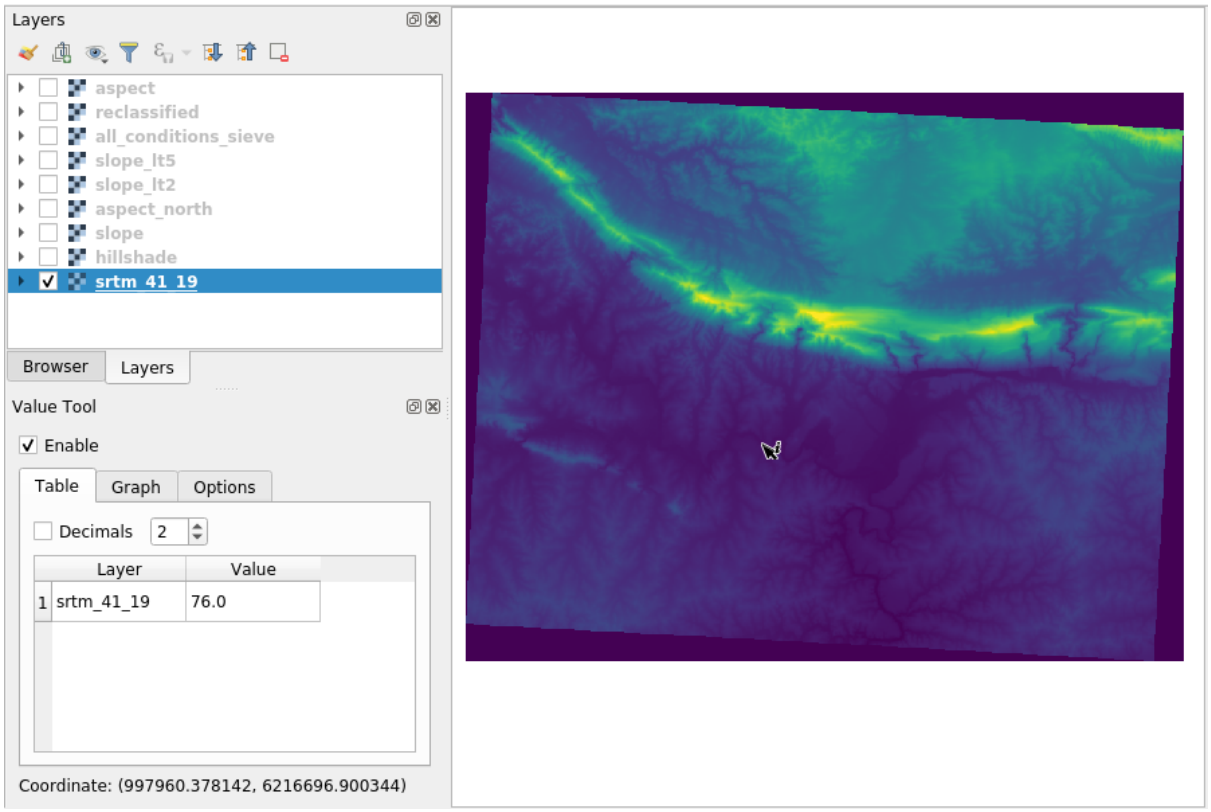


The new *Value Tool* panel will appear.

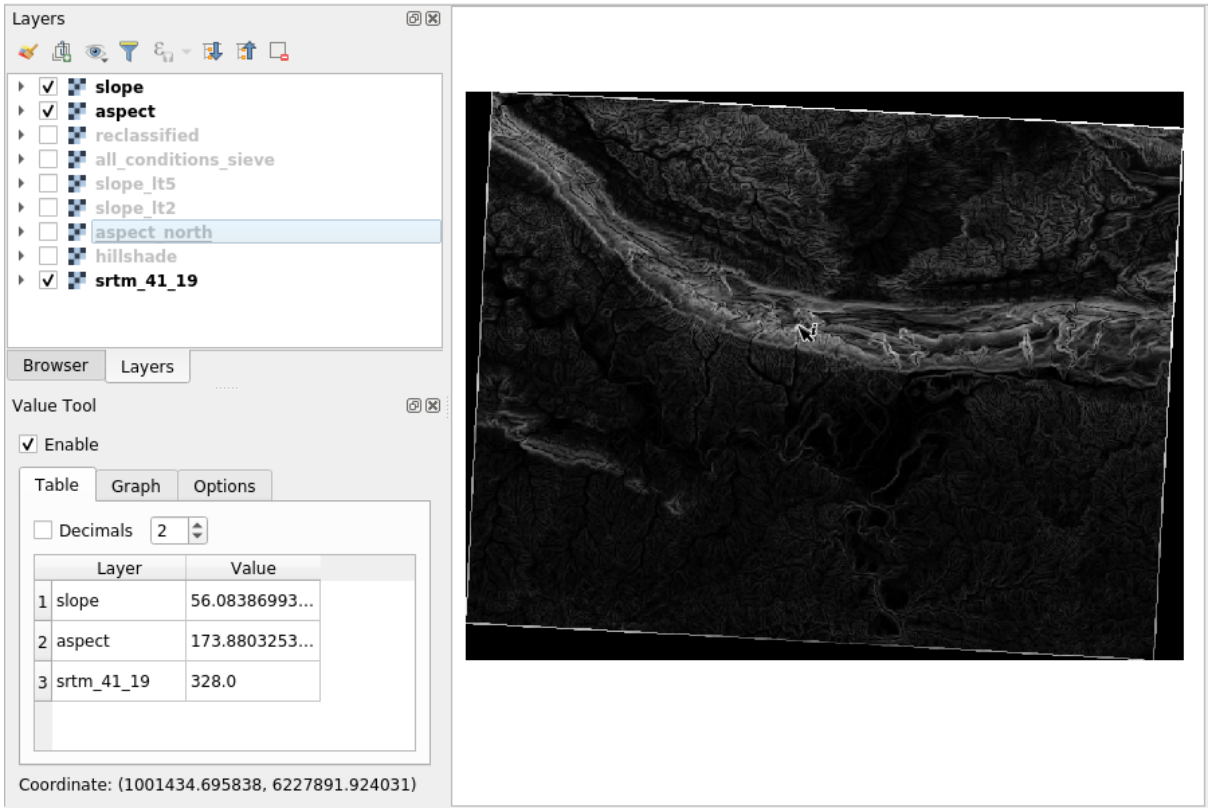
Tipp: If you close the panel you can reopen it by enabling it in the *View* ► *Panels* ► *Value Tool* or by clicking on the icon in the toolbar.

---

4. To use the plugin just check the *Enable* checkbox and be sure that the `srtm_41_19` layer is active (checked) in the *Layers* panel.
5. Move the cursor over the map to see the value of the pixels.



6. But there is more. The Value Tool plugin allows you to query **all** the active raster layers in the *Layers* panel. Set the *aspect* and *slope* layers active again and hover the mouse on the map:



7.3.12 In Conclusion

You've seen how to derive all kinds of analysis products from a DEM. These include hillshade, slope and aspect calculations. You've also seen how to use the raster calculator to further analyze and combine these results. Finally you learned how to reclassify a layer and how to query the results.

7.3.13 What's Next?

Now you have two analyses: the vector analysis which shows you the potentially suitable plots, and the raster analysis that shows you the potentially suitable terrain. How can these be combined to arrive at a final result for this problem? That's the topic for the next lesson, starting in the next module.

KAPITEL 8

Module: Fertigstellung der Analyse

Sie haben nun zwei Hälften einer Analyse: einen Vektor- und einen Rasterteil. In diesem Modul sehen Sie, wie Sie sie kombinieren können. Sie werden die Analyse abschließen und die Endergebnisse präsentieren.

8.1 Lesson: Raster to Vector Conversion

Converting between raster and vector formats allows you to make use of both raster and vector data when solving a GIS problem, as well as using the various analysis methods unique to these two forms of geographic data. This increases the flexibility you have when considering data sources and processing methods for solving a GIS problem.

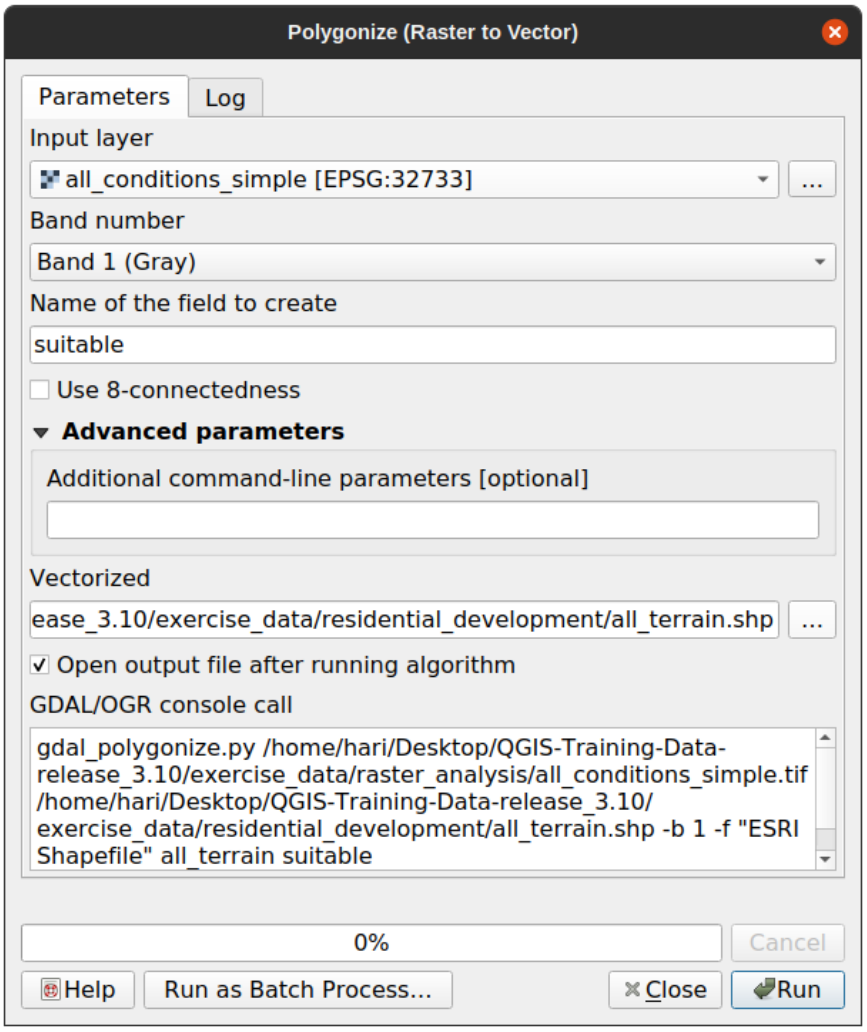
To combine a raster and vector analysis, you need to convert the one type of data to the other. Let's convert the raster result of the previous lesson to a vector.

The goal for this lesson: To get the raster result into a vector that can be used to complete the analysis.

8.1.1 Follow Along: The *Raster to Vector* Tool

Start with the map from the last module, `raster_analysis.qgs`. There you should have the `all_conditions_simple.tif` calculated during the previous exercises.

- Click on *Raster* ► *Conversion* ► *Polygonize (Raster to Vector)*. The tool dialog will appear.
- Set it up like this:



- Change the field name (describing the values of the raster) to *suitable*.
- Save the layer under `exercise_data/residential_development` as `all_terrain.shp`.

Now you have a vector file which contains all the values of the raster, but the only areas you're interested in are those that are suitable; i.e., those polygons where the value of *suitable* is `1`. You can change the style of this layer if you want to have a clearer visualization of it.

8.1.2 Try Yourself

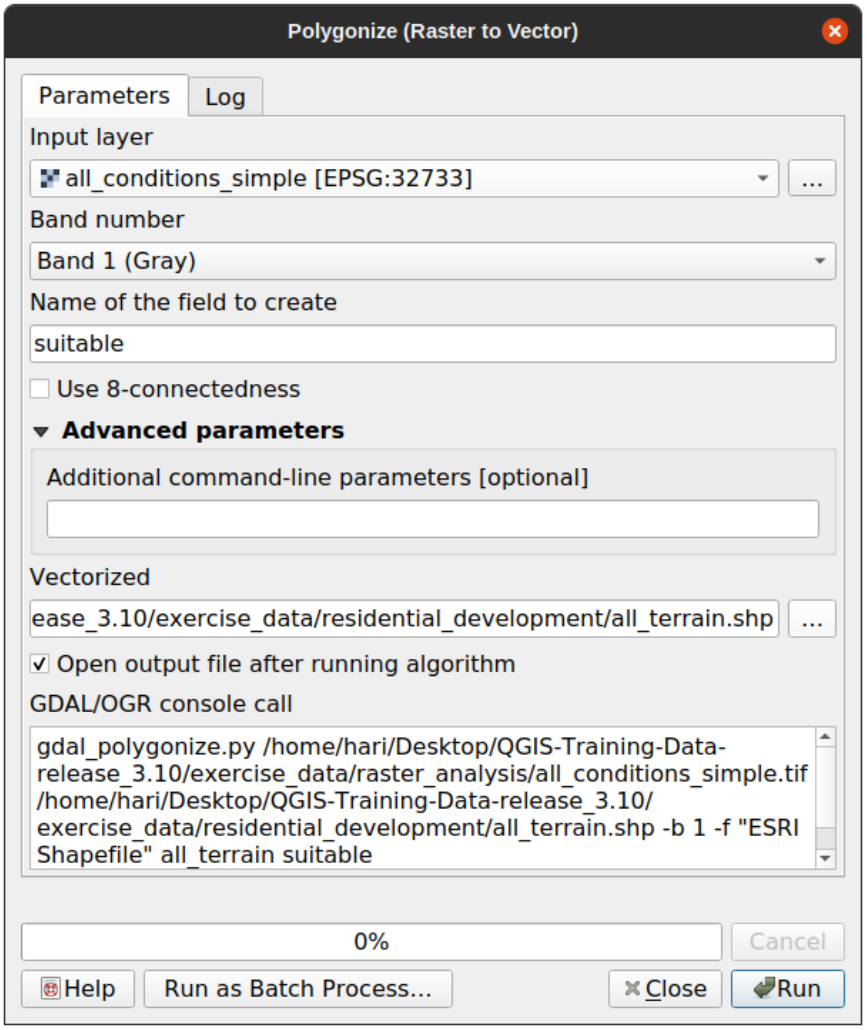
Refer back to the module on vector analysis.

- Create a new vector file that contains only the polygons where *suitable* has the value of `1`.
- Save the new file under `exercise_data/residential_development/` as `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:

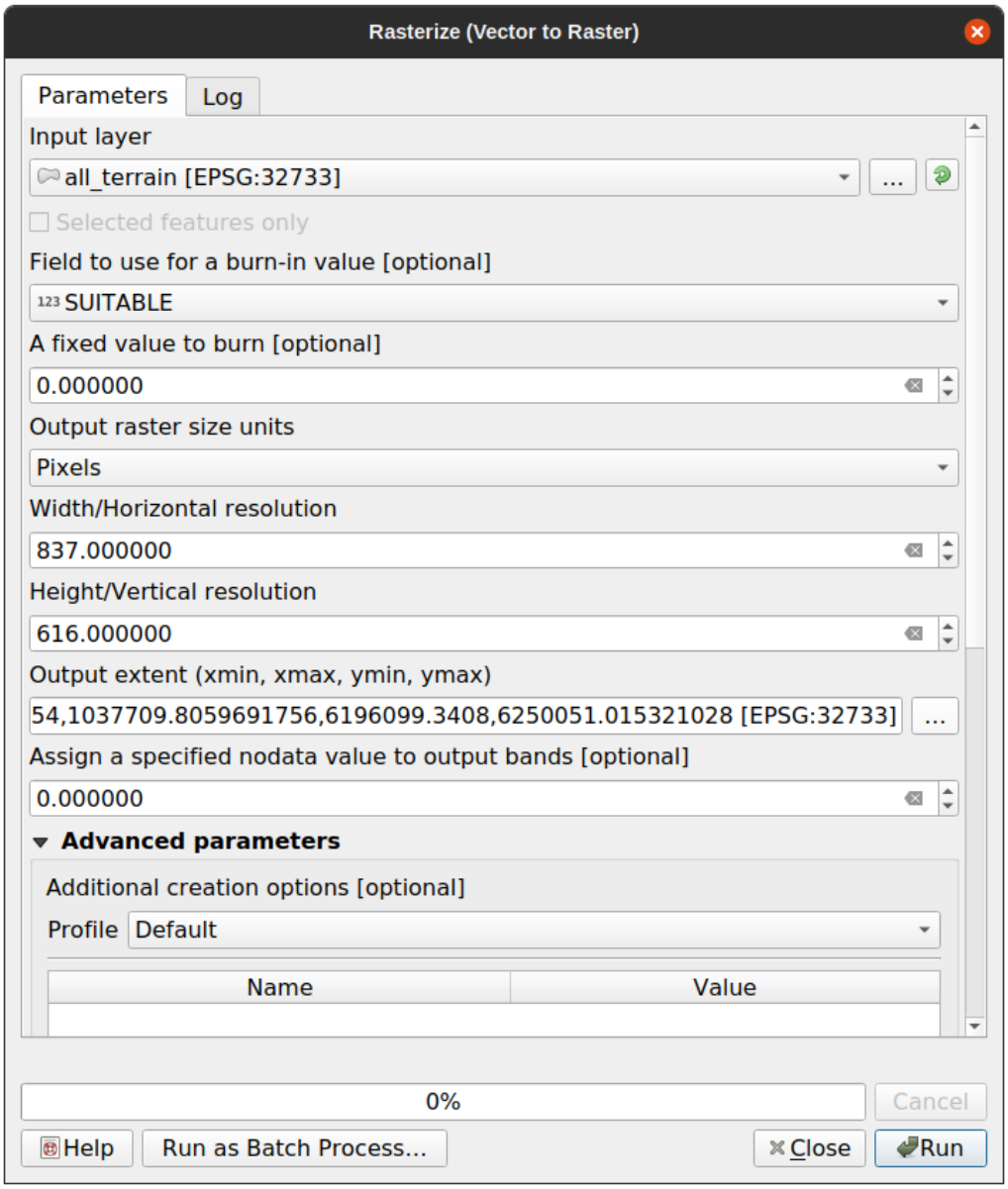


4. You can save this layer by right-clicking on the *all_terrain* layer in the *Layers* panel and choosing *Save As...*, then continue as per the instructions.

8.1.3 Follow Along: The *Vector to Raster* Tool

Although unnecessary for our current problem, it's useful to know about the opposite conversion from the one performed above. Convert to raster the `suitable_terrain.shp` vector file you just created in previous step.

- Click on *Raster* ► *Conversion* ► *Rasterize (Vector to Raster)* to start this tool, then set it up as in the screenshot below:



- *Input layer* is *all_terrain*.
- Field name is *suitable*.
- *Output raster size units* is *Pixels*.
- *Width* and *Height* are `837` and `661`, respectively.
- Get the *Output extent* from the *all_terrain* layer.
- Set output file *Rasterized* to `exercise_data/residential_development/raster_conversion.tif`.

Bemerkung: The size of the output image is specified here to be the same as the original raster which was vectorized. To view the dimensions of an image, open its metadata (*Metadata* tab in the *Layer Properties*).

- Click *OK* on the dialog to begin the conversion process.
- When it is complete, gauge its success by comparing the new raster with the original one. They should match up exactly, pixel for pixel.

8.1.4 In Conclusion

Converting between raster and vector formats allows you to widen the applicability of data, and need not lead to data degradation.

8.1.5 What's Next?

Now that we have the results of the terrain analysis available in vector format, they can be used to solve the problem of which buildings we should consider for the residential development.

8.2 Lesson: Combining the Analyses

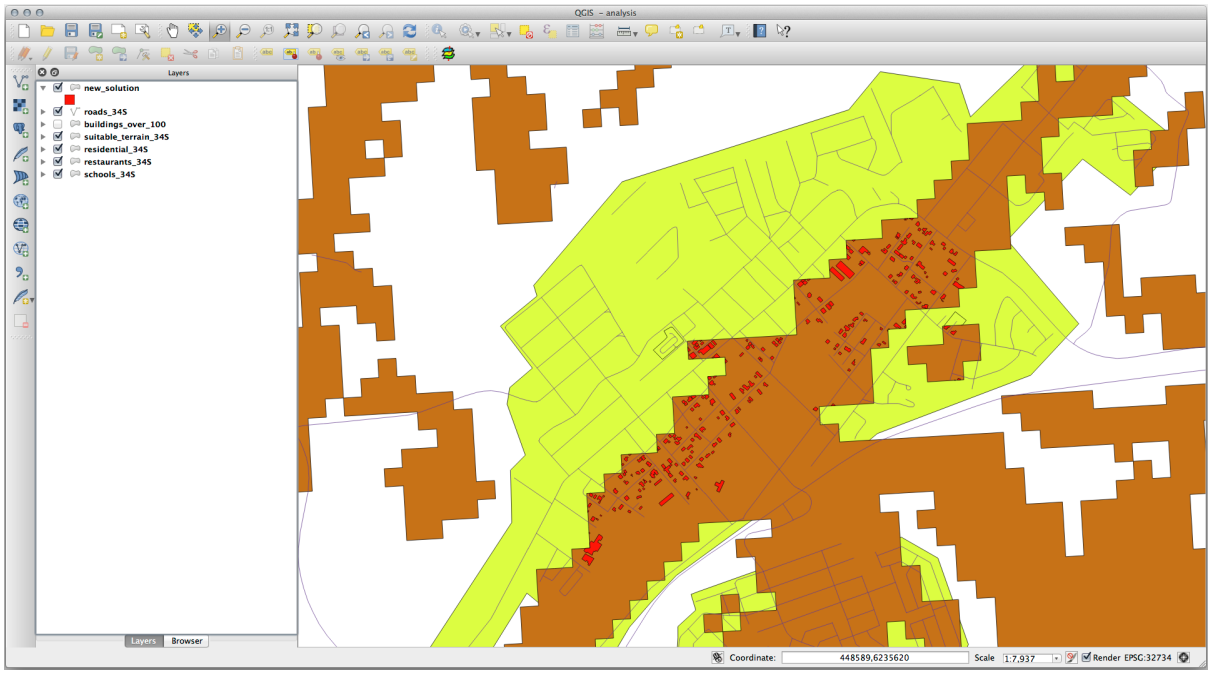
Using the vectorized results of the raster analysis will allow you to select only those buildings on suitable terrain.

The goal for this lesson: To use the vectorized terrain results to select suitable plots.

8.2.1 Try Yourself

1. Save your current map (`raster_analysis.qgs`).
2. Open the map which you created during the vector analysis earlier (you should have saved the file as `analysis.qgs`).
3. In the *Layers* panel, enable these layers:
   - *hillshade*,
   - *solution* (or *buildings_over_100*)
4. In addition to these layers, which should already be loaded in the map from when you worked on it before, also add the `suitable_terrain.shp` dataset.
5. If you are missing some layers, you should find them in `exercise_data/residential_development/`
6. Use the *Intersection* tool (*Vector* ► *Geoprocessing Tools*) to create a new vector layer called `new_solution.shp` which contains only those buildings which intersect the *suitable_terrain* layer.

You should now have a layer showing certain buildings as your solution, for example:



8.2.2 Try Yourself Inspecting the Results

Look at each of the buildings in your *new_solution* layer. Compare them with the *suitable_terrain* layer by changing the symbology for the *new_solution* layer so that it has outlines only. What do you notice about some of the buildings? Are they all suitable just because they intersect with the *suitable_terrain* layer? Why or why not? Which ones would you deem to be unsuitable?

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.

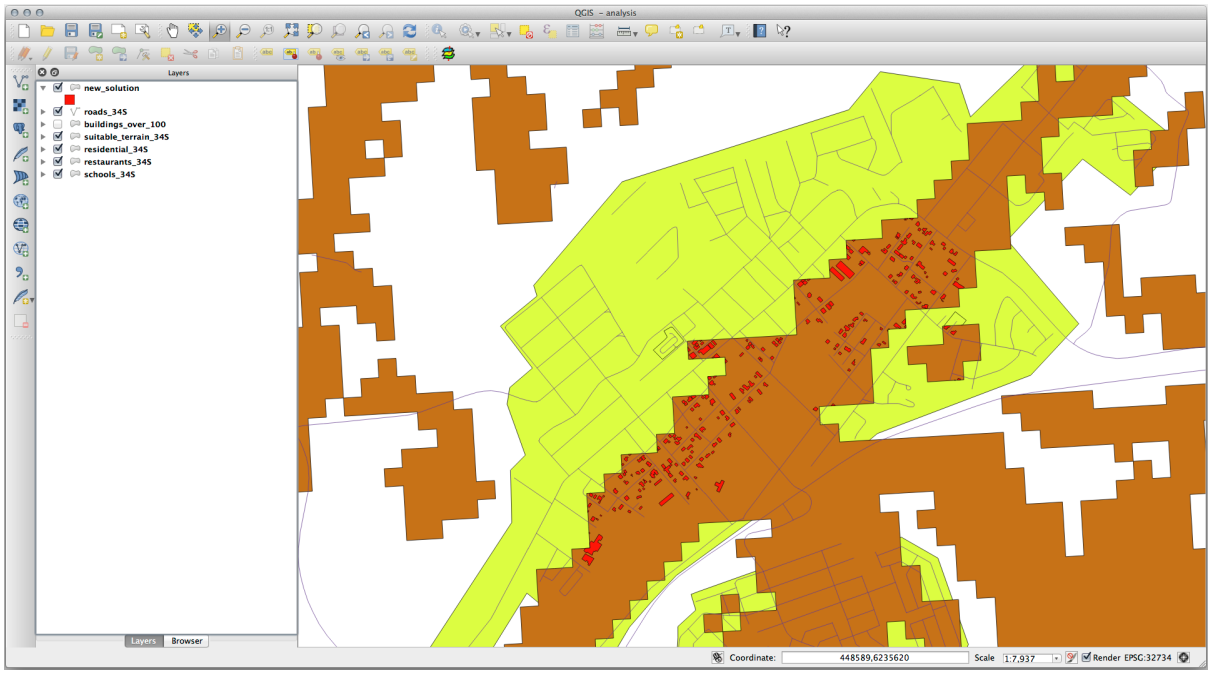
8.2.3 Try Yourself Refining the Analysis

You can see from the results that some buildings which were included were not really suitable, so we can now refine the analysis.

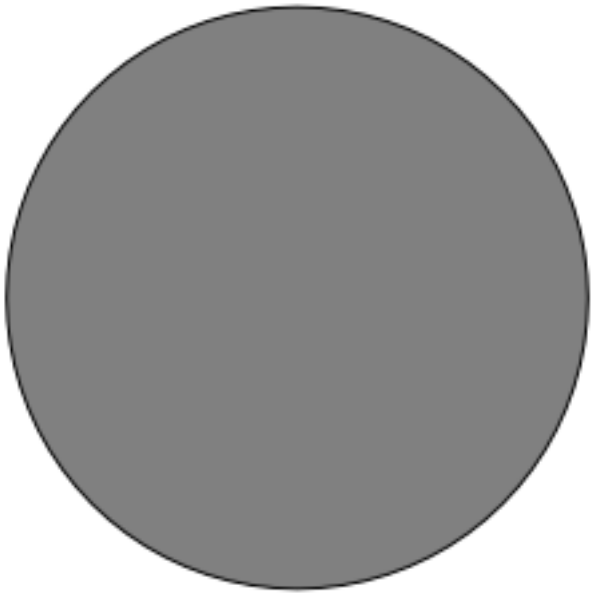
We want to ensure that our analysis returns only those buildings which fall entirely within the *suitable_terrain* layer. How would you achieve this? Use one or more Vector Analysis tools and remember that our buildings are all over 100m squared in size.

Answer

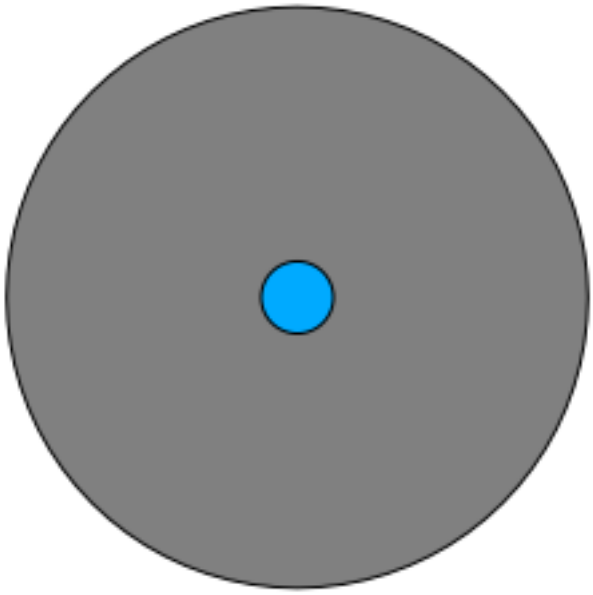
At the moment, your analysis should look something like this:



Consider a circular area, continuous for 100 meters in all directions.



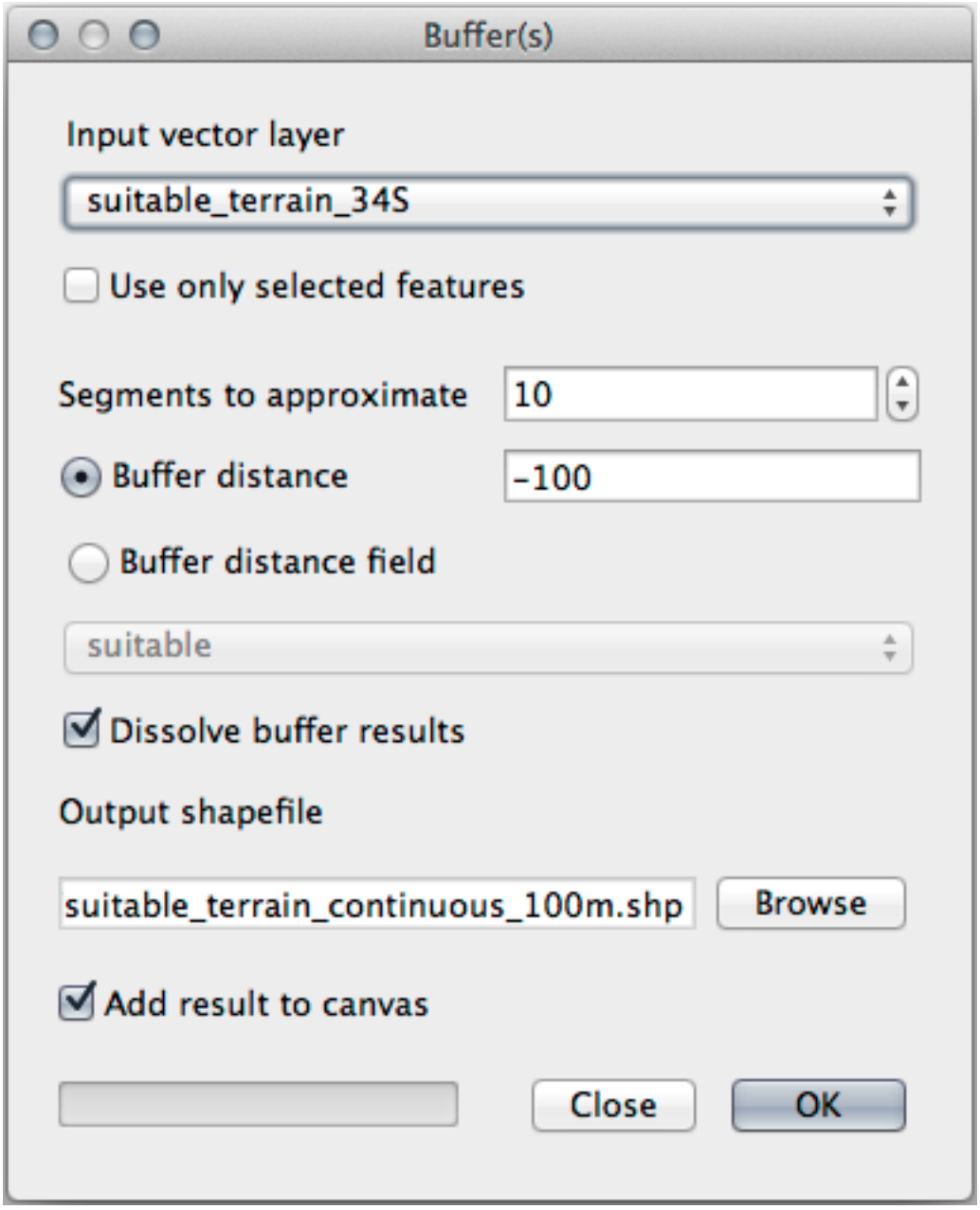
If it is greater than 100 meters in radius, then subtracting 100 meters from its size (from all directions) will result in a part of it being left in the middle.



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.

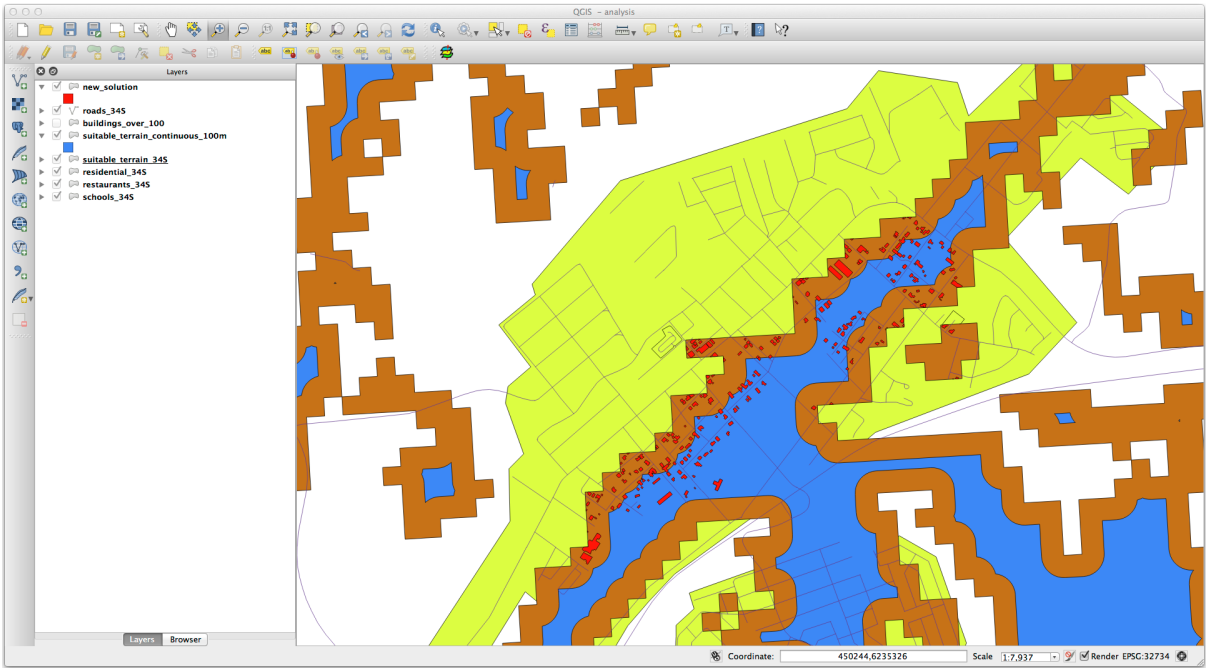
To demonstrate:

1. Go to *Vector* ► *Geoprocessing Tools* ► *Buffer(s)* to open the Buffer(s) dialog.
2. Set it up like this:

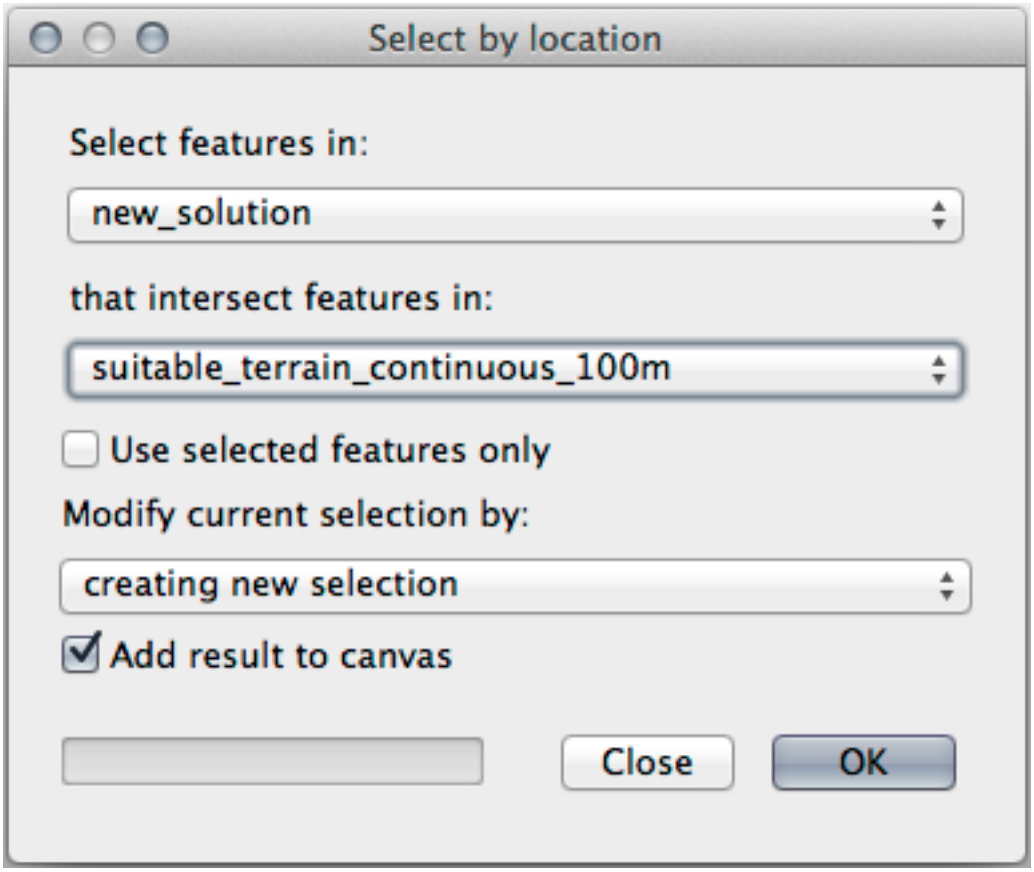


3. Use the *suitable_terrain* layer with `10` segments and a buffer distance of `-100`. (The distance is automatically in meters because your map is using a projected CRS.)
4. Save the output in `exercise_data/residential_development/` as `suitable_terrain_continuous100m.shp`.
5. If necessary, move the new layer above your original *suitable_terrain* layer.

   Your results will look like something like this:

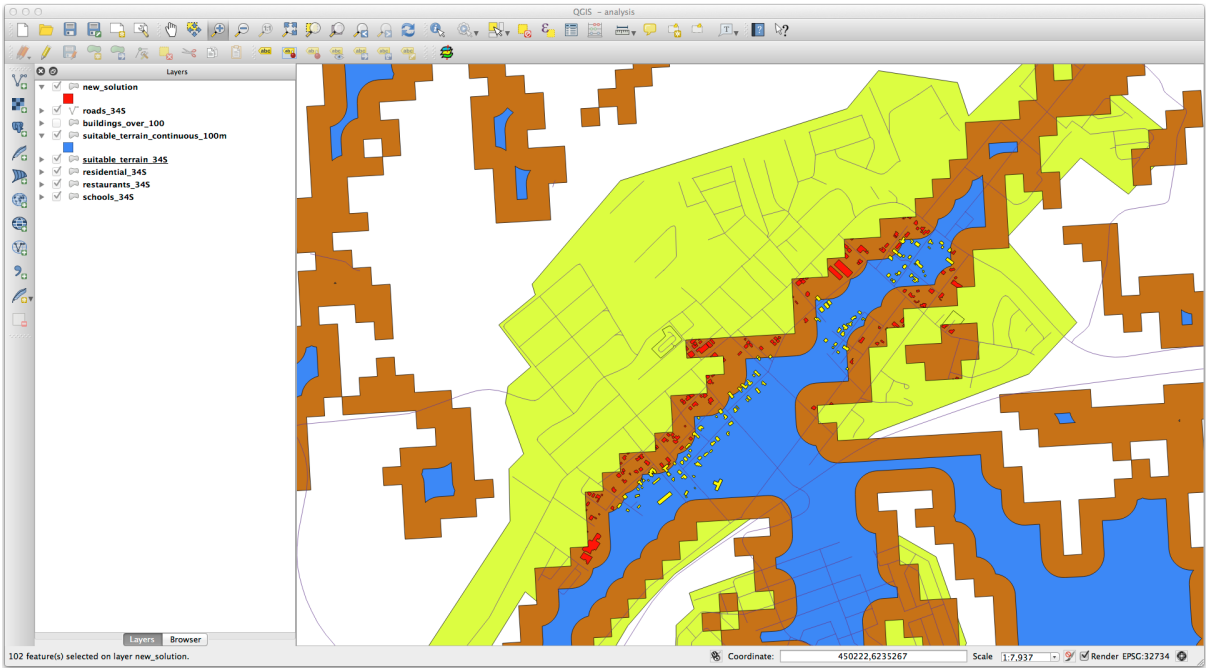


6. Now use the *Select by Location* tool (*Vector* ► *Research Tools* ► *Select by location*).
7. Set up like this:



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 In Conclusion

You have now answered the original research question, and can offer an opinion (with reasons, backed by analysis) for a recommendation regarding which property to develop.

8.2.5 What's Next?

Next you will present these results as part of your second assignment.

8.3 Zuordnung

Using the print layout, make a new map representing the results of your analysis. Include these layers:

- *places* (with labels),
- *hillshade*,
- *solution* (or *new_solution*),
- *roads* and
- either *aerial_photos* or *DEM*.

Write a short explanatory text to accompany it. Include in this text the criteria that were used in considering a house for purchase and subsequent development, as well as explaining your recommendations for which buildings are suitable.

8.4 Lesson: Supplementary Exercise

In this lesson, you will be guided through a complete GIS analysis in QGIS.

Bemerkung: Lesson developed by Linfiniti Consulting (South Africa) and Siddique Motala (Cape Peninsula University of Technology)

8.4.1 Problem Statement

You are tasked with finding areas in and around the Cape Peninsula that are suitable habitats for a rare fynbos plant species. The extent of your area of investigation covers Cape Town and the Cape Peninsula between Melkbosstrand in the north and Strand in the south. Botanists have provided you with the following preferences exhibited by the species in question:

- It grows on east facing slopes
- It grows on slopes with a gradient between 15% and 60%
- It grows in areas that have a total annual rainfall of > 1000 mm
- It will only be found at least 250 m away from any human settlement
- The area of vegetation in which it occurs should be at least 6000 m^2 in area

As a student at the University, you have agreed to search for the plant in four different suitable areas of land. You want those four suitable areas to be the ones that are closest to the University of Cape Town where you live. Use your GIS skills to determine where you should go to look.

8.4.2 Solution Outline

The data for this exercise can be found in the `exercise_data/more_analysis` folder.

You are going to find the four suitable areas that are closest to the University of Cape Town.

The solution will involve:

1. Analyzing a DEM raster layer to find the east facing slopes and the slopes with the correct gradients
2. Analyzing a rainfall raster layer to find the areas with the correct amount of rainfall
3. Analyzing a zoning vector layer to find areas that are away from human settlement and are of the correct size

8.4.3 Follow Along: Setting up the Map

1. Click on the Current CRS button in the lower right corner of the screen. Under the *CRS* tab of the dialog that appears, use the „Filter“ tool to search for „33S“. Select the entry *WGS 84 / UTM zone 33S* (with EPSG code `32733`).
2. Klicken Sie auf *OK*
3. Save the project file by clicking on the Save Project toolbar button, or use the *Project* ► *Save As...* menu item.

   Save it in a new directory called `Rasterprac`, that you should create somewhere on your computer. You will save whatever layers you create in this directory as well. Save the project as `your_name_fynbos.qgs`.

8.4.4 Loading Data into the Map

In order to process the data, you will need to load the necessary layers (street names, zones, rainfall, DEM, districts) into the map canvas.

For vectors...

1. Click on the Open Data Source Manager button in the *Data Source Manager Toolbar*, and enable the *Vector* tab in the dialog that appears, or use the *Layer* ► *Add Layer* ► *Add Vector Layer…* menu item
2. Ensure that *File* is selected
3. Click on the … button to browse for vector dataset(s)
4. In the dialog that appears, open the `exercise_data/more_analysis/Streets` directory
5. Select the file `Street_Names_UTM33S.shp`
6. Click *Open*.

   The dialog closes and shows the original dialog, with the file path specified in the text field next to *Vector dataset(s)*. This allows you to ensure that the correct file is selected. It is also possible to enter the file path in this field manually, should you wish to do so.
7. Click *Add*. The vector layer will be loaded into your map. Its color is automatically assigned. You will change it later.
8. Rename the layer to `Streets`
   1. Right-click on it in the *Layers* panel (by default, the pane along the left-hand side of the screen)
   2. Click *Rename* in the dialog that appears and rename it, pressing the `Enter` key when done
9. Repeat the vector adding process, but this time select the `Generalised_Zoning_Dissolve_UTM33S.shp` file in the `Zoning` directory.
10. Rename it to `Zoning`.
11. Load also the vector layer `admin_boundaries/Western_Cape_UTM33S.shp` into your map.
12. Rename it to `Districts`.

For rasters...

1. Click on the Open Data Source Manager button and enable the *Raster* tab in the dialog that appears, or use the *Layer* ► *Add Layer* ► *Add Raster Layer…* menu item
2. Ensure that *File* is selected
3. Navigate to the appropriate file, select it, and click *Open*
4. Do this for each of the following two raster files, `DEM/SRTM.tif` and `rainfall/reprojected/rainfall.tif`
5. Rename the SRTM raster to `DEM` and the rainfall raster to `Rainfall` (with an initial capital)

8.4.5 Changing the layer order

Click and drag layers up and down in the *Layers* panel to change the order they appear in on the map so that you can see as many of the layers as possible.

Now that all the data is loaded and properly visible, the analysis can begin. It is best if the clipping operation is done first. This is so that no processing power is wasted on computing values in areas that are not going to be used anyway.

8.4.6 Find the Correct Districts

Due to the aforementioned area of investigation, we need to limit our districts to the following ones:

- `Bellville`
- `Cape`
- `Goodwood`
- `Kuils River`
- `Mitchells Plain`
- `Simon Town`
- `Wynberg`

1. Right-click on the `Districts` layer in the *Layers* panel.
2. In the menu that appears, select the *Filter…* menu item. The *Query Builder* dialog appears.
3. You will now build a query to select only the candidate districts:
    1. In the *Fields* list, double-click on the `NAME_2` field to make it appear in the *SQL where clause* text field below
    2. Click the *IN* button to append it to the SQL query
    3. Open the brackets
    4. Click the *All* button below the (currently empty) *Values* list.

       After a short delay, this will populate the *Values* list with the values of the selected field (`NAME_2`).
    5. Double-click the value `Bellville` in the *Values* list to append it to the SQL query.
    6. Add a comma and double-click to add `Cape` district
    7. Repeat the previous step for the remaining districts
    8. Close the brackets

       The final query should be (the order of the districts in the brackets does not matter):

```
"NAME_2" in ('Bellville', 'Cape', 'Goodwood', 'Kuils River',
             'Mitchells Plain', 'Simon Town', 'Wynberg')
```

Bemerkung: You can also use the `OR` operator; the query would look like this:

```
"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. Click *OK* twice.

       The districts shown in your map are now limited to those in the list above.

8.4.7 Clip the Rasters

Now that you have an area of interest, you can clip the rasters to this area.

1. Open the clipping dialog by selecting the menu item *Raster* ► *Extraction* ► *Clip Raster by Mask Layer…*
2. In the *Input layer* dropdown list, select the `DEM` layer
3. In the *Mask layer* dropdown list, select the `Districts` layer
4. Scroll down and specify an output location in the *Clipped (mask)* text field by clicking the … button and choosing *Save to File…*
    1. Navigate to the `Rasterprac` directory
    2. Enter a file name - `DEM_clipped.tif`
    3. Save
5. Make sure that *Open output file after running algorithm* is checked
6. Click *Run*

    After the clipping operation has completed, leave the *Clip Raster by Mask Layer* dialog open, to be able to reuse the clipping area
7. Select the `Rainfall` raster layer in the *Input layer* dropdown list and save your output as `Rainfall_clipped.tif`
8. Do not change any other options. Leave everything the same and click *Run*.
9. After the second clipping operation has completed, you may close the *Clip Raster by Mask Layer* dialog
10. Save the map

Align the rasters

For our analysis we need the rasters to have the same CRS and they have to be aligned.

First we change the resolution of our rainfall data to 30 meters (pixel size):

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. Make sure that *Open output file after running algorithm* is checked

Then we align the DEM:

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. Make sure that *Open output file after running algorithm* is checked

In order to properly see what's going on, the symbology for the layers needs to be changed.

8.4.8 Changing the symbology of vector layers

1. In the *Layers* panel, right-click on the *Streets* layer
2. Select *Properties* from the menu that appears
3. Switch to the *Symbology* tab in the dialog that appears
4. Click on the *Line* entry in the top widget
5. Select a symbol in the list below or set a new one (color, transparency, …)
6. Click *OK* to close the *Layer Properties* dialog. This will change the rendering of the *Streets* layer.
7. Follow a similar process for the *Zoning* layer and choose an appropriate color for it

8.4.9 Changing the symbology of raster layers

Raster layer symbology is somewhat different.

1. Open the *Properties* dialog for the *Rainfall30* raster layer
2. Switch to the *Symbology* tab. You'll notice that this dialog is very different from the version used for vector layers.
3. Expand *Min/Max Value Settings*
4. Ensure that the button *Mean +/- standard deviation* is selected
5. Make sure that the value in the associated box is `2.00`
6. For *Contrast enhancement*, make sure it says *Stretch to MinMax*
7. For *Color gradient*, change it to *White to Black*
8. Klicken Sie auf *OK*

   The `Rainfall30` raster, if visible, should change colors, allowing you to see different brightness values for each pixel
9. Repeat this process for the `DEM30` layer, but set the standard deviations used for stretching to `4.00`

8.4.10 Clean up the map

1. Remove the original `Rainfall` and `DEM` layers, as well as `Rainfall_clipped` and `DEM_clipped` from the *Layers* panel:
   - Right-click on these layers and select *Remove*.

   Bemerkung: This will not remove the data from your storage device, it will merely take it out of your map.
2. Save the map
3. You can now hide the vector layers by unchecking the box next to them in the *Layers* panel. This will make the map render faster and will save you some time.

8.4.11 Create the hillshade

In order to create the hillshade, you will need to use an algorithm that was written for this purpose.

1. In the *Layers* panel, ensure that `DEM30` is the active layer (i.e., it is highlighted by having been clicked on)
2. Click on the *Raster* ► *Analysis* ► *Hillshade…* menu item to open the *Hillshade* dialog
3. Scroll down to *Hillshade* and save the output in your `Rasterprac` directory as `hillshade.tif`
4. Make sure that *Open output file after running algorithm* is checked
5. Click *Run*
6. Wait for it to finish processing.

The new `hillshade` layer has appeared in the *Layers* panel.

1. Right-click on the `hillshade` layer in the *Layers* panel and bring up the *Properties* dialog
2. Click on the *Transparency* tab and set the *Global Opacity* slider to `20%`
3. Klicken Sie auf *OK*
4. Note the effect when the transparent hillshade is superimposed over the clipped DEM. You may have to change the order of your layers, or click off the `Rainfall30` layer in order to see the effect.

8.4.12 Slope

1. Click on the *Raster* ► *Analysis* ► *Slope…* menu item to open the *Slope* algorithm dialog
2. Select `DEM30` as *Input layer*
3. Check *Slope expressed as percent instead of degrees*. Slope can be expressed in different units (percent or degrees). Our criteria suggest that the plant of interest grows on slopes with a gradient between 15% and 60%. So we need to make sure our slope data is expressed as a percent.
4. Specify an appropriate file name and location for your output.
5. Make sure that *Open output file after running algorithm* is checked
6. Click *Run*

The slope image has been calculated and added to the map. As usual, it is rendered in grayscale. Change the symbology to a more colorful one:

1. Open the layer *Properties* dialog (as usual, via the right-click menu of the layer)
2. Click on the *Symbology* tab
3. Where it says *Singleband gray* (in the *Render type* dropdown menu), change it to *Singleband pseudocolor*
4. Choose *Mean +/- standard deviation x* for *Min / Max Value Settings* with a value of `2.0`
5. Select a suitable *Color ramp*
6. Click *Run*

8.4.13 Try Yourself Aspect

Use the same approach as for calculating the slope, choosing *Aspect…* in the *Raster* ► *Analysis* menu.

Remember to save the project periodically.

8.4.14 Reclassifying rasters

1. Choose *Raster* ► *Raster calculator…*
2. Specify your `Rasterprac` directory as the location for the *Output layer* (click on the … button), and save it as `slope15_60.tif`
3. Ensure that the *Open output file after running algorithm* box is selected.

   In the *Raster bands* list on the left, you will see all the raster layers in your *Layers* panel. If your Slope layer is called *slope*, it will be listed as `slope@1`. Indicating band 1 of the slope raster.
4. The slope needs to be between `15` and `60` degrees.

   Using the list items and buttons in the interface, build the following expression:

   ```
   (slope@1 > 15) AND (slope@1 < 60)
   ```
5. Set the *Output layer* field to an appropriate location and file name.
6. Klicken Sie auf *Starten*.

Now find the correct aspect (east-facing: between `45` and `135` degrees) using the same approach.

1. Build the following expression:

   ```
   (aspect@1 > 45) AND (aspect@1 < 135)
   ```

You will know it worked when all of the east-facing slopes are white in the resulting raster (it's almost as if they are being lit by the morning sunlight).

Find the correct rainfall (greater than `1000` mm) the same way. Use the following expression:

```
Rainfall30@1 > 1000
```

Now that you have all three criteria each in separate rasters, you need to combine them to see which areas satisfy all the criteria. To do so, the rasters will be multiplied with each other. When this happens, all overlapping pixels with a value of `1` will retain the value of `1` (i.e. the location meets the criteria), but if a pixel in any of the three rasters has the value of `0` (i.e. the location does not meet the criteria), then it will be `0` in the result. In this way, the result will contain only the overlapping areas that meet all of the appropriate criteria.

8.4.15 Combining rasters

1. Open the *Raster Calculator* (*Raster* ► *Raster Calculator…*)
2. Build the following expression (with the appropriate names for your layers):

   ```
   [aspect45_135] * [slope15_60] * [rainfall_1000]
   ```
3. Set the output location to the `Rasterprac` directory
4. Name the output raster `aspect_slope_rainfall.tif`
5. Ensure that *Open output file after running algorithm* is checked
6. Click *Run*

The new raster now properly displays the areas where all three criteria are satisfied.

Save the project.

The next criterion that needs to be satisfied is that the area must be 250 m away from urban areas. We will satisfy this requirement by ensuring that the areas we compute are inside rural areas, and are 250 m or more from the edge of the area. Hence, we need to find all rural areas first.

8.4.16 Finding rural areas

1. Hide all layers in the *Layers* panel
2. Unhide the Zoning vector layer
3. Right-click on it and bring up the *Attribute Table* dialog. Note the many different ways that the land is zoned here. We want to isolate the rural areas. Close the Attribute table.
4. Right-click on the Zoning layer and select *Filter…* to bring up the *Query Builder* dialog
5. Build the following query:

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

   See the earlier instructions if you get stuck.
6. Click *OK* to close the *Query Builder* dialog. The query should return one feature.

You should see the rural polygons from the Zoning layer. You will need to save these.

1. In the right-click menu for Zoning, select *Export* ► *Save Features As…*
2. Save your layer under the Rasterprac directory
3. Name the output file rural.shp
4. Klicken Sie auf *OK*
5. Save the project

Now you need to exclude the areas that are within 250m from the edge of the rural areas. Do this by creating a negative buffer, as explained below.

8.4.17 Creating a negative buffer

1. Click the menu item *Vector* ► *Geoprocessing Tools* ► *Buffer…*
2. In the dialog that appears, select the rural layer as your input vector layer (*Selected features only* should not be checked)
3. Set *Distance* to −250. The negative value means that the buffer will be an internal buffer. Make sure that the units are meters in the dropdown menu.
4. Check *Dissolve result*
5. In *Buffered*, place the output file in the Rasterprac directory, and name it rural_buffer.shp
6. Click *Save*
7. Click *Run* and wait for the processing to complete
8. Close the *Buffer* dialog.

   Make sure that your buffer worked correctly by noting how the rural_buffer layer is different from the rural layer. You may need to change the drawing order in order to observe the difference.
9. Remove the rural layer
10. Save the project

Now you need to combine your `rural_buffer` vector layer with the `aspect_slope_rainfall` raster. To combine them, we will need to change the data format of one of the layers. In this case, you will vectorize the raster, since vector layers are more convenient when we want to calculate areas.

8.4.18 Vectorizing the raster

1. Click on the menu item *Raster* ► *Conversion* ► *Polygonize (Raster to Vector)…*
2. Select the `aspect_slope_rainfall` raster as *Input layer*
3. Set *Name of the field to create* to `suitable` (the default field name is `DN` - Digital number data)
4. Save the output. Under *Vectorized*, select *Save file as*. Set the location to `Rasterprac` and name the file `aspect_slope_rainfall_all.shp`.
5. Ensure that *Open output file after running algorithm* is checked
6. Click *Run*
7. Close the dialog when processing is complete

All areas of the raster have been vectorized, so you need to select only the areas that have a value of `1` in the `suitable` field. (Digital Number.

1. Open the *Query Builder* dialog (right-click - *Filter…*) for the new vector layer
2. Build this query:

```
"suitable" = 1
```

3. Klicken Sie auf *OK*
4. After you are sure the query is complete (and only the areas that meet all three criteria, i.e. with a value of `1` are visible), create a new vector file from the results, using the *Export* ► *Save Features As…* in the layer's right-click menu
5. Save the file in the `Rasterprac` directory
6. Name the file `aspect_slope_rainfall_1.shp`
7. Remove the `aspect_slope_rainfall_all` layer from your map
8. Save your project

When we use an algorithm to vectorize a raster, sometimes the algorithm yields what is called „Invalid geometries", i.e. there are empty polygons, or polygons with mistakes in them, that will be difficult to analyze in the future. So, we need to use the „Fix Geometry" tool.

8.4.19 Fixing geometry

1. In the *Processing Toolbox*, search for „Fix geometries", and *Execute…* it
2. For the *Input layer*, select `aspect_slope_rainfall_1`
3. Under *Fixed geometries*, select *Save file as*, and save the output to `Rasterprac` and name the file `fixed_aspect_slope_rainfall.shp`.
4. Ensure that *Open output file after running algorithm* is checked
5. Click *Run*
6. Close the dialog when processing is complete

Now that you have vectorized the raster, and fixed the resulting geometry, you can combine the aspect, slope, and rainfall criteria with the distance from human settlement criteria by finding the intersection of the `fixed_aspect_slope_rainfall` layer and the `rural_buffer` layer.

8.4.20 Determining the Intersection of vectors

1. Click the menu item *Vector* ► *Geoprocessing Tools* ► *Intersection…*
2. In the dialog that appears, select the `rural_buffer` layer as *Input layer*
3. For the *Overlay layer*, select the `fixed_aspect_slope_rainfall` layer
4. In *Intersection*, place the output file in the `Rasterprac` directory
5. Name the output file `rural_aspect_slope_rainfall.shp`
6. Click *Save*
7. Click *Run* and wait for the processing to complete
8. Close the *Intersection* dialog.

   Make sure that your intersection worked correctly by noting that only the overlapping areas remain.
9. Save the project

The next criteria on the list is that the area must be greater than `6000` m^2. You will now calculate the polygon areas in order to identify the areas that are the appropriate size for this project.

8.4.21 Calculating the area for each polygon

1. Open the new vector layer's right-click menu
2. Select *Open attribute table*
3. Click the Toggle editing button in the top left corner of the table, or press `Ctrl+e`
4. Click the Open field calculator button in the toolbar along the top of the table, or press `Ctrl+i`
5. In the dialog that appears, make sure that *Create new field* is checked, and set the *Output field name* to `area` The output field type should be a decimal number (real). Set *Precision* to `1` (one decimal).
6. In the *Expression* area, type:

   ```
   $area
   ```

   This means that the field calculator will calculate the area of each polygon in the vector layer and will then populate a new integer column (called `area`) with the computed value.
7. Klicken Sie auf *OK*
8. Do the same thing for another new field called `id`. In *Field calculator expression*, type:

   ```
   $id
   ```

   This ensures that each polygon has a unique ID for identification purposes.
9. Click Toggle editing again, and save your edits if prompted to do so

8.4.22 Selecting areas of a given size

Now that the areas are known:

1. Build a query (as usual) to select only the polygons that are larger than `6000` m^2. The query is:

```
"area" > 6000
```

2. Save the selection in the `Rasterprac` directory as a new vector layer called `suitable_areas.shp`.

You now have the suitable areas that meet all of the habitat criteria for the rare fynbos plant, from which you will pick the four areas that are nearest to the University of Cape Town.

8.4.23 Digitize the University of Cape Town

1. Create a new vector layer in the `Rasterprac` directory as before, but this time, use *Point* as *Geometry type* and name it `university.shp`
2. Ensure that it is in the correct CRS (`Project CRS:EPSG:32733 - WGS 84 / UTM zone 33S`)
3. Finish creating the new layer (click *OK*)
4. Hide all layers except the new `university` layer and the `Streets` layer.
5. Add a background map (OSM):
    1. Go to the *Browser* panel and navigate to *XYZ Tiles* ► *OpenStreetMap*
    2. Drag and drop the `OpenStreetMap` entry to the bottom of the *Layers* panel

    Using your internet browser, look up the location of the University of Cape Town. Given Cape Town's unique topography, the university is in a very recognizable location. Before you return to QGIS, take note of where the university is located, and what is nearby.
6. Ensure that the `Streets` layer clicked on, and that the `university` layer is highlighted in the *Layers* panel
7. Navigate to the *View* ► *Toolbars* menu item and ensure that *Digitizing* is selected. You should then see a toolbar icon with a pencil on it (Toggle editing). This is the *Toggle Editing* button.
8. Click the *Toggle editing* button to enter *edit mode*. This allows you to edit a vector layer
9. Click the Add Point Feature button, which should be nearby the Toggle editing button
10. With the *Add feature* tool activated, left-click on your best estimate of the location of the University of Cape Town
11. Supply an arbitrary integer when asked for the `id`
12. Klicken Sie auf *OK*
13. Click the Save Layer Edits button
14. Click the *Toggle editing* button to stop your editing session
15. Save the project

8.4.24 Find the locations that are closest to the University of Cape Town

1. Go to the *Processing Toolbox*, locate the *Join Attributes by Nearest* algorithm (*Vector general* ► *Join Attributes by Nearest*) and execute it
2. *Input layer* should be `university`, and *Input layer 2* `suitable_areas`
3. Set an appropriate output location and name (*Joined layer*)
4. Set the *Maximum nearest neighbors* to `4`
5. Ensure that *Open output file after running algorithm* is checked
6. Leave the rest of the parameters with their default values
7. Click *Run*

The resulting point layer will contain four features - they will all have the location of the university and its attributes, and in addition, the attributes of the nearby suitable areas (including the `id`), and the distance to that location.

1. Open the attribute table of the result of the join
2. Note the `id` of the four nearest suitable areas, and then close the attribute table
3. Open the attribute table of the `suitable_areas` layer
4. Build a query to select the four suitable areas closest to the university (selecting them using the `id` field)

This is the final answer to the research question.

For your submission, create a fully labeled layout that includes the semi-transparent hillshade layer over an appealing raster of your choice (such as the DEM or the slope raster, for example). Also include the university and the `suitable_areas` layer, with the four suitable areas that are closest to the university highlighted. Follow all the best practices for cartography in creating your output map.

KAPITEL 9

Module: Erweiterungen

Erweiterungen bieten Ihnen die Möglichkeiten den Funktionsumfang von QGIS zu vergrößern. In diesem Modul wird Ihnen gezeigt, wie Erweiterungen Aktiviert und genutzt werden können.

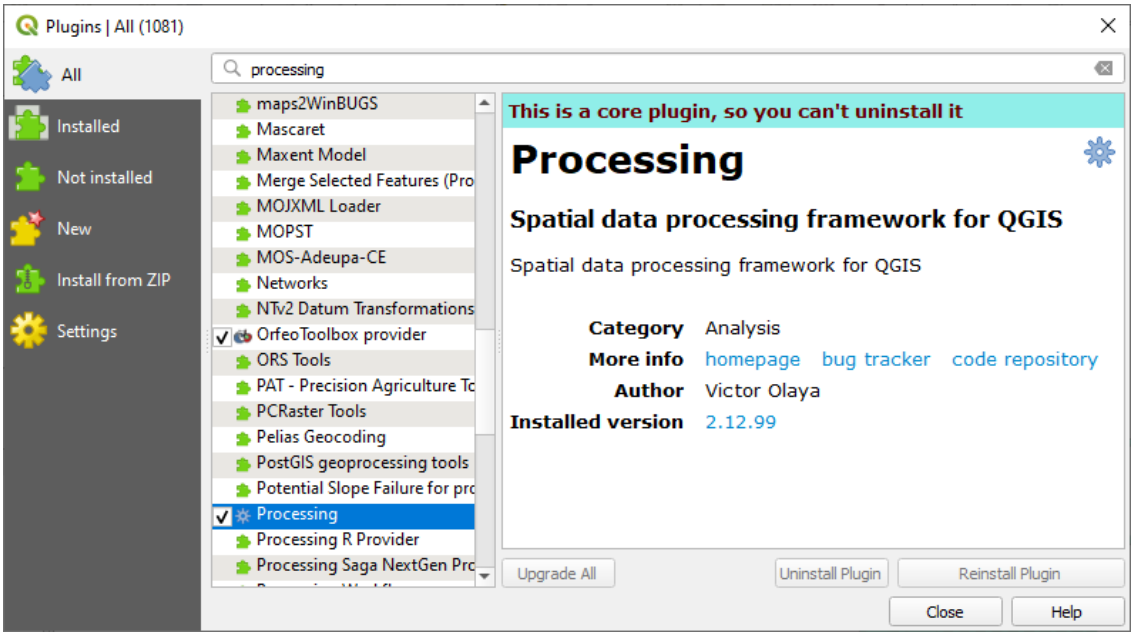
9.1 Lesson: Installing and Managing Plugins

To begin using plugins, you need to know how to download, install and activate them. To do this, you will learn how to use the *Plugin Installer* and *Plugin Manager*.

The goal for this lesson: To understand and use QGIS' plugin system.

9.1.1 Follow Along: Managing Plugins

1. To open the *Plugin Manager*, click on the menu item *Plugins* ► *Manage and Install Plugins*.
2. In the dialog that opens, find the *Processing* plugin:

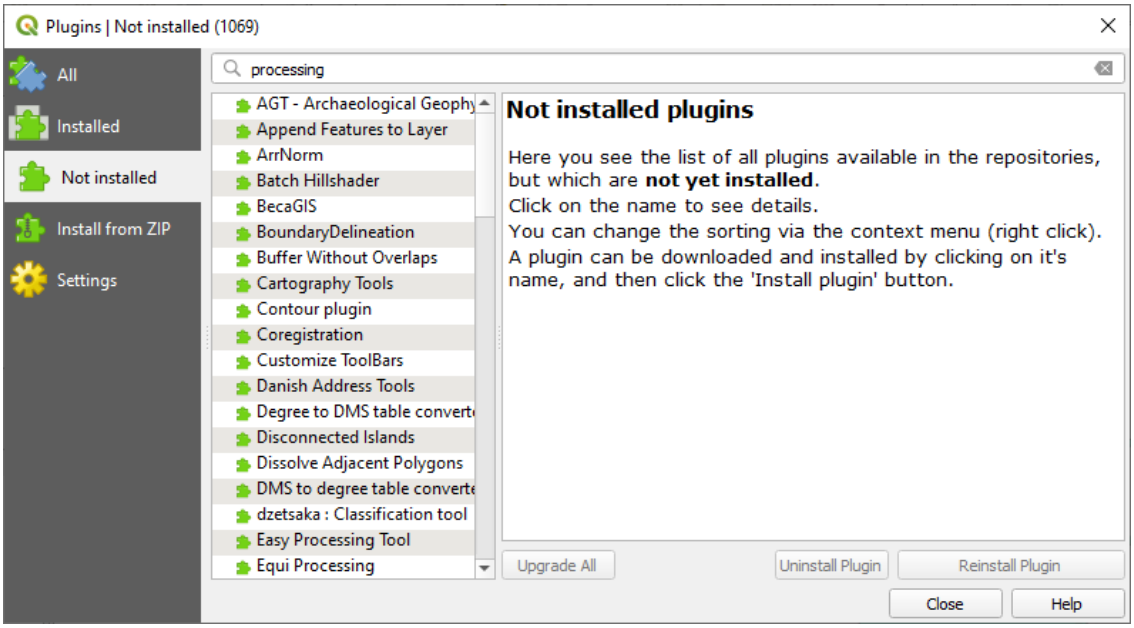


3. Click in the box next to this plugin and uncheck it to deactivate it.
4. Click *Close*.
5. Looking at the menu, you will notice that the *Processing* menu is is now gone. This means that many of the processing functions you have been using before have disappeared! For example look at the *Vector* ► and *Raster* ► menus. This is because they are part of the *Processing* plugin, which needs to be activated to use them.
6. Open the *Plugin Manager* again and reactivate the *Processing* plugin by clicking in the checkbox next to it.
7. *Close* the dialog. The *Processing* menu and functions should be available again.

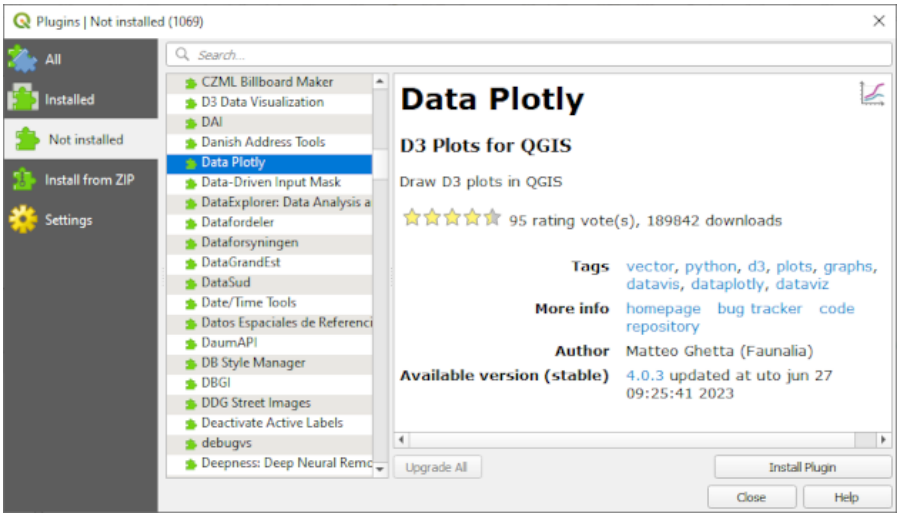
9.1.2 Follow Along: Installing New Plugins

The list of plugins that you can activate and deactivate draws from the plugins that you currently have installed. To install new plugins:

1. Select the *Not Installed* option in the *Plugin Manager* dialog. The plugins available for you to install will be listed here. This list will vary depending on your existing system setup.



2. Find information about the plugin by selecting it in the list



3. Install the one(s) you are interested in by clicking the *Install Plugin* button below the plugin information panel.

Bemerkung: if the plugin has some error it will be listed in the *Invalid* tab. You can then contact the plugin owner to fix the problem.

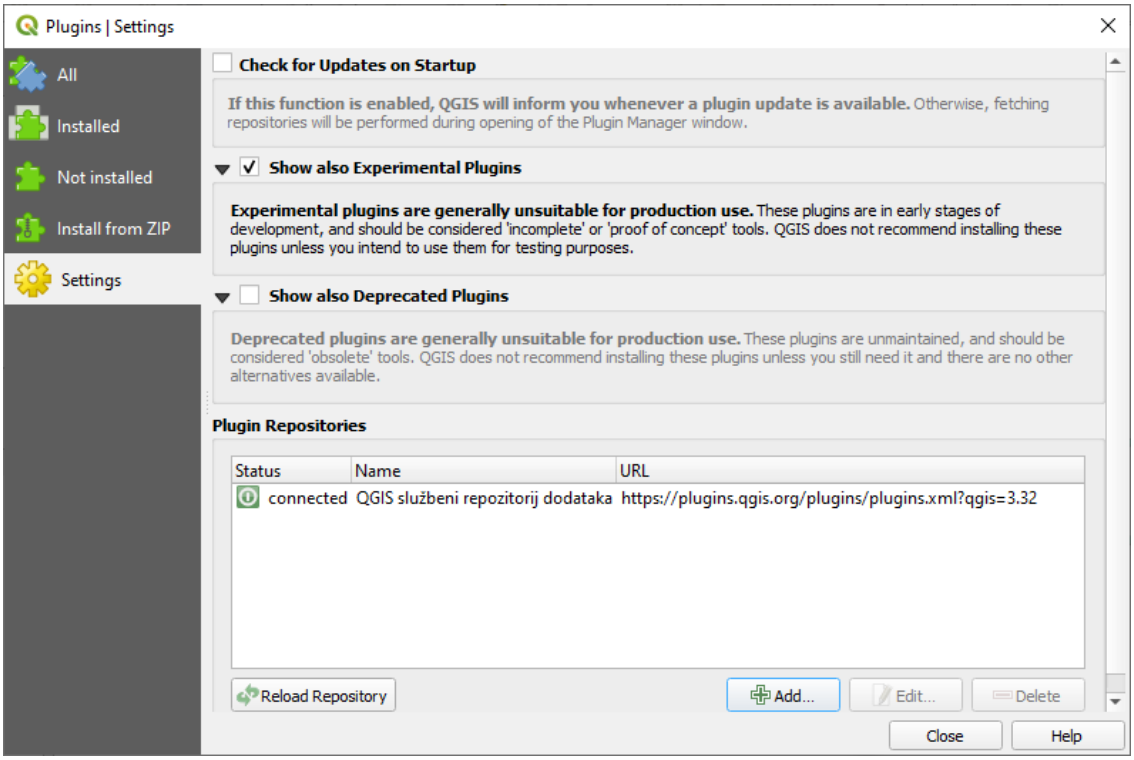
9.1.3 Follow Along: Configuring Additional Plugin Repositories

The plugins that are available to you for installation depend on which plugin *repositories* you are configured to use.

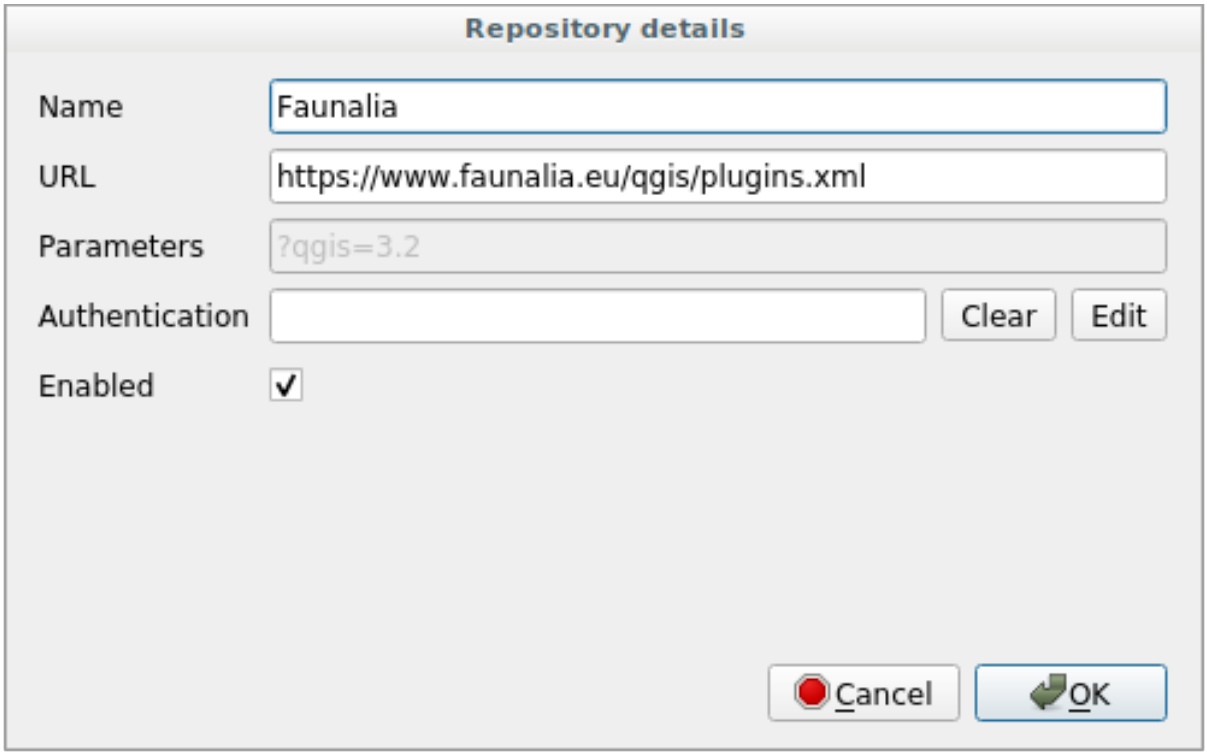
QGIS plugins are stored online in repositories. By default, only the official repository is active, meaning that you can only access plugins that are published there. Given the diversity of available tools, this repository should meet most of your needs.

It is possible, however, to try out more plugins than the default ones. First, you want to configure additional repositories. To do this:

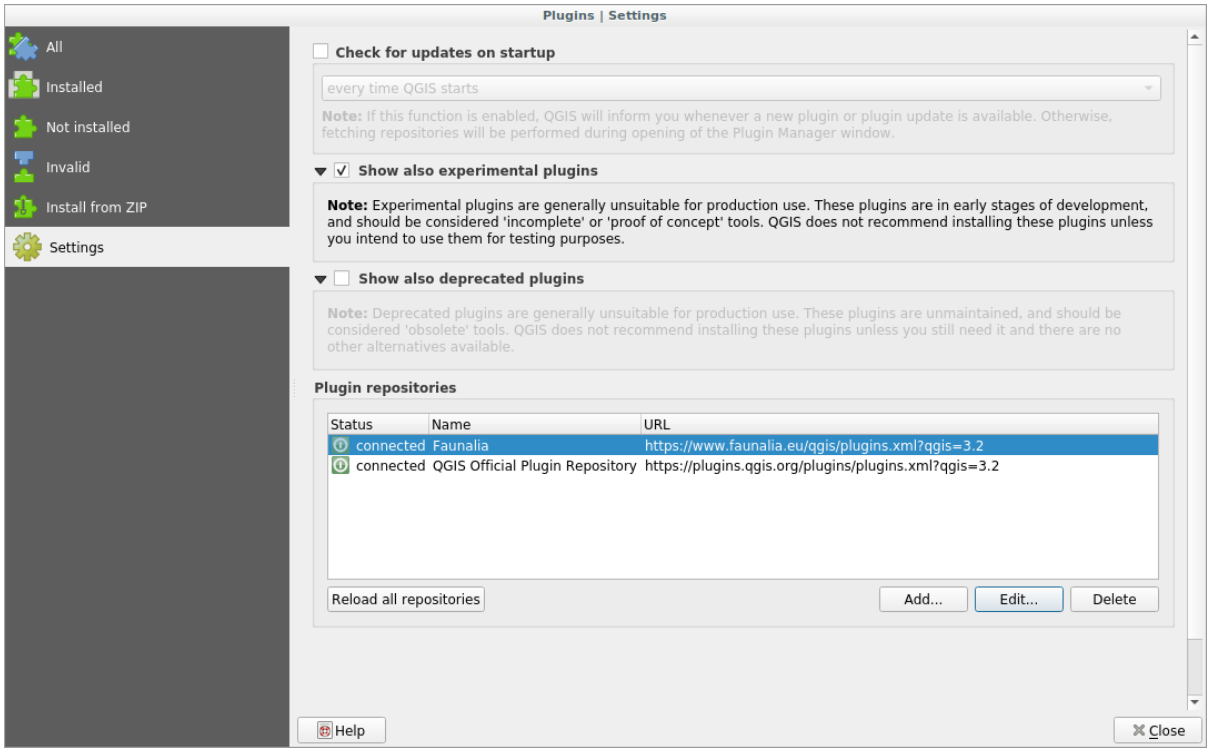
1. Open the *Settings* tab in the *Plugin Manager* dialog



2. Click *Add* to find and add a new repository.
3. Provide a Name and URL for the new repository you want to configure and make sure the *Enabled* checkbox is selected.



4. You will now see the new plugin repo listed in the list of configured Plugin Repositories



5. You can also select the option to display Experimental Plugins by selecting the *Show also experimental plugins* checkbox.
6. If you now switch back to the *Not Installed* tab, you will see that additional plugins are available for installation.
7. To install a plugin, click on it in the list and then on the *Install plugin* button.

9.1.4 In Conclusion

Installing plugins in QGIS should be straightforward and effective!

9.1.5 What's Next?

Next we'll introduce you to some useful plugins as examples.

9.2 Lesson: Nützliche QGIS Erweiterungen

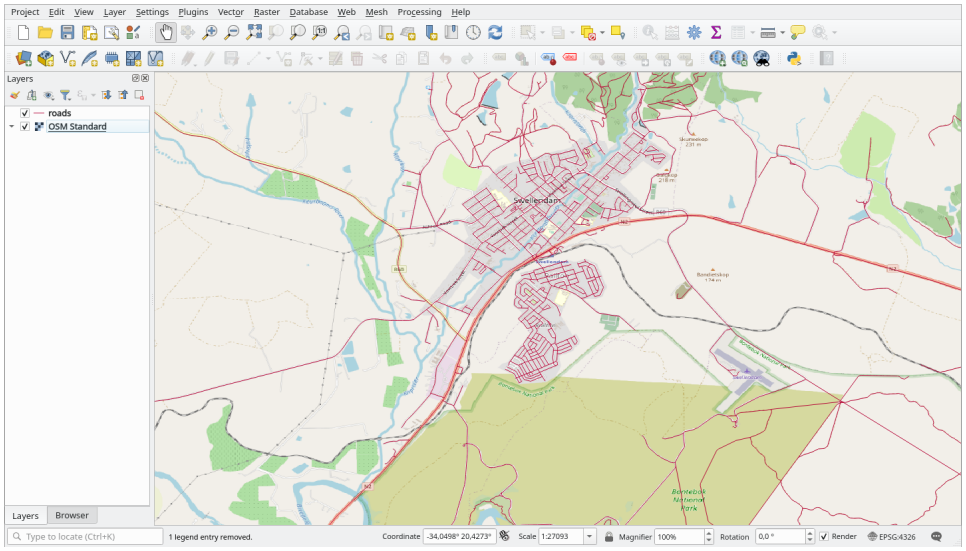
Now that you can install, enable and disable plugins, let's see how this can help you in practice by looking at some examples of useful plugins.

The goal for this lesson: To familiarize yourself with the plugin interface and get acquainted with some useful plugins.

9.2.1 Follow Along: The QuickMapServices Plugin

The QuickMapServices plugin is a simple and easy to use plugin that adds base maps to your QGIS project. It has many different options and settings. Let's start to explore some of its features.

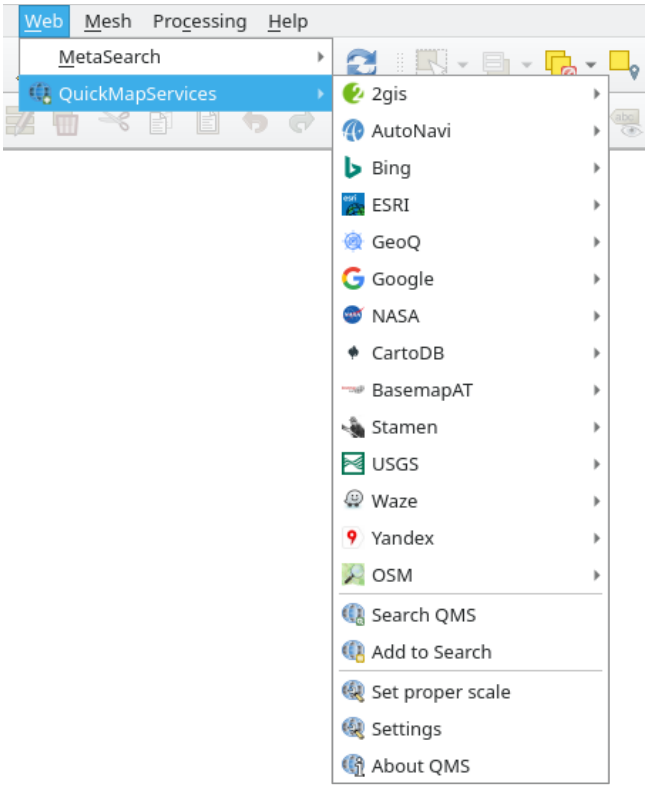
1. Start a new map and add the *roads* layer from the `training_data` Geopackage.
2. Install the **QuickMapServices** plugin.
3. Click on *Web* ► *QuickMapServices*. The first menu lists different map providers (`OSM`, `NASA`) with available maps.
4. Click on an entry and you would load the base map into your project.



Nice! But one of the main strengths of QMS is to provide access to many data providers. Let's add them.

1. Click on *Web* ► *QuickMapServices* ► *Settings*
2. Go to the *More services* tab.
3. Read carefully the message of this tab and if you agree click on the *Get Contributed pack* button.
4. Click *Save*.

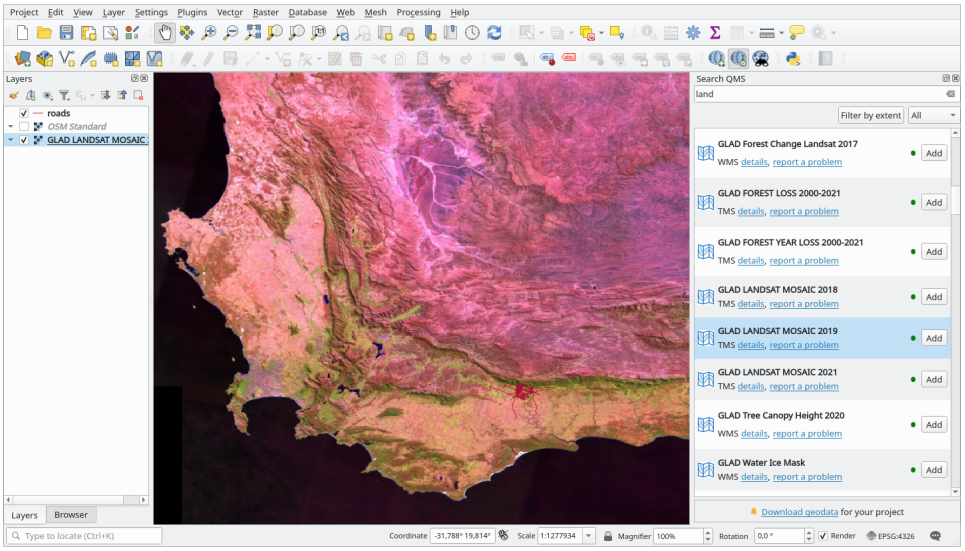
5. Reopen the *Web* ► *QuickMapServices* menu you will see that more providers are available.



6. Choose the one that best fits your needs, and load the data in the project!

It is also possible to search trough the now available data providers

1. Open the plugin's search tab by clicking on *Web* ► *QuickMapServices* ► *Search QMS*. This option of the plugin allows you to filter the available base maps by the current extent of the map canvas or using a search word.
2. Click on the *Filter by extent* and you should see one service available. If no service is found, zoom out and pan around the world (or your location) or search with a keyword.
3. Click on the *Add* button next to a returned dataset to load it.
4. The base map will be loaded and you will have a background for the map.



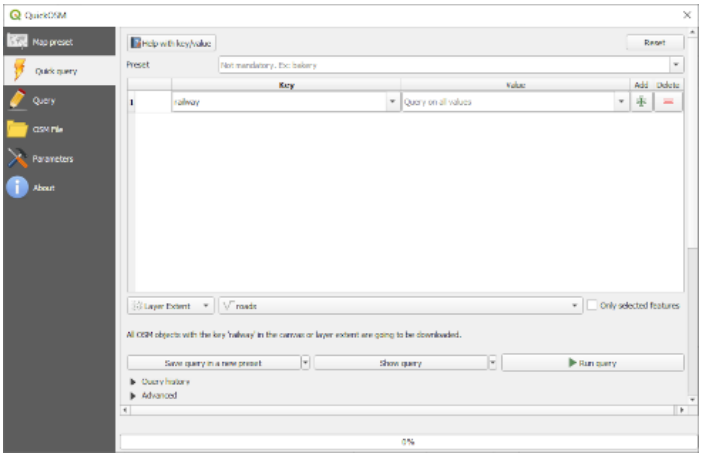
9.2.2 Follow Along: The QuickOSM Plugin

With an incredible simple interface, the QuickOSM plugin allows you to download OpenStreetMap data.

1. Start a new empty project and add the *roads* layer from the `training_data` GeoPackage.
2. Install the **QuickOSM** plugin. The plugin adds two new buttons in the QGIS Toolbar and is accessible in the *Vector* ► *QuickOSM* menu.
3. Open the QuickOSM dialog. The plugin has many different tabs: we will use the *Quick Query* one.
4. You can download specific features by selecting a generic *Key* or be more specific and choose a specific *Key* and *Value* pair.

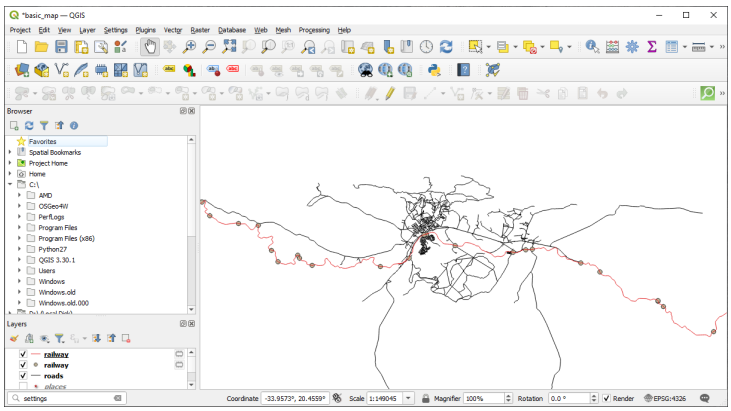
   Tipp: if you are not familiar with the *Key* and *Value* system, click on the *Help with key/value* button. It will open a web page with a complete description of this concept of OpenStreetMap.

5. Look for *railway* in the *Key* menu and let the *Value* be empty: so we are downloading all the *railway* features without specifying any values.
6. Select *Layer Extent* in the next drop-down menu and choose *roads*.
7. Click on the *Run query* button.



After some seconds the plugin will download all the features tagged in OpenStreetMap as `railway` and load them directly into the map.

Nothing more! All the layers are loaded in the legend and are shown in the map canvas.



Warnung: QuickOSM creates temporary layer when downloading the data. If you want to save them permanently, click on the icon next to the layer and choose the options you prefer. Alternatively you can open the `Advanced` menu in QuickOSM and choose where to save the data in the *Directory* menu.

9.2.3 Follow Along: The QuickOSM Query engine

The quickest way to download data from QuickOSM plugin is using the *Quick query* tab and set some small parameters. But if you need some more specific data?

If you are an OpenStreetMap query master you can use QuickOSM plugin also with your personal queries.

QuickOSM has an incredible data parser that, together with the amazing query engine of Overpass, lets you download data with your specific needs.

For example: we want to download the mountain peaks that belongs into a specific mountain area known as Dolomites.

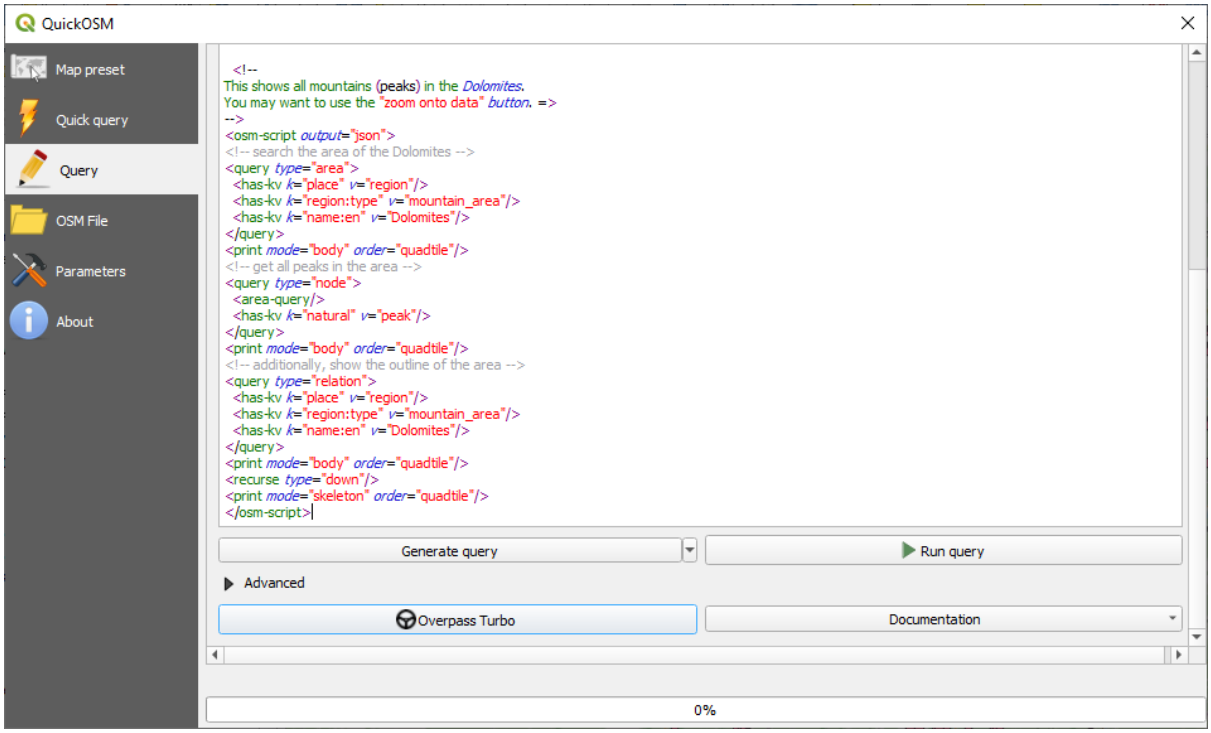
You cannot achieve this task with the *Quick query* tab, you have to be more specific and write your own query. Let's try to do this.

1. Start a new project.
2. Open the QuickOSM plugin and click on the *Query* tab.
3. Copy and paste the following code into the query canvas:

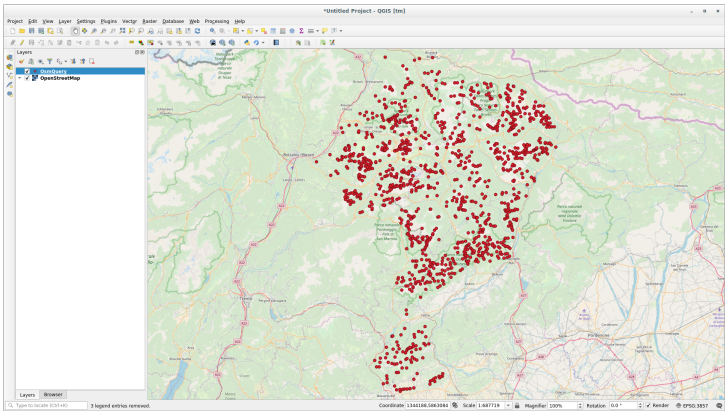
```
  <!--
This shows all mountains (peaks) in the Dolomites.
You may want to use the "zoom onto data" button. =>
-->
<osm-script output="json">
<!-- search the area of the Dolomites -->
<query type="area">
  <has-kv k="place" v="region"/>
  <has-kv k="region:type" v="mountain_area"/>
  <has-kv k="name:en" 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>
```

Bemerkung: This query is written in a `xml` like language. If you are more used to the `Overpass QL` you can write the query in this language.

4. And click on *Run Query*:



The mountain peaks layer will be downloaded and shown in QGIS:



You can write complex queries using the Overpass Query language. Take a look at some example and try to explore the query language.

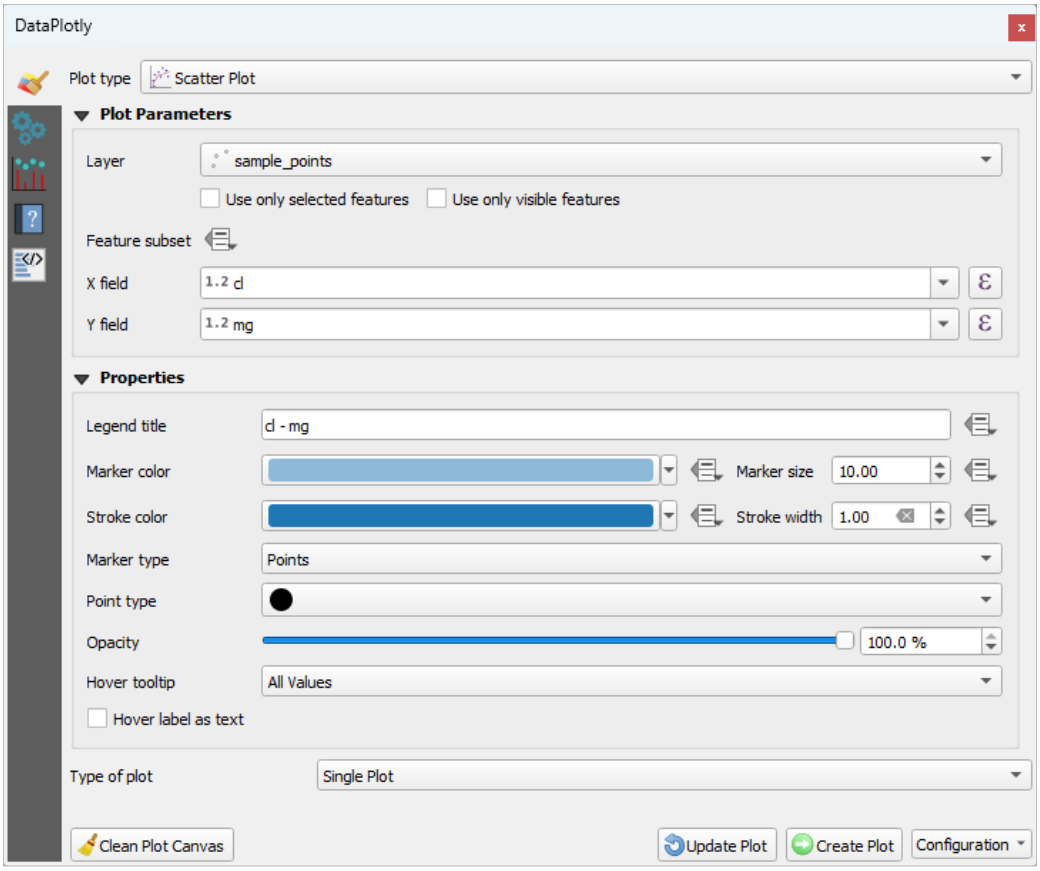
9.2.4 Follow Along: The DataPlotly Plugin

The **DataPlotly** plugin allows you to create D3 plots of vector attributes data thanks to the plotly library.

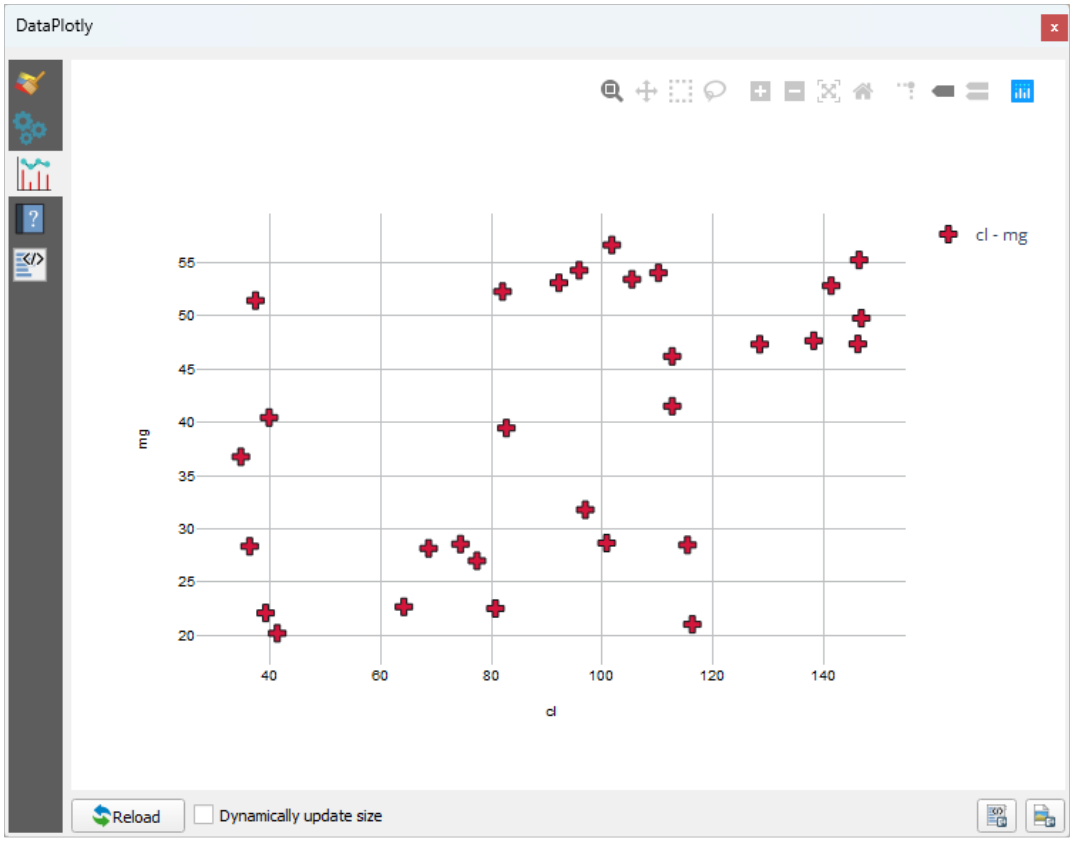
1. Start a new project
2. Load the *sample_points* layer from the `exercise_data/plugins` folder
3. Install the plugin following the guidelines described in *Follow Along: Installing New Plugins* searching *Data Plotly*
4. Open the plugin by clicking on the new icon in the toolbar or in the *Plugins* ► *Data Plotly* menu

In the following example we are creating a simple `Scatter Plot` of two fields of the *sample_points* layer. In the DataPlotly Panel:

1. Choose *sample_points* in the Layer filter, *cl* for the *X Field* and *mg* for the *Y Field*:



2. If you want you can change the colors, the marker type, the transparency and many other settings: try to change some parameters to create the plot below.



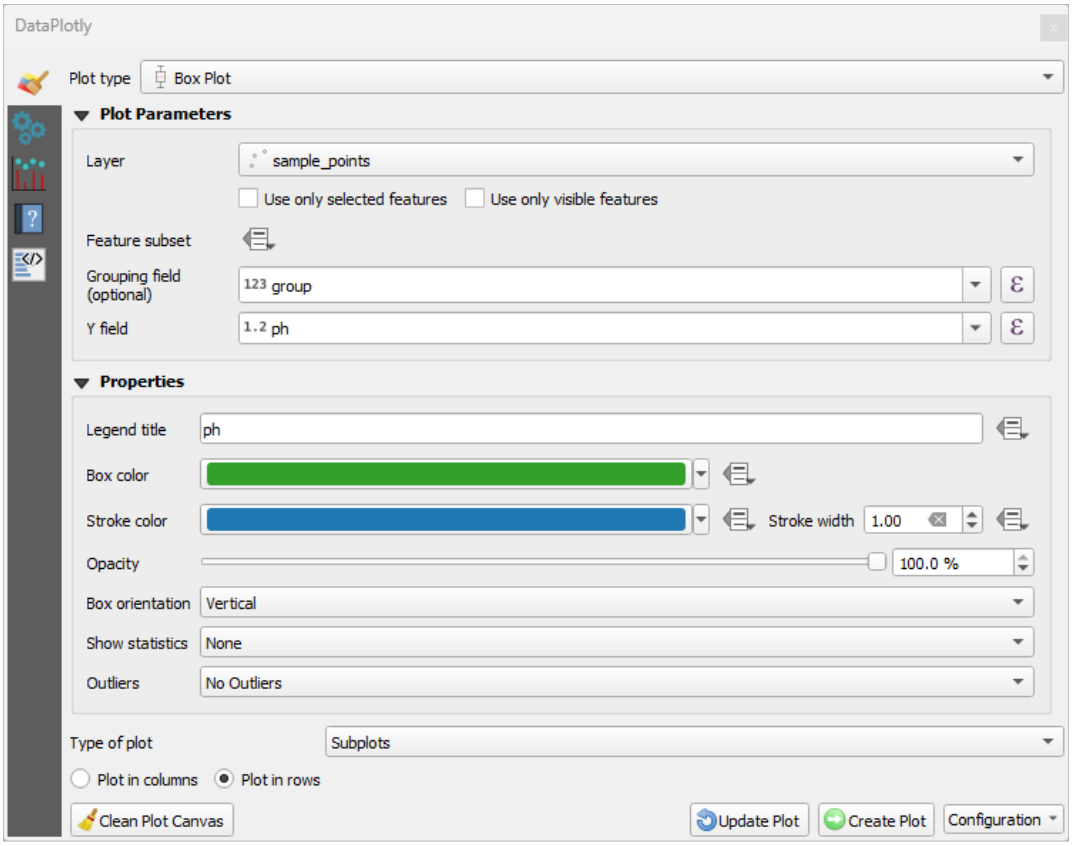
3. Once you have set all the parameters, click on the *Create Plot* button to create the plot.

The plot is interactive: this means you can use all the upper buttons to resize, move, or zoom in/out the plot canvas. Moreover, each element of the plot is interactive: by clicking or selecting one or more point on the plot, the corresponding point(s) will be selected in the plot canvas.

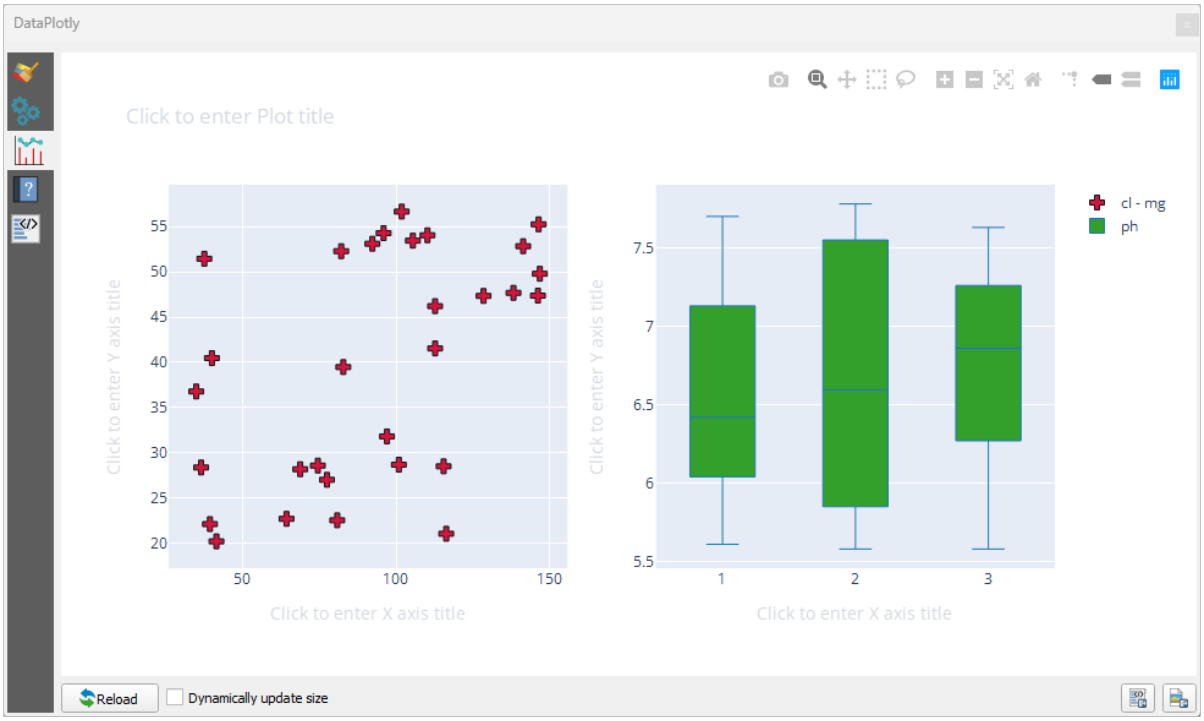
You can save the plot as a `png` static image or as an `html` file by clicking on the or on the button in the lower right corner of the plot.

There is more. Sometimes it can be useful to have two (or more) plots showing different plot types with different variables on the same page. Let's do this!

1. Go back to the main plot settings tab by clicking on the button in the upper left corner of the plugin panel
2. Change the *Plot Type* to *Box Plot*
3. Choose *group* as *Grouping Field* and *ph* as *Y Field*
4. In the lower part of the panel, change the *Type of Plot* from *SinglePlot* to *SubPlots* and let the default option *Plot in Rows* selected.



5. Once done click on the *Create Plot* button to draw the plot



Now both scatter plot and box plot are shown in the same plot page. You still have the chance to click on each plot item and select the corresponding features in the map canvas.

Tipp: Each plot has its own manual page available in the tab. Try to explore all the plot types and see all the other settings available.

9.2.5 In Conclusion

There are many useful plugins available for QGIS. Using the built-in tools for installing and managing these plugins, you can find new plugins and make optimum use of them.

9.2.6 What's Next?

Next we'll look at how to use layers that are hosted on remote servers in real time.

KAPITEL 10

Module: Datenquellen aus dem Internet

Wenn Sie Informationen für Ihre Karte verwenden möchten, müssen Sie sich nicht auf Daten beschränken, die auf Ihrem Computer gespeichert sind. Es gibt Online-Datenquellen, aus denen Sie Daten laden können, sofern Sie mit dem Internet verbunden sind.

In diesem Modul werden wir zwei Arten internet-basierter GIS-Dienste behandeln: Web Mapping Services (WMS) und Web Feature Services (WFS).

10.1 Lesson: Web Mapping Services

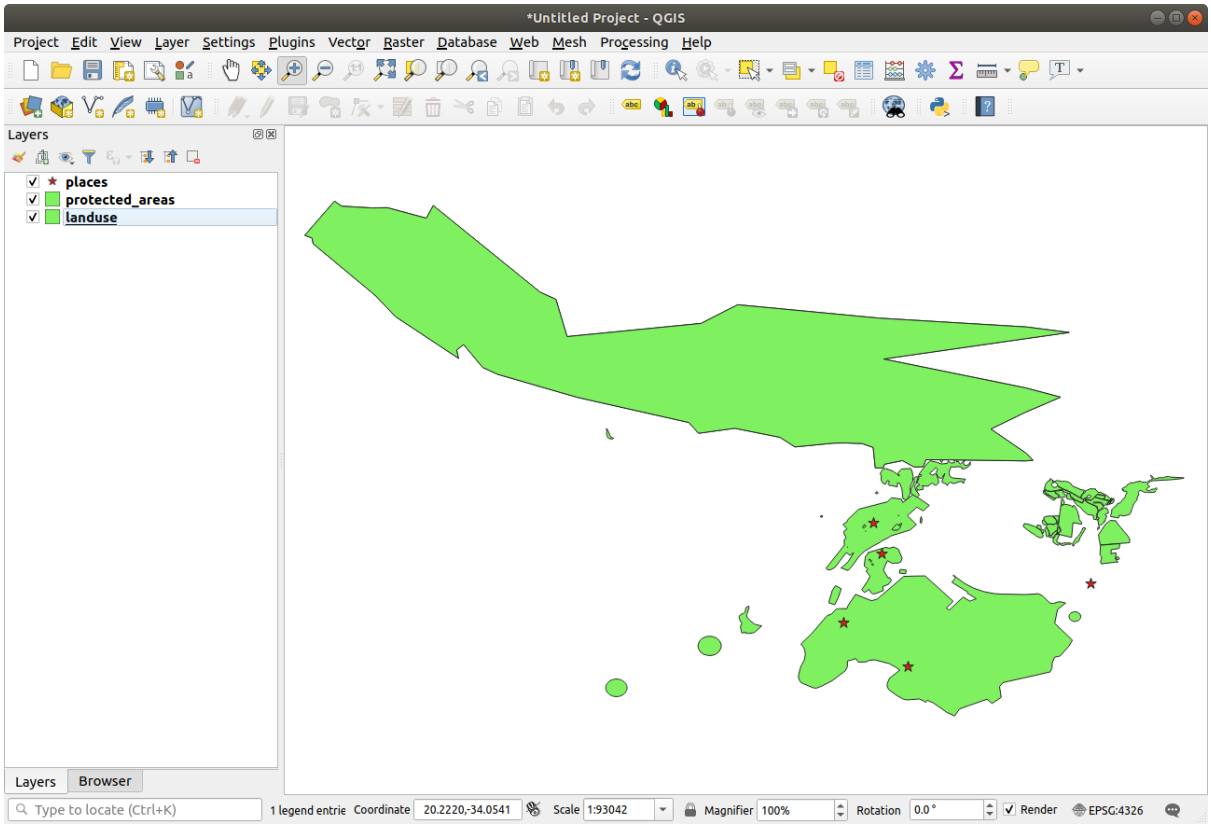
A Web Mapping Service (WMS) is a service hosted on a remote server. Similar to a website, you can access it as long as you have a connection to the server. Using QGIS, you can load a WMS directly into your existing map.

From the lesson on plugins, you will remember that it's possible to load a new raster image from Google, for example. However, this is a once-off transaction: once you have downloaded the image, it doesn't change. A WMS is different in that it's a live service that will automatically refresh its view if you pan or zoom on the map.

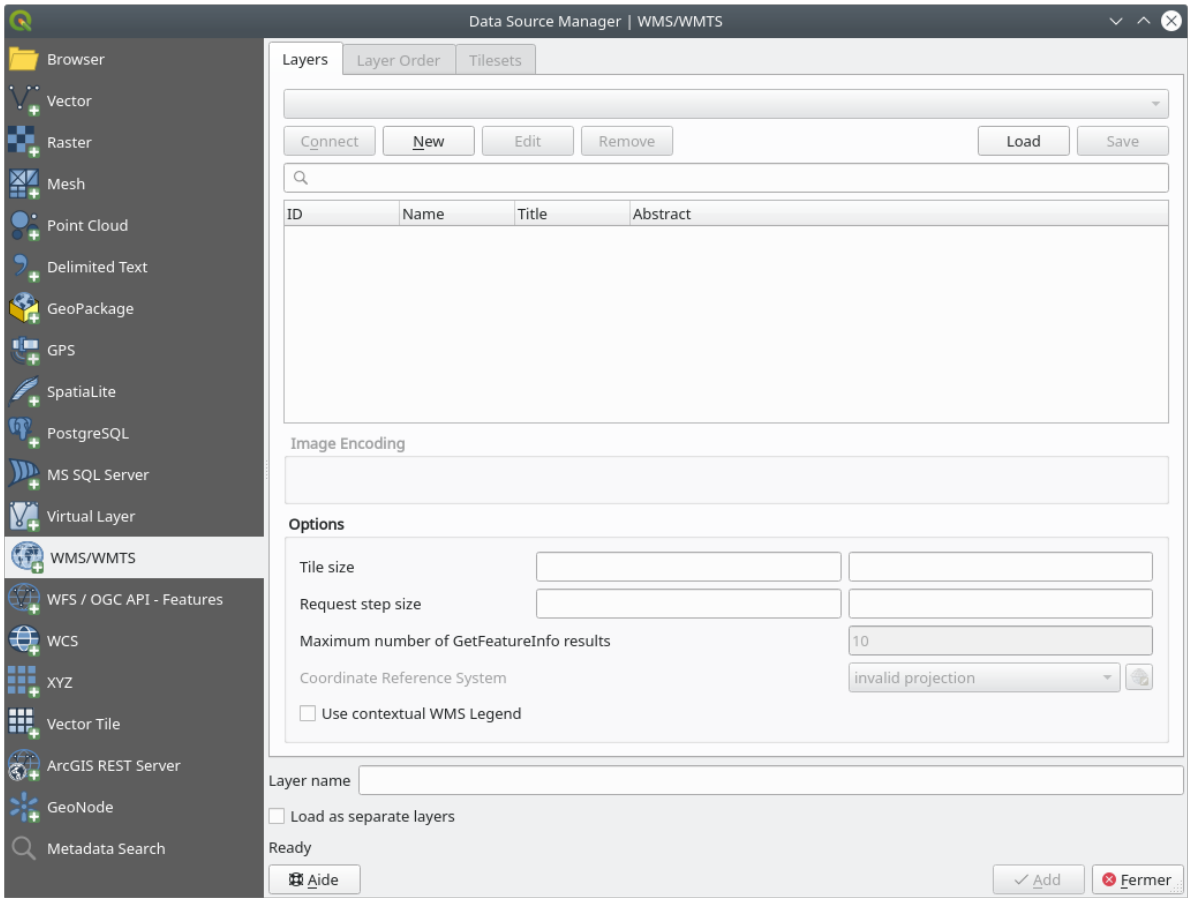
The goal for this lesson: To use a WMS and understand its limitations.

10.1.1 Follow Along: Loading a WMS Layer

For this exercise, you can either use the basic map you made at the start of the course, or just start a new map and load some existing layers into it. For this example, we used a new map and loaded the original *places*, *landuse* and *protected_areas* layers and adjusted the symbology:



1. Load these layers into a new map, or use your original map with only these layers visible.
2. Before starting to add the WMS layer, deactivate „on the fly“ projection (*Project* ► *Properties...* ► *CRS* tab, check *No CRS (or unknown/non-Earth projection)*. This may cause the layers to no longer overlap properly, but don't worry: we'll fix that later.
3. To add WMS layers, click on the button to open the *Data Source Manager* dialog and enable the *WMS/WMTS* tab.

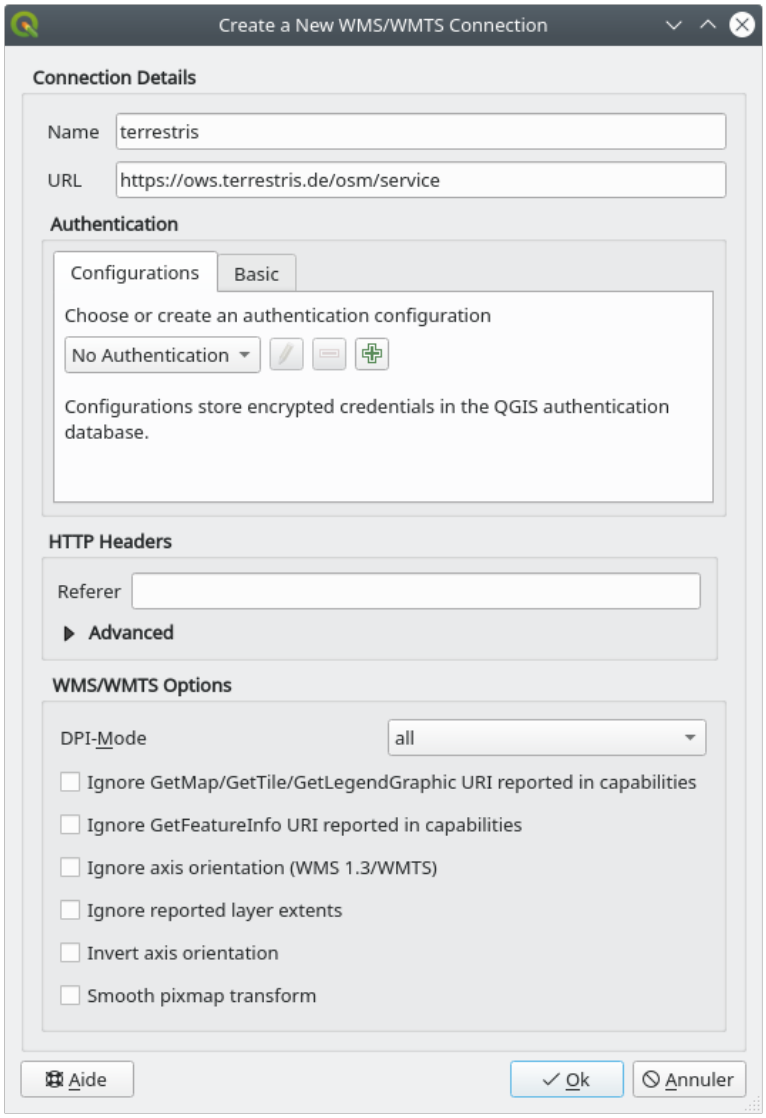


Remember how you connected to a SpatiaLite or GeoPackage database at the beginning of the course. The *landuse*, *buildings*, and *roads* layers are stored in a database. To use those layers, you first needed to connect to the database. Using a WMS is similar, with the exception that the layers are on a remote server.

4. To create a new connection to a WMS, click on the *New* button.

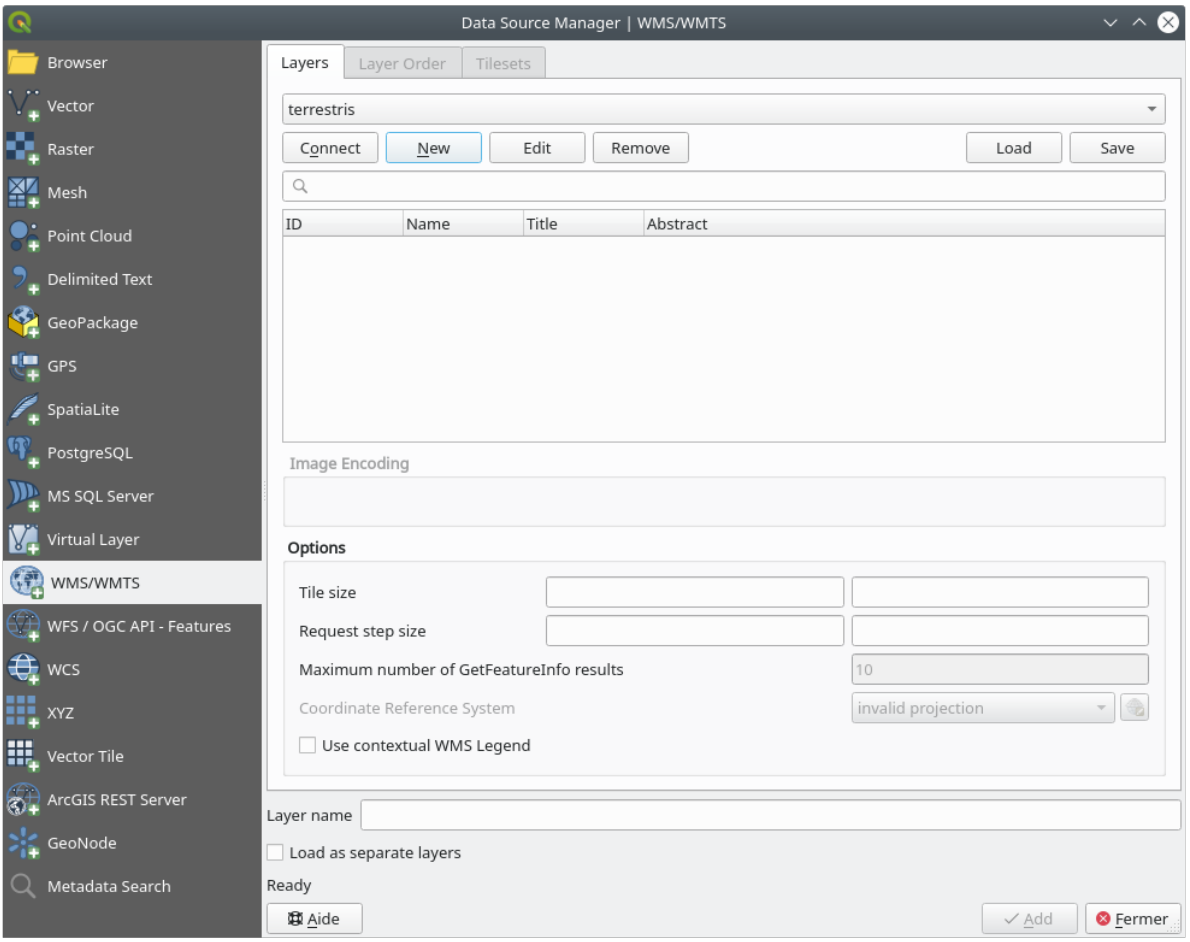
   You'll need a WMS address to continue. There are several free WMS servers available on the Internet. One of these is terrestris, which makes use of the OpenStreetMap dataset.

5. To make use of this WMS, set it up in your current dialog, like this:

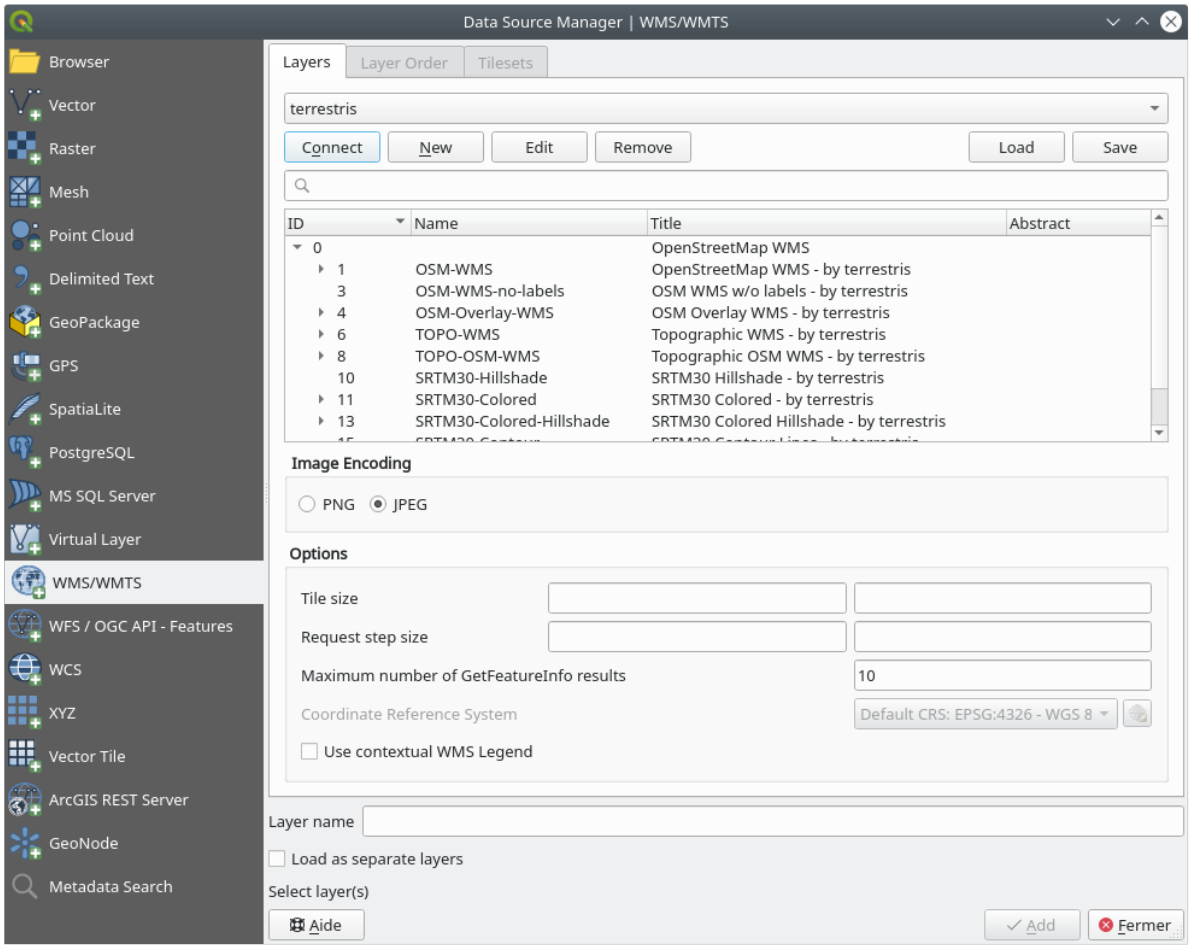


- The value of the *Name* field should be `terrestris`.
- The value of the *URL* field should be `https://ows.terrestris.de/osm/service`.

6. Click *OK*. You should see the new WMS server listed:

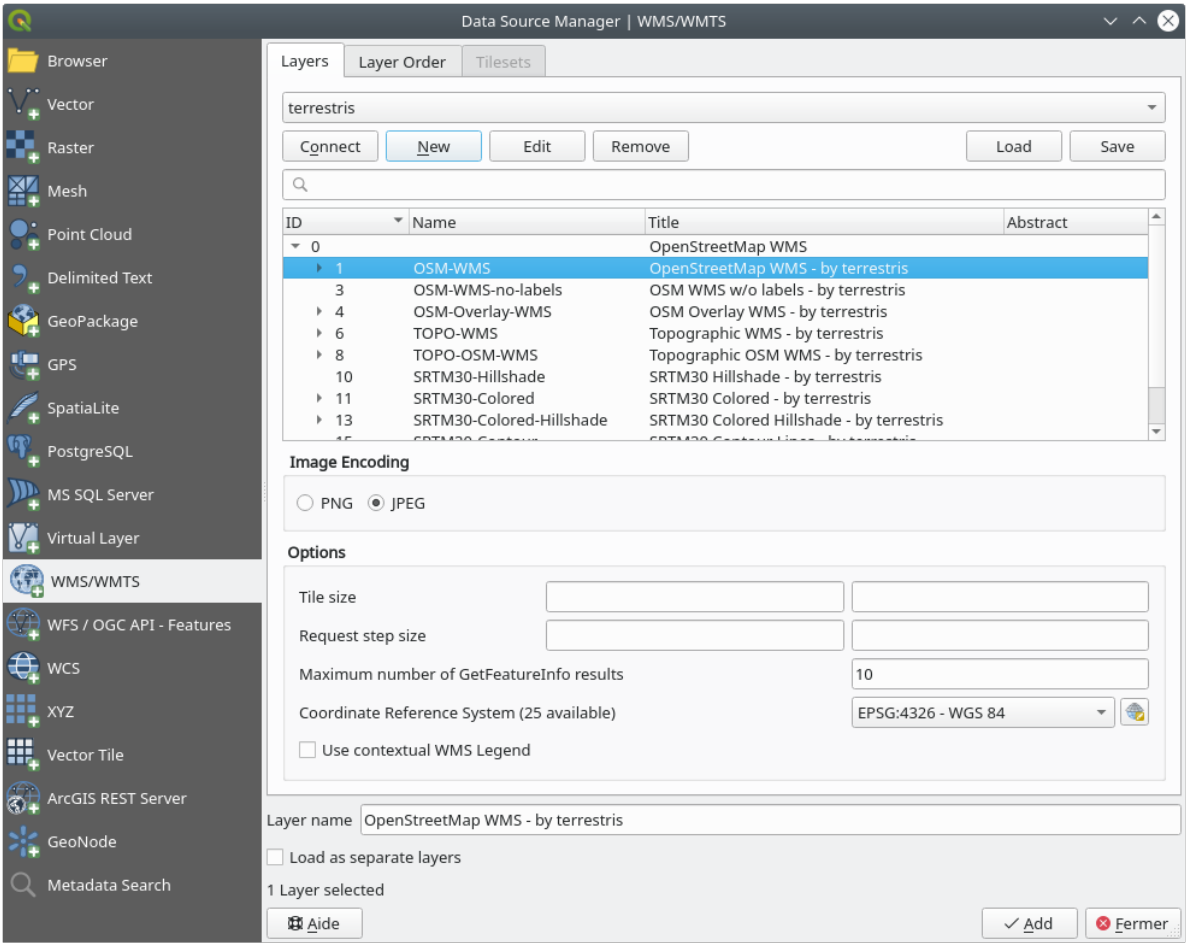


7. Click *Connect*. In the list below, you should now see these new entries loaded:



These are all the layers hosted by this WMS server.

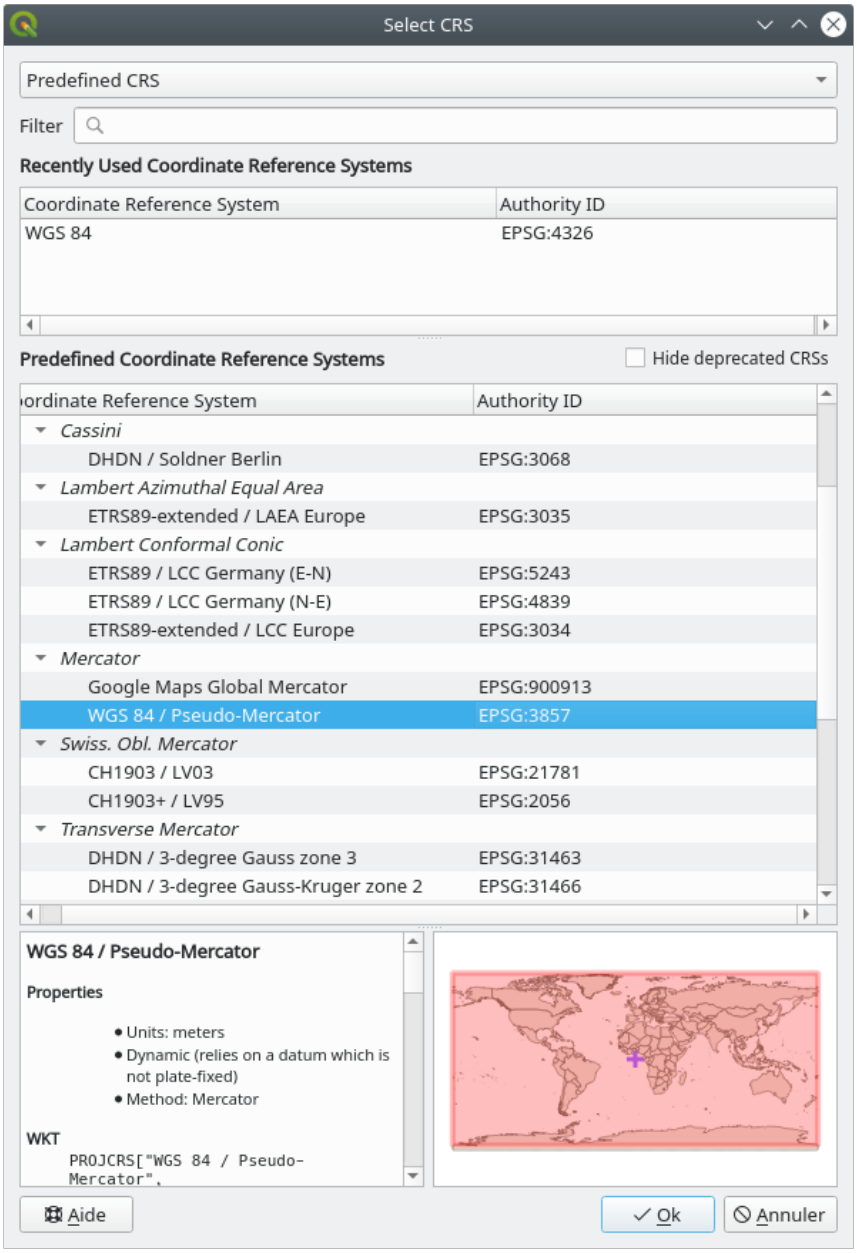
8. Click once on the *OSM-WMS* layer. This will display the default *Coordinate Reference System* in use and the number of CRSes supported by the dataset:



Since we're not using `EPSG:4326 – WGS 84` for our map, let's find one that meets our needs.

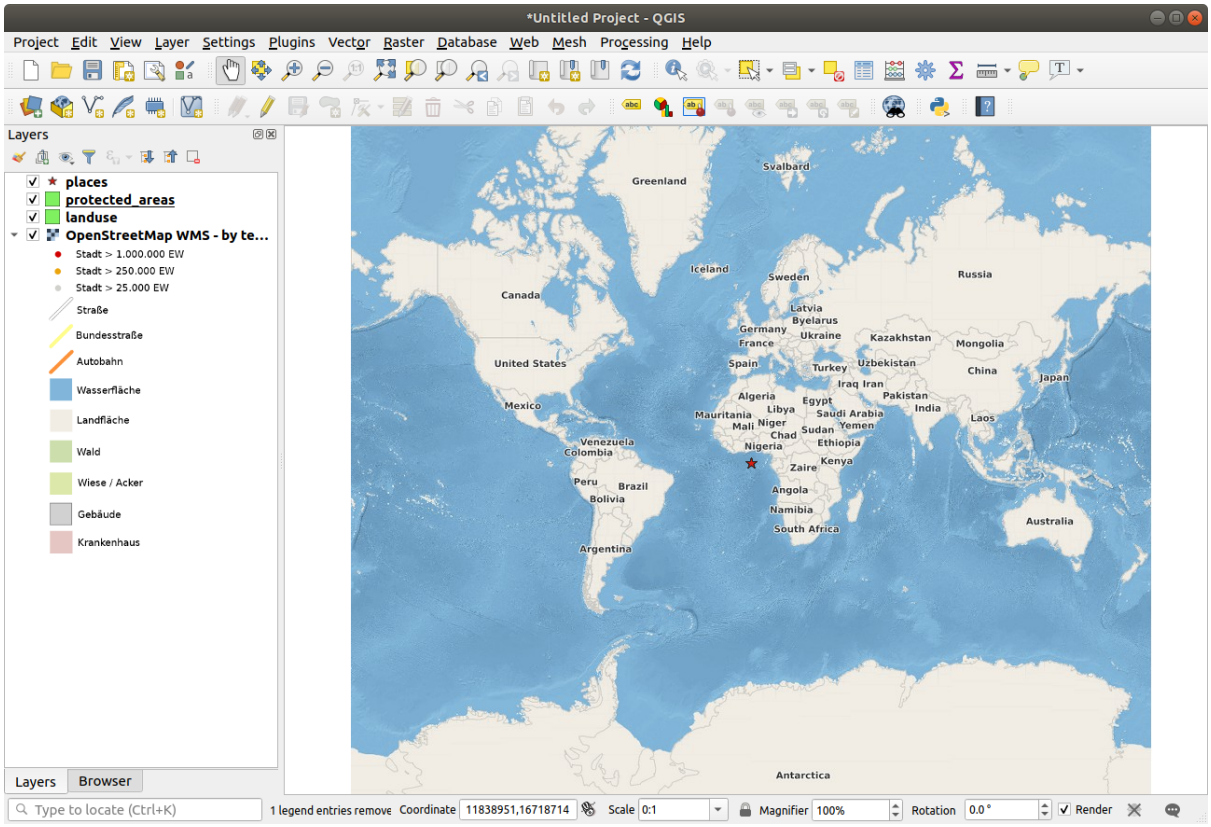
1. Click the

button. You will see a standard *Coordinate Reference System Selector* dialog showing all the CRS the data is published with.
2. We want a *projected* CRS, so let's choose *WGS 84 / Pseudo-Mercator* with `EPSG:3857`. You can use the top *Filter* widget.



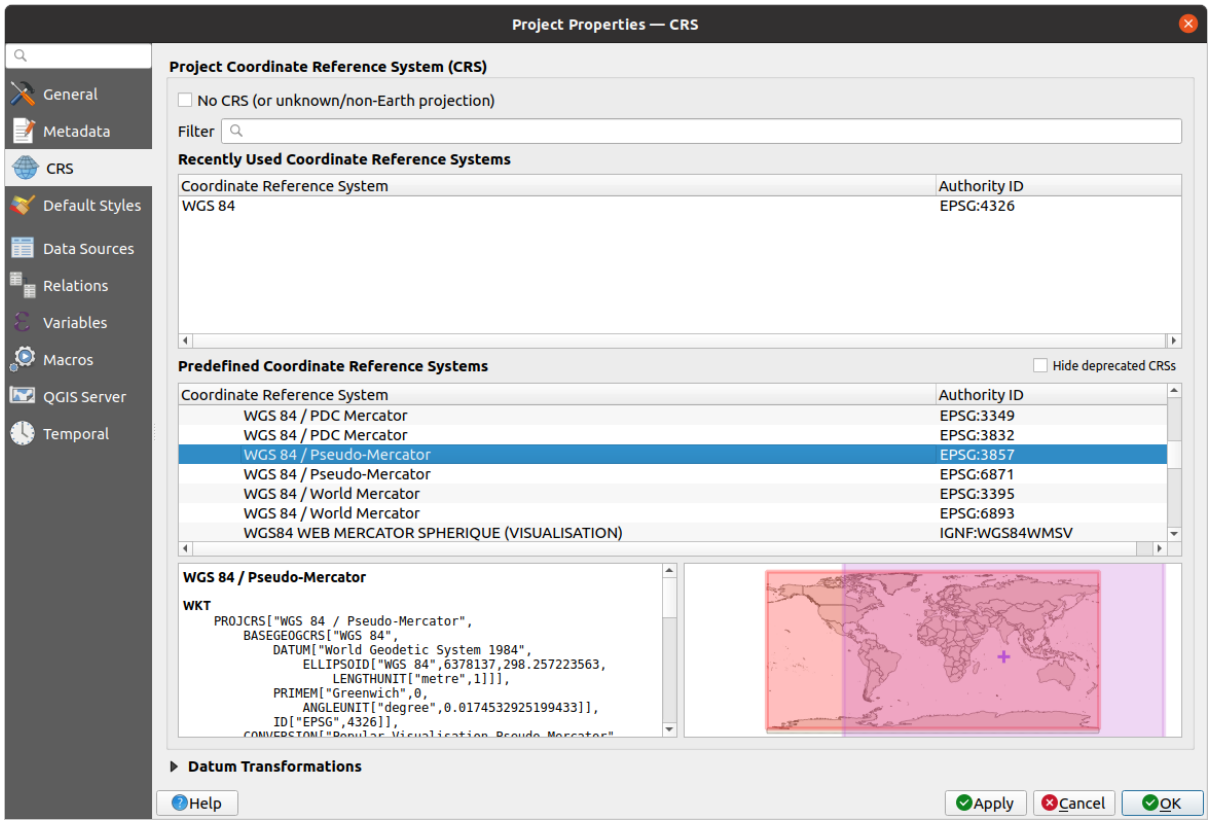
1. Click *OK*. The Coordinate Reference System associated with the entry has changed.

9. Click *Add* to load the layer in your project using *Layer name`* (default is *OpenStreetMap WMS - by terrestris*).
10. Close the *Data Source Manager* dialog if not done automatically
11. In the *Layers* panel, click and drag it to the bottom of the list.
12. Zoom out in order to get a global view of the layers. You will notice that your layers aren't located correctly (near west of Africa). This is because „on the fly" projection is disabled.



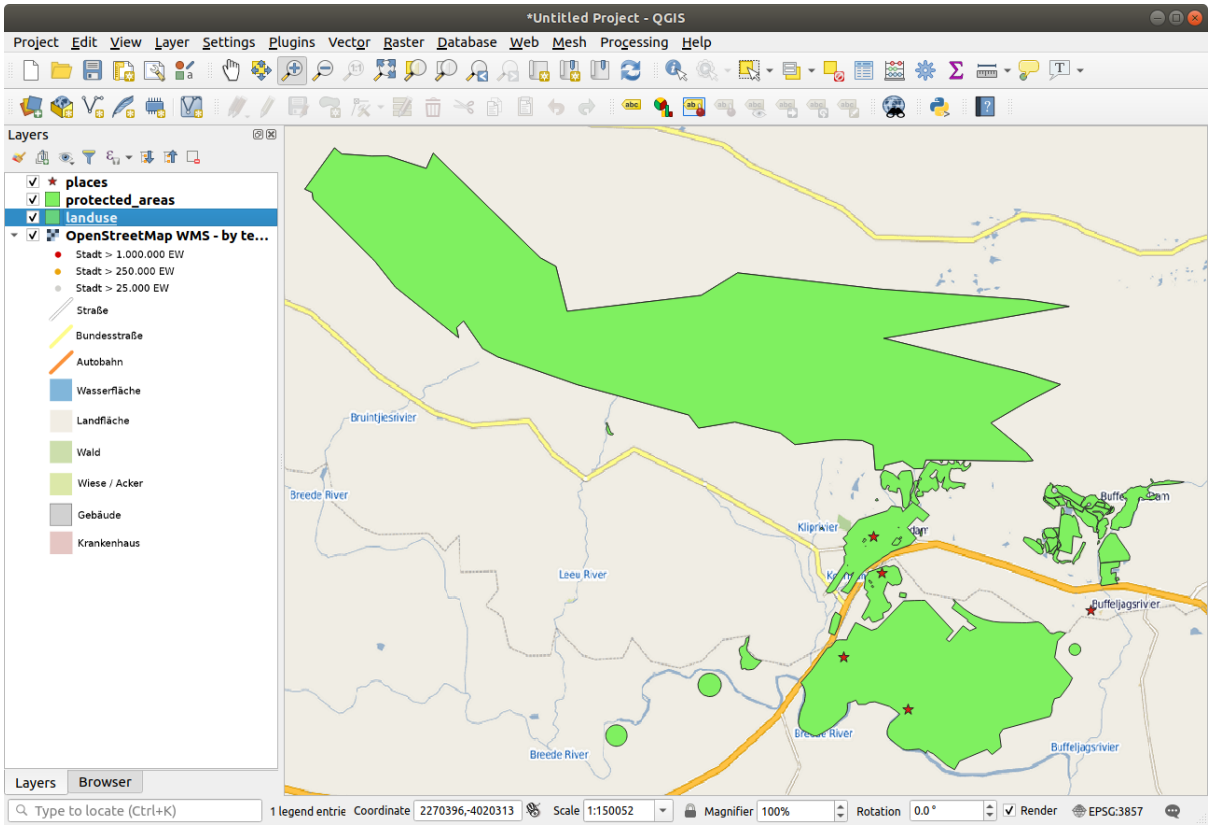
13. Let's enable the reprojection again, but using the same projection as the *OpenStreetMap WMS* layer, which is *WGS 84 / Pseudo Mercator*.

    1. Open the *Project* ► *Properties...* ► *CRS* tab
    2. Uncheck *No CRS (or unknown/non-Earth projection)*
    3. Choose *WGS 84 / Pseudo Mercator* from the list.



4. Click *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:



Note how the WMS layer's streets and our own streets overlap. That's a good sign!

The nature and limitations of WMS

By now you may have noticed that this WMS layer actually has many features in it. It has streets, rivers, nature reserves, and so on. What's more, even though it looks like it's made up of vectors, it seems to be a raster, but you can't change its symbology. Why is that?

This is how a WMS works: it's a map, similar to a normal map on paper, that you receive as an image. What usually happens is that you have vector layers, which QGIS renders as a map. But using a WMS, those vector layers are on the WMS server, which renders it as a map and sends that map to you as an image. QGIS can display this image, but can't change its symbology, because all that is handled on the server.

This has several advantages, because you don't need to worry about the symbology. It's already worked out, and should be nice to look at on any competently designed WMS.

On the other hand, you can't change the symbology if you don't like it, and if things change on the WMS server, then they'll change on your map as well. This is why you sometimes want to use a Web Feature Service (WFS) instead, which gives you vector layers separately, and not as part of a WMS-style map.

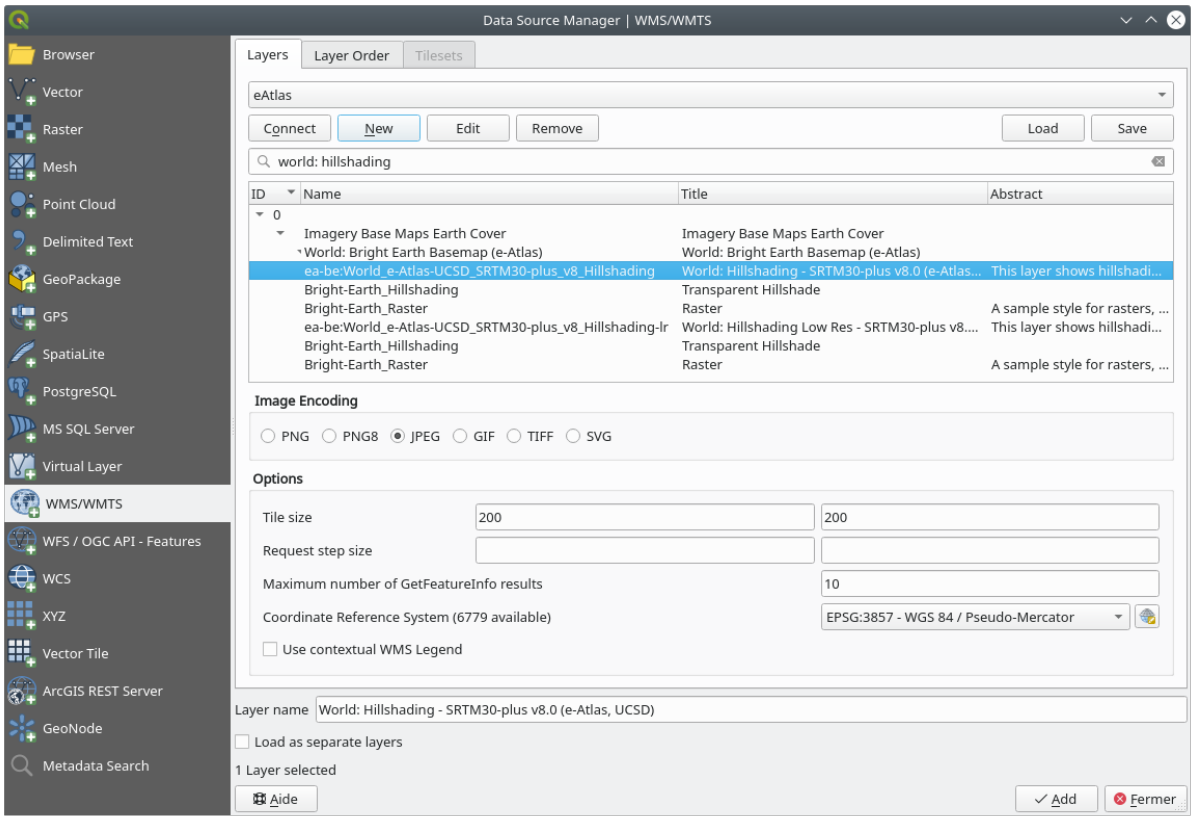
This will be covered in the next lesson, however. First, let's add another WMS layer.

10.1.2 Try Yourself

1. Add the `eAtlas` WMS server at this URL: `https://maps.eatlas.org.au/maps/wms`
2. Load a *World: Hillshading* layer into the map.
3. You might want to set its *Encoding* to *JPEG* and its *Tile size* option to `200` by `200`, so that it loads faster.
4. Your map should look like this (you may need to re-order the layers, and apply some transparency):

Answer

1. Go to the Data Source Manager, WMS / WMTS tab and create a new connection entry
2. Use the text box to filter the list of layers and select the corresponding layer
3. Remember to check/turn its CRS into *EPSG:3857 - WGS 84 / Pseudo Mercator* as the rest of the map.



4. After the layer is loaded, you can modify its *Opacity* value (under the *Transparency* properties tab)

10.1.3 Try Yourself

Part of the difficulty of using WMS is finding a good (free) server.

- Find a new WMS at directory.spatineo.com (or elsewhere online). It must not have associated fees or restrictions, and must have coverage over the Swellendam study area.

  Remember that what you need in order to use a WMS is only its URL (and preferably some sort of description).

Answer

Spatineo is one of many places you can search for a OGC data. Type in the text area a name, keyword, location of your interest to see if you get some working results. For this lesson, you might want to filter the results to include WMS only.

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 *MapServer Demonstration Server* WMS at https://demo.mapserver.org/cgi-bin/wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities. It does not require fees or have access constraints, and it is global. Therefore, it does

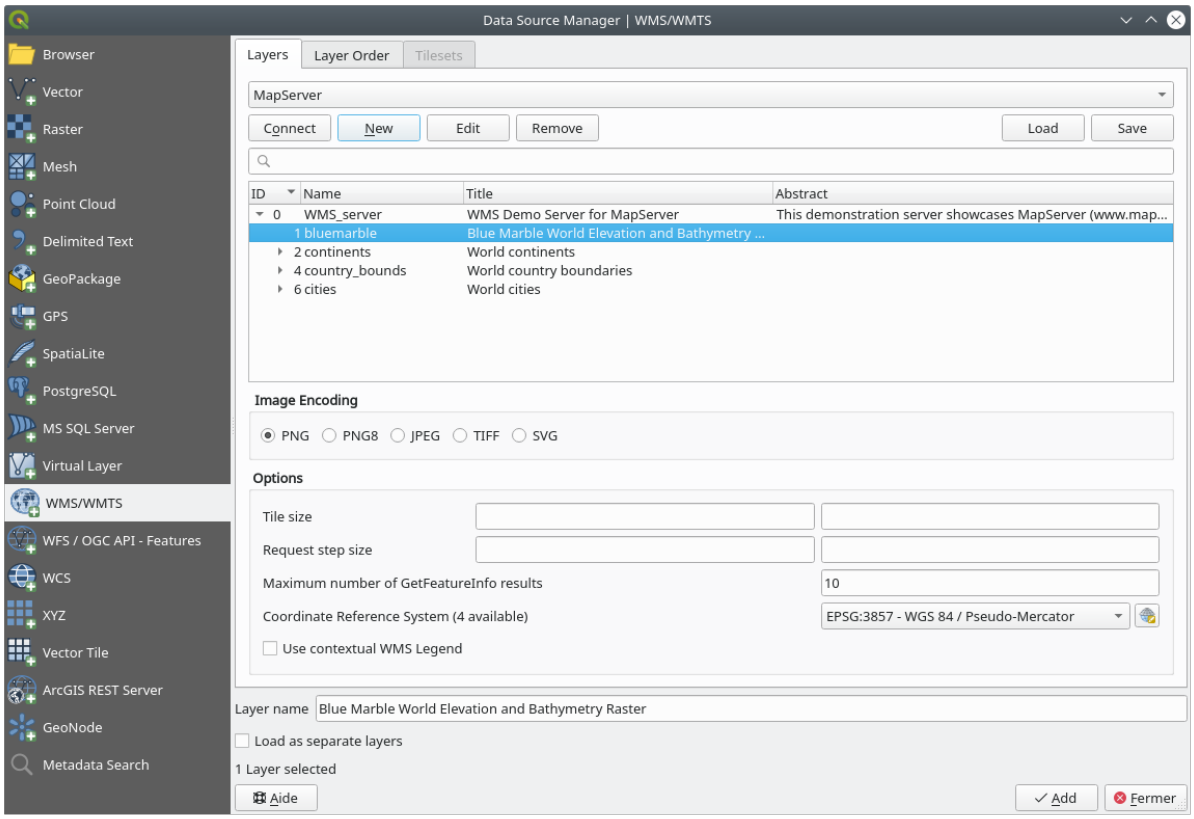
satisfy the requirements. Keep in mind, however, that this is merely an example. There are many other WMS servers to choose from.

10.1.4 Try Yourself

- Add the *bluemarble* layer from the *MapServer Demonstration Server*. Is that a suitable dataset for our study area?

Answer

1. Hide all other WMS layers to prevent them from rendering unnecessarily in the background.
2. Use the same approach as before to add the new server and the appropriate layer as hosted on that server:



3. 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

10.1.5 In Conclusion

Using a WMS, you can add inactive maps as backdrops for your existing map data.

10.1.6 Further Reading

- Spatineo Directory
- OpenStreetMap.org list of WMS servers

10.1.7 What's Next?

Now that you've added an inactive map as a backdrop, you'll be glad to know that it's also possible to add features (such as the other vector layers you added before). Adding features from remote servers is possible by using a Web Feature Service (WFS). That's the topic of the next lesson.

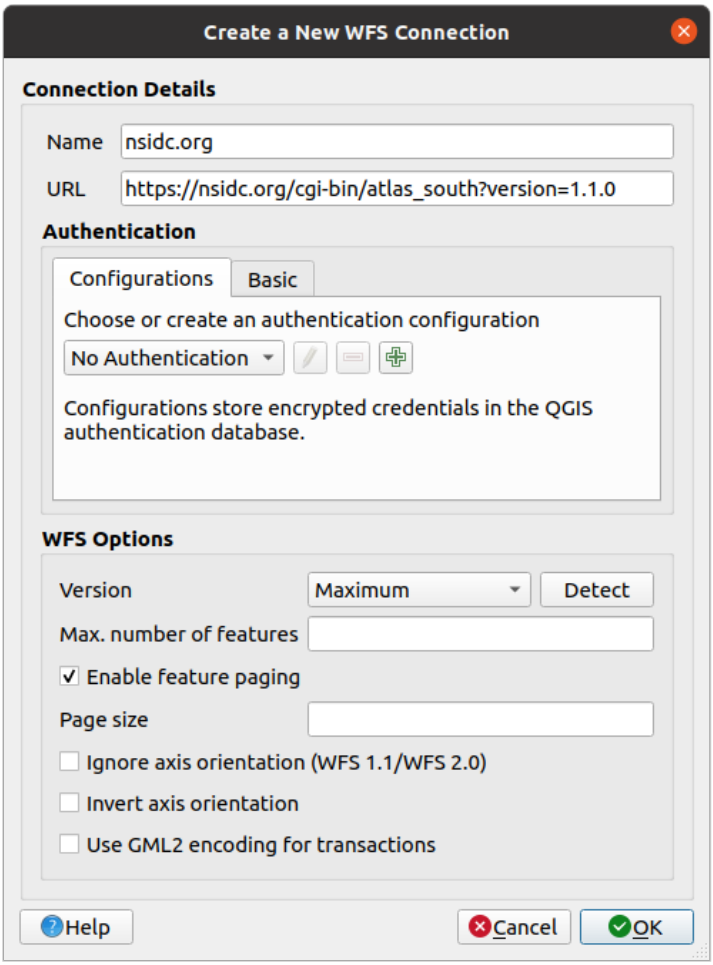
10.2 Lesson: Web Feature Services

A Web Feature Service (WFS) provides its users with GIS data in formats that can be loaded directly in QGIS. Unlike a WMS, which provides you only with a map which you can't edit, a WFS gives you access to the features themselves.

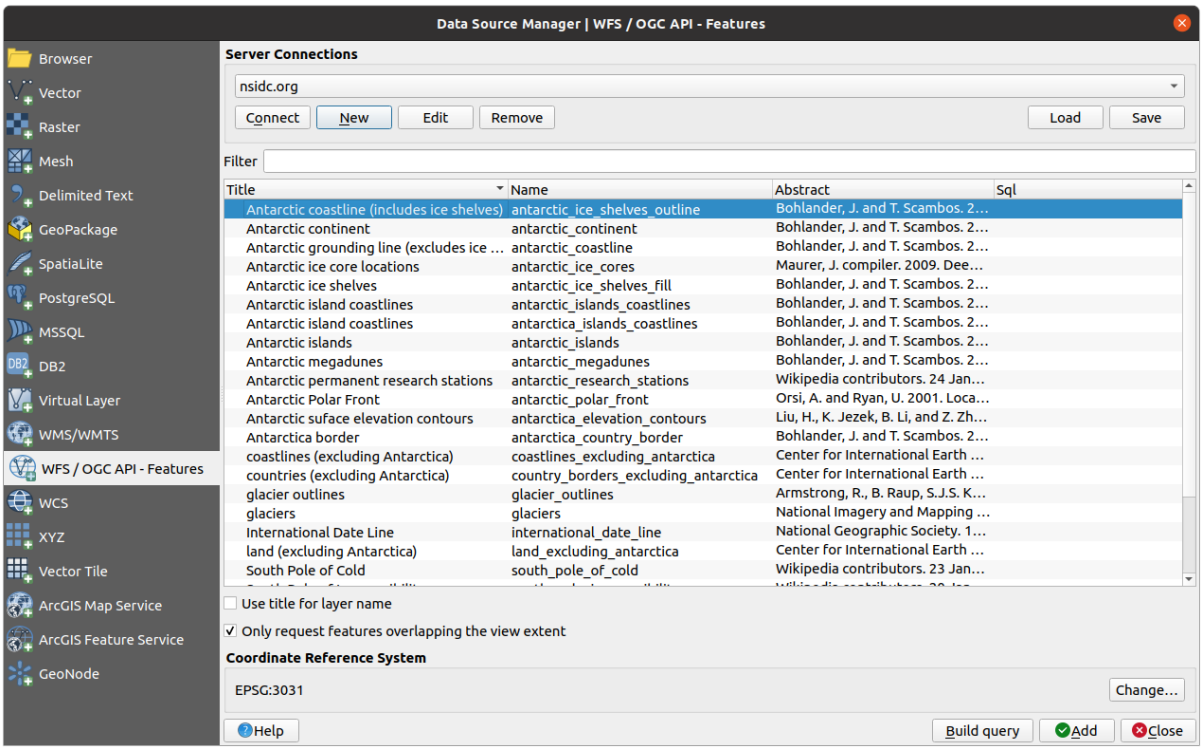
The goal for this lesson: To use a WFS and understand how it differs from a WMS.

10.2.1 Follow Along: Loading a WFS Layer

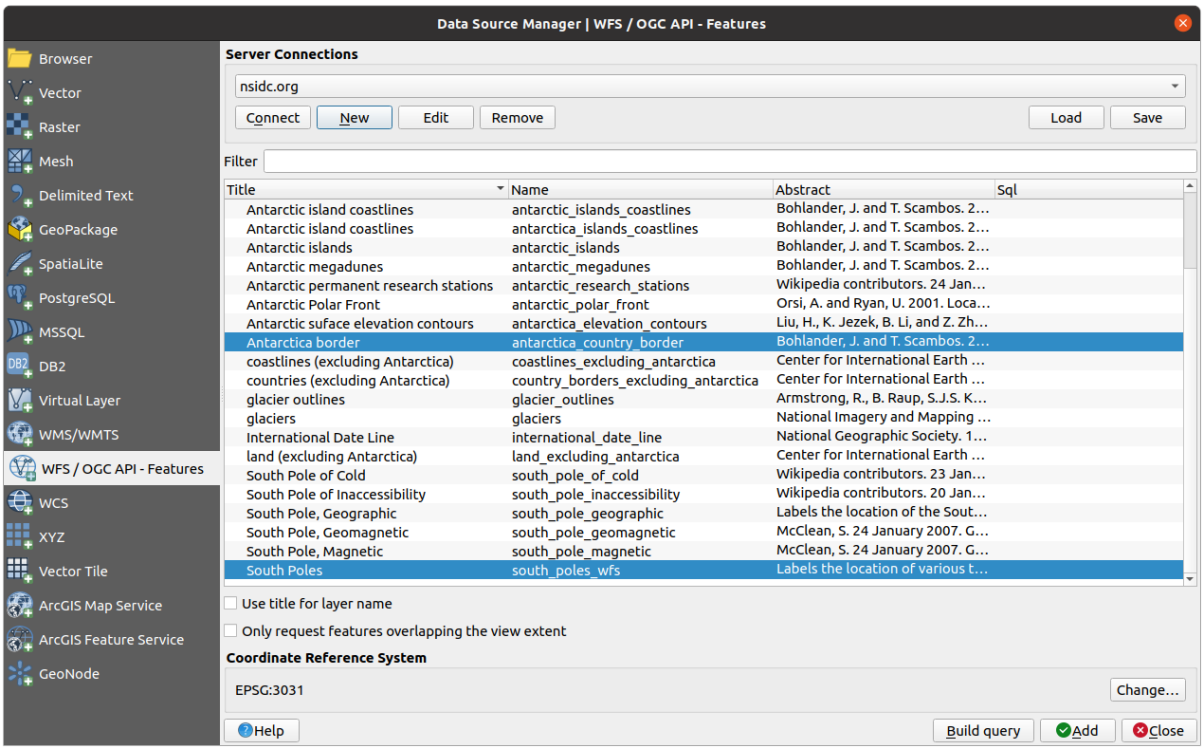
1. Start a new map. This is for demo purposes and won't be saved.
2. Click the Open Data Source Manager button.
3. Enable the *WFS / OGC API - Features* tab.
4. Click the *New* button.
5. In the dialog that appears, enter the *Name* as `nsidc.org` and the *URL* as `https://nsidc.org/cgi-bin/atlas_south?version=1.1.0`.



6. Click *OK*, and the new connection will appear in your *Server connections*.
7. Click the *Connect*. A list of the available layers will appear:

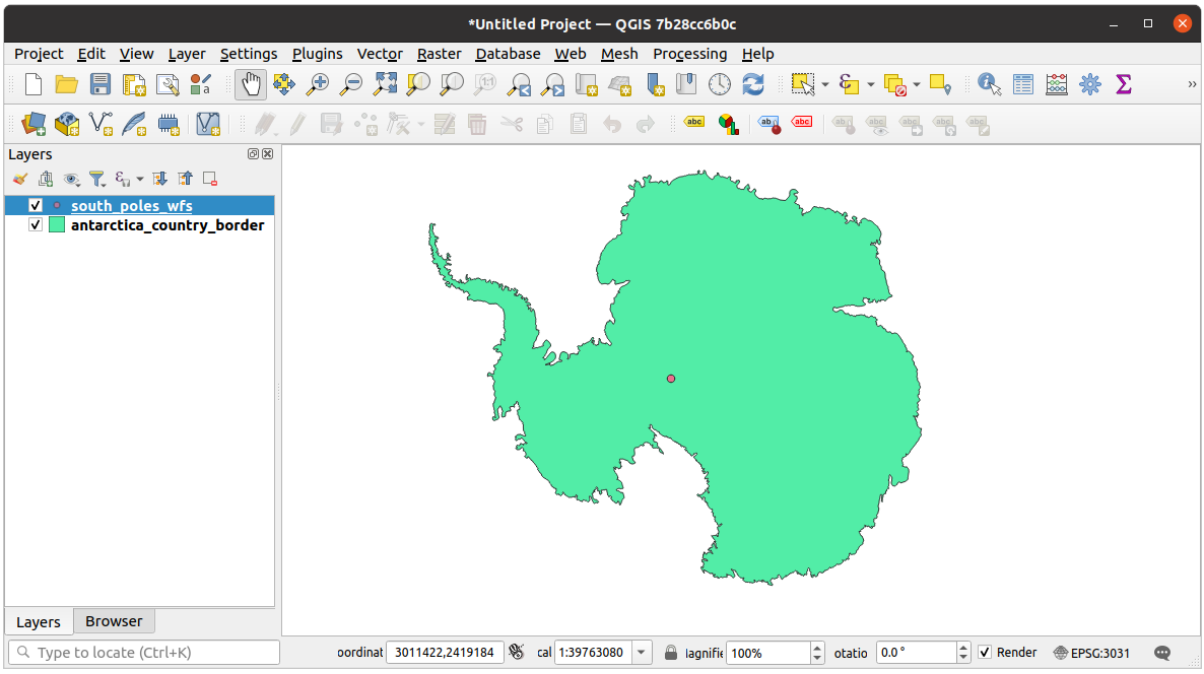


8. Uncheck the *Only request features overlapping the view extent* option below the layers list, since your current map canvas may not cover our area of interest: Antarctica.
9. Find the layer *antarctica_country_border*. You can use the *Filter* box at the top.
10. Click on the layer to select it:
11. Find and select also the layer *south_poles_wfs*. You might need to hold `Ctrl`.



12. Click *Add*.

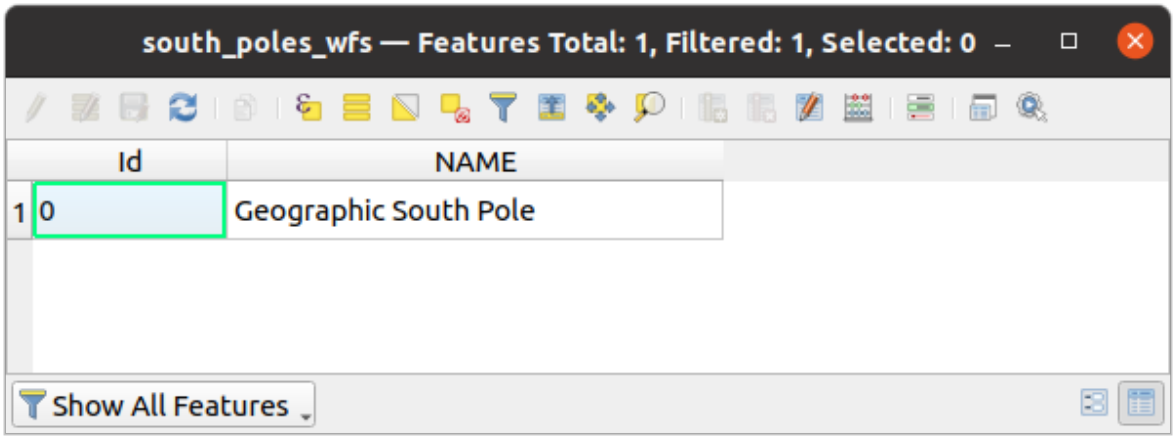
    It may take a while to load the layers. When they are loaded, they will appear in the map, showing the outlines of Antarctica and a few points over.



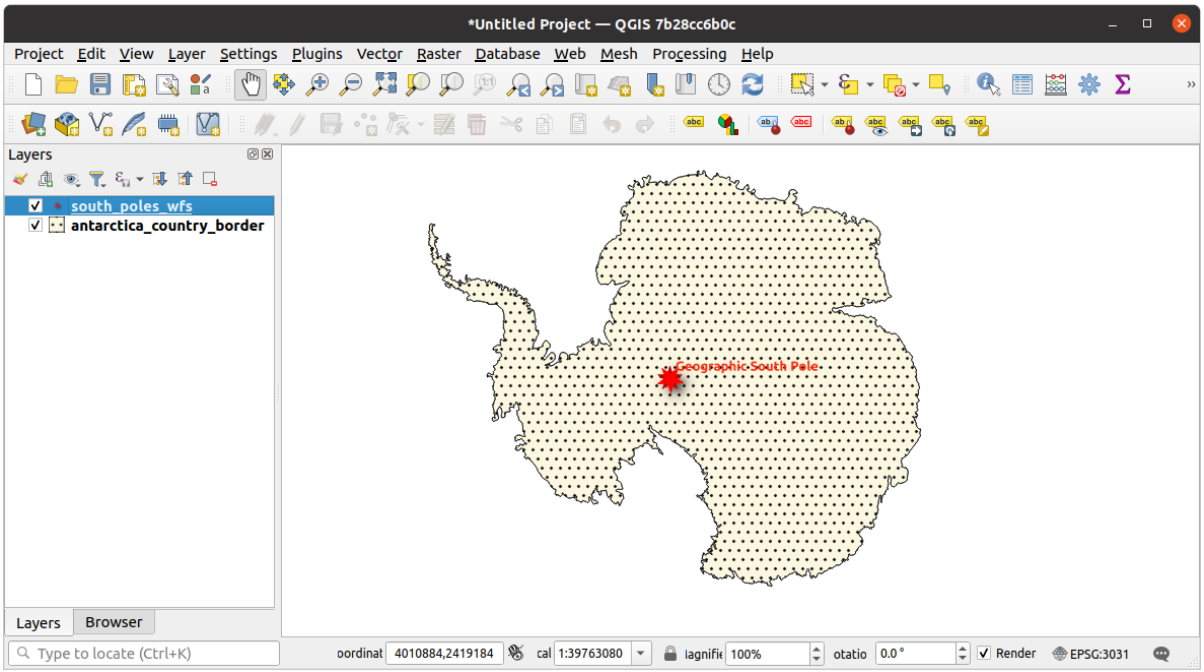
How is this different from having a WMS layer?

13. Select any of the layers and you'll notice that feature selection and attribute table tools are enabled. These are vector layers.

14. Select the *south_poles_wfs* layer and open its attribute table. You should see this:



Since the points have attributes, we are able to label them, as well as change their symbology. Here's an example:



Differences from WMS layers

A Web Feature Service returns the layer itself, not just a map rendered from it. This gives you direct access to the data, meaning that you can change its symbology and run analysis functions on it. However, this is at the cost of much more data being transmitted. This will be especially obvious if the layers you're loading have complicated shapes, a lot of attributes, or many features; or even if you're just loading a lot of layers. WFS layers typically take a very long time to load because of this.

10.2.2 Follow Along: Querying a WFS Layer

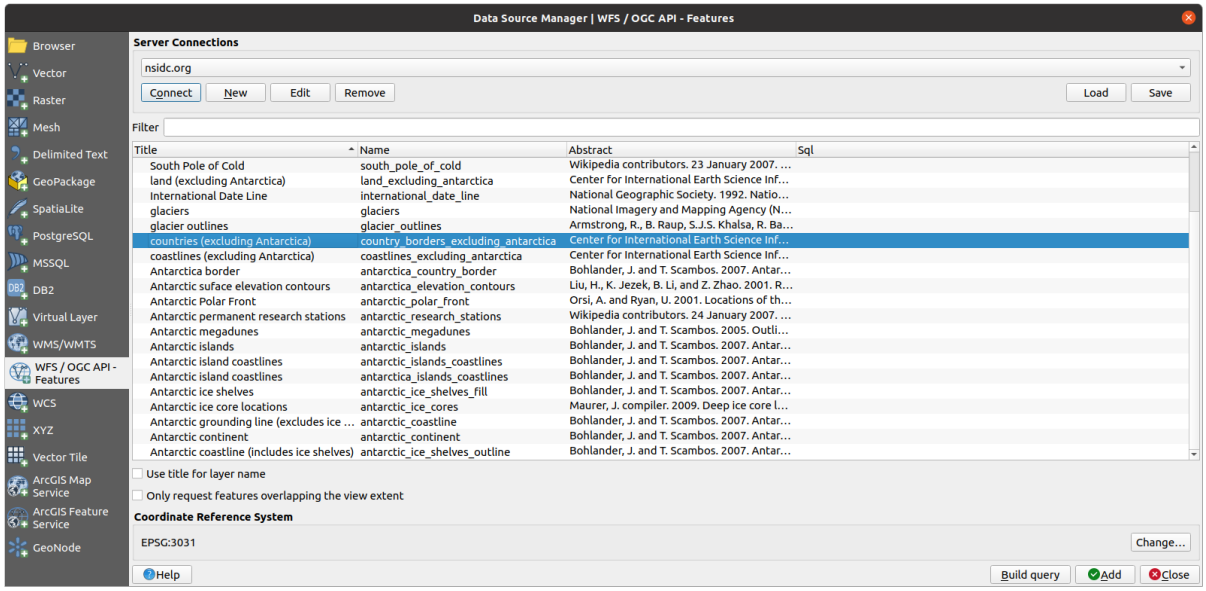
Although it is of possible to query a WFS layer after having loaded it, it's often more efficient to query it before you load it. That way, you're only requesting the features you want, meaning that you use far less bandwidth.

For example, on the WFS server we're currently using, there is a layer called *countries (excluding Antarctica)*. Let's say that we want to know where South Africa is relative to the *south_poles_wfs* layer (and perhaps also the *antarctica_country_border* layer) that's already been loaded.

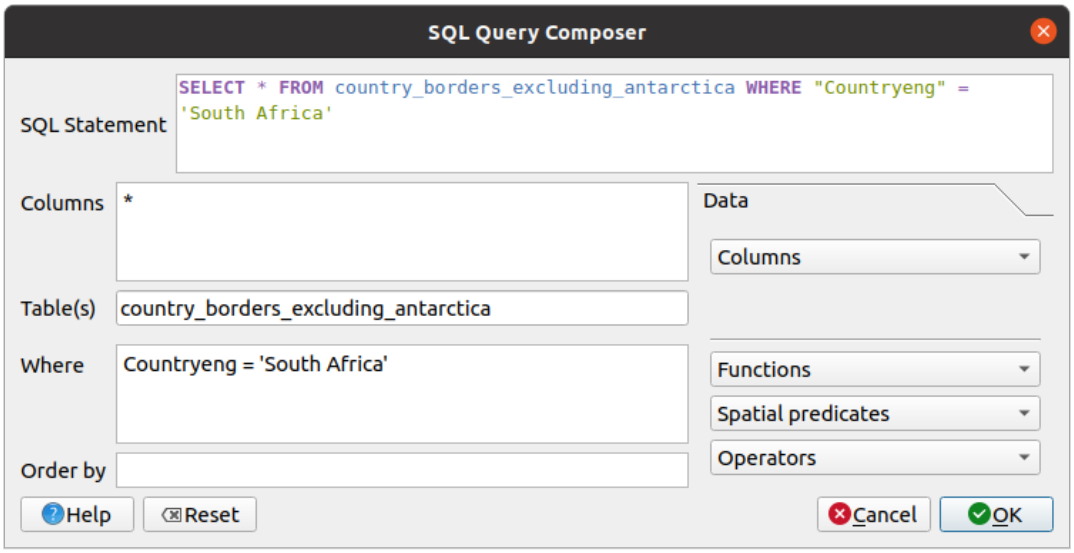
There are two ways to do this. You can load the whole *countries …* layer, and then build a query as usual once it's loaded. However, transmitting the data for all the countries in the world and then only using the data for South Africa seems a bit wasteful of bandwidth. Depending on your connection, this dataset can take several minutes to load.

The alternative is to build the query as a filter before even loading the layer from the server.

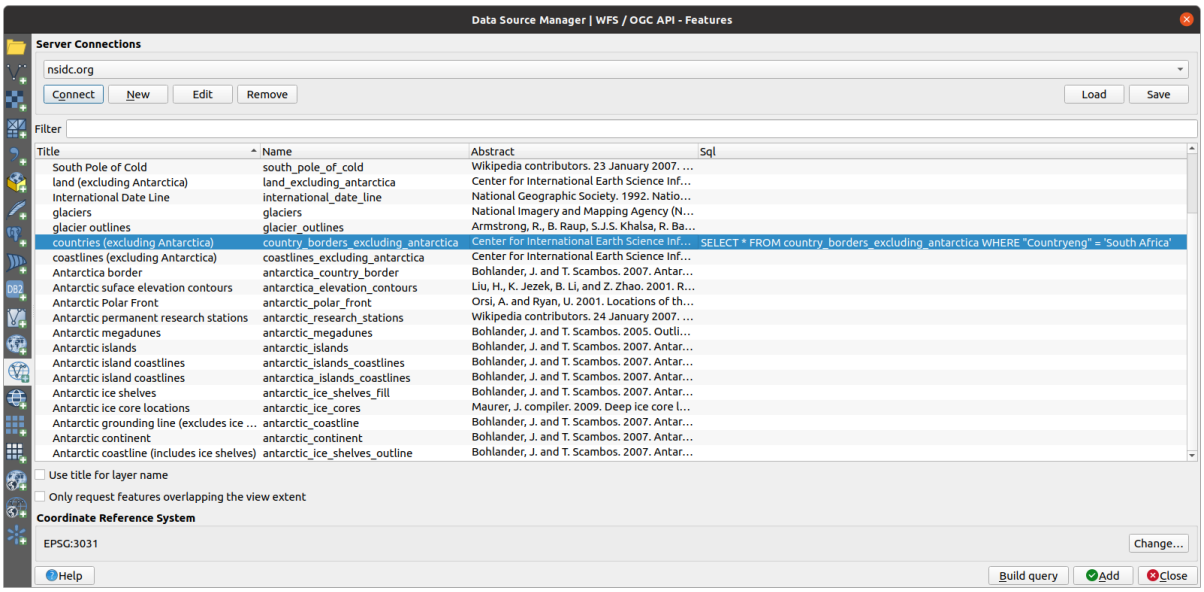
1. Enable the *WFS / OGC API Features* tab in the *Data Source Manager* dialog
2. Connect to the server we used before and you should see the list of available layers.
3. Find and double-click the *countries (excluding Antarctica)* layer. The layer name is `country_borders_excluding_antarctica`. You can also select the layer and press *Build query* button at the bottom of the dialog:



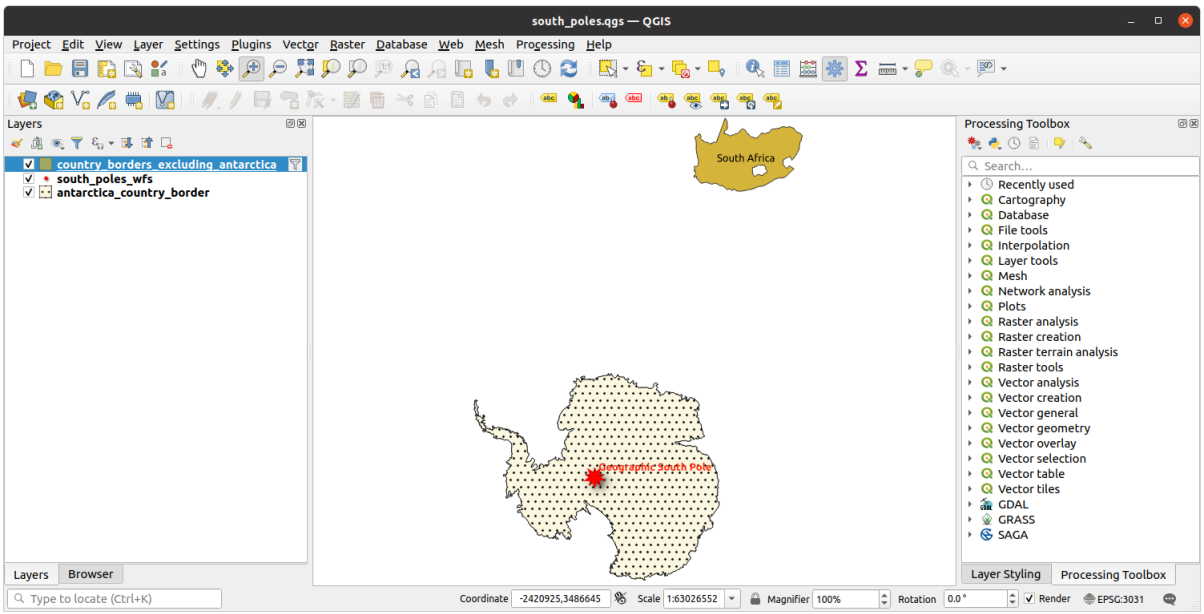
4. In the dialog that appears, type the following `SELECT * FROM country_borders_excluding_antarctica WHERE "Countryeng" = 'South Africa'` query in the *SQL Statement* box.



5. Press *OK*.
6. The expression used will appear as the *Sql* value of the target layer:



7. Click *Add* with the layer selected as above. Only the country with the `Countryeng` value of `South Africa` will load from that layer:



Did you notice the icon next to the `country_borders_excluding_antarctica` layer? It indicates that the loaded layer is filtered and does not display in the project all of its features.

8. You don't have to, but if you tried both methods, you'll notice that this is a lot faster than loading all the countries before filtering them!

Notes on WFS availability

It is rare to find a WFS hosting features you need, if your needs are very specific. The reason why Web Feature Services are relatively rare is because of the large amounts of data that must be transmitted to describe a whole feature. It is therefore not very cost-effective to host a WFS rather than a WMS, which sends only images.

The most common type of WFS you'll encounter will therefore probably be on a local network or even on your own computer, rather than on the Internet.

10.2.3 In Conclusion

WFS layers are preferable over WMS layers if you need direct access to the attributes and geometries of the layers. However, considering the amount of data that needs to be downloaded (which leads to speed problems and also a lack of easily available public WFS servers) it's not always possible to use a WFS instead of a WMS.

10.2.4 What's Next?

Next, you'll see how to use QGIS Server to provide OGC services.

KAPITEL 11

Module: QGIS Server

Dieses Modul wurde von Tudor Bărăscu entwickelt.

In diesem Modul werden wir die Installation und Nutzung von QGIS Server behandeln.

Um mehr über QGIS Server zu erfahren, lesen Sie die 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.

Bemerkung: 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 python3-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.

Bemerkung: 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.

Bemerkung: 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 *Lesson: Web Mapping Services* 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:

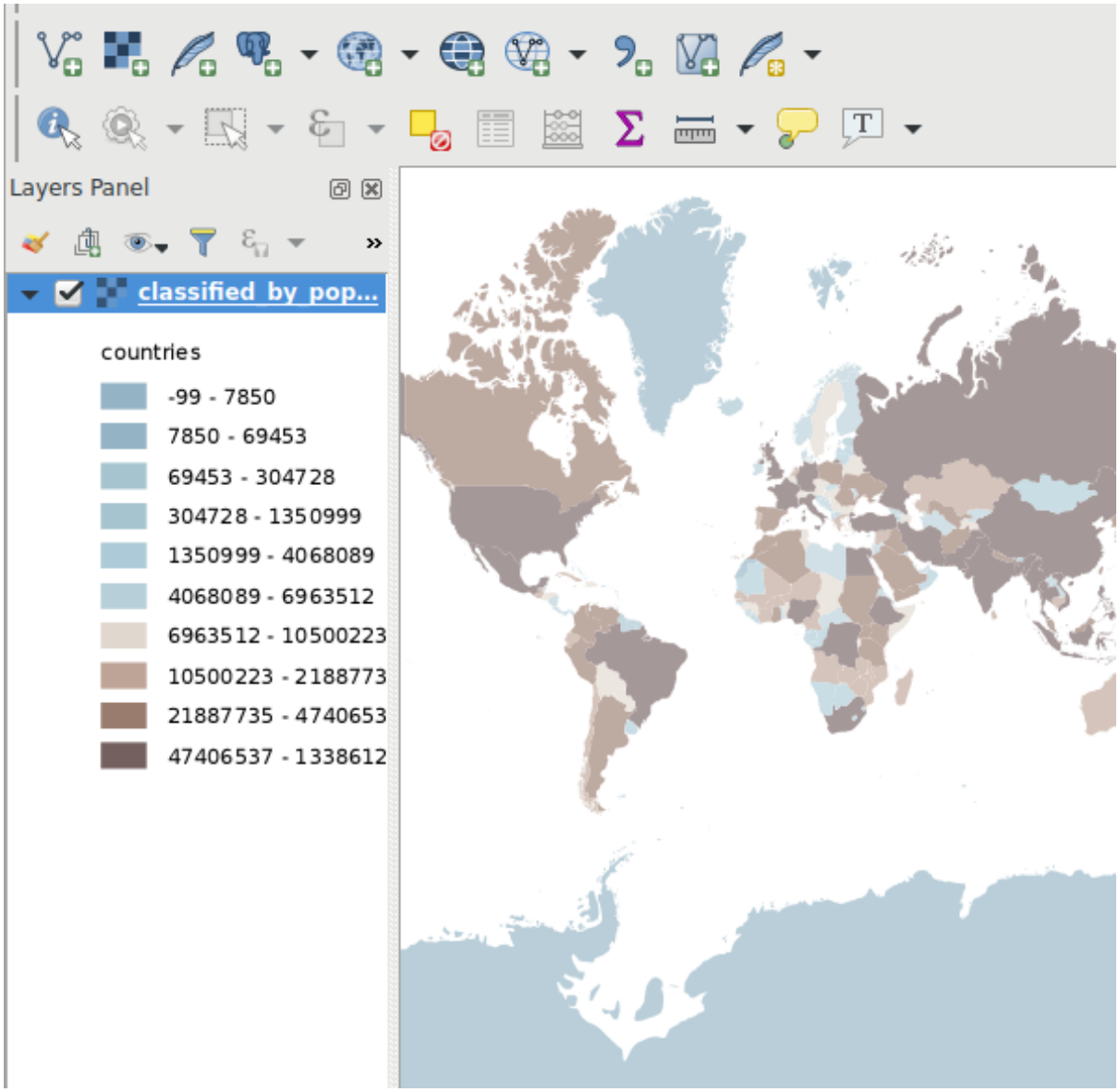


Abb. 11.1: QGIS Desktop consuming the QGIS Server countries layer WMS service

Bemerkung: 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 Logging

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:

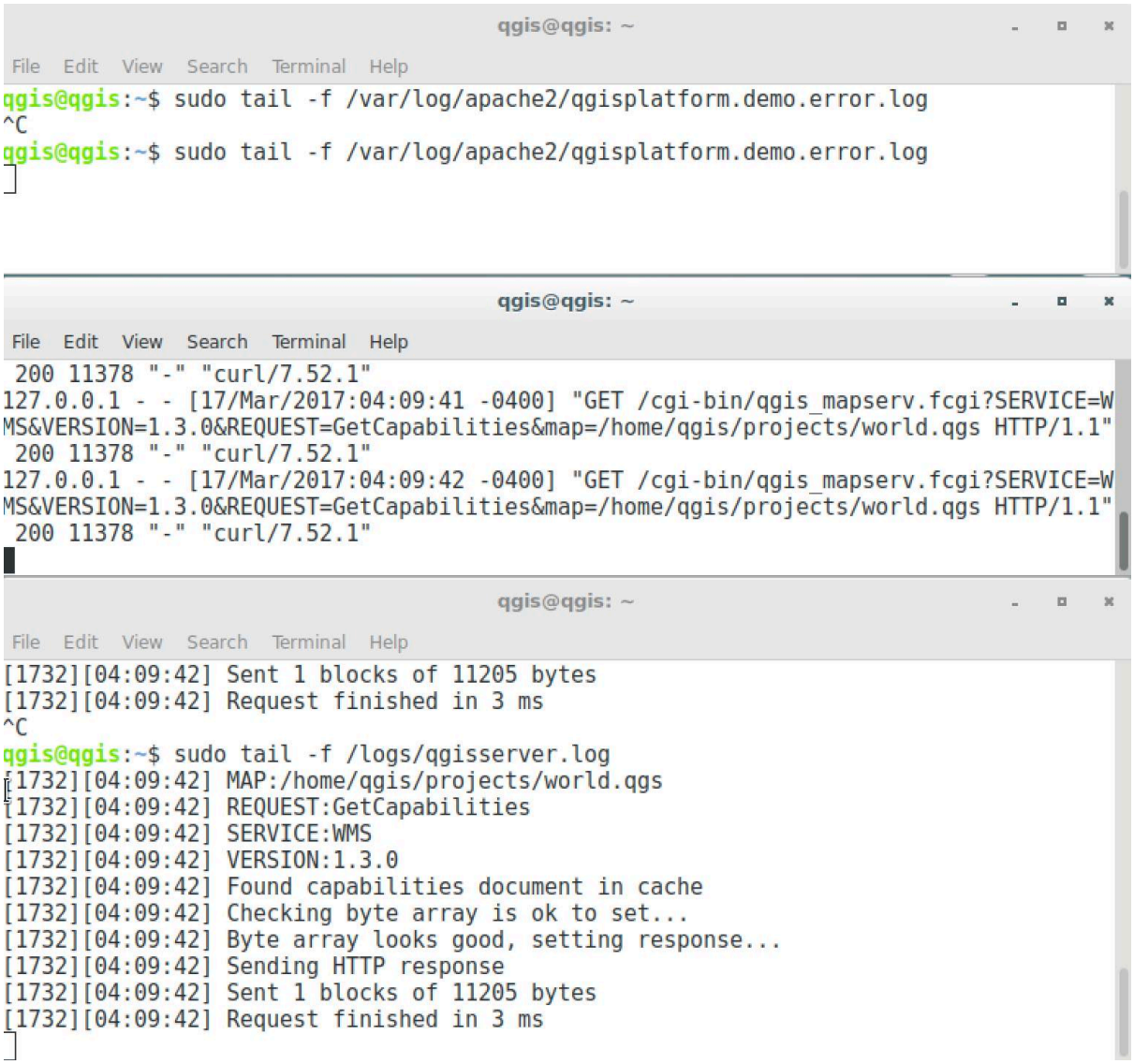


Abb. 11.2: Using the `tail` command to visualise QGIS Server logs output

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.

Bemerkung:

- 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

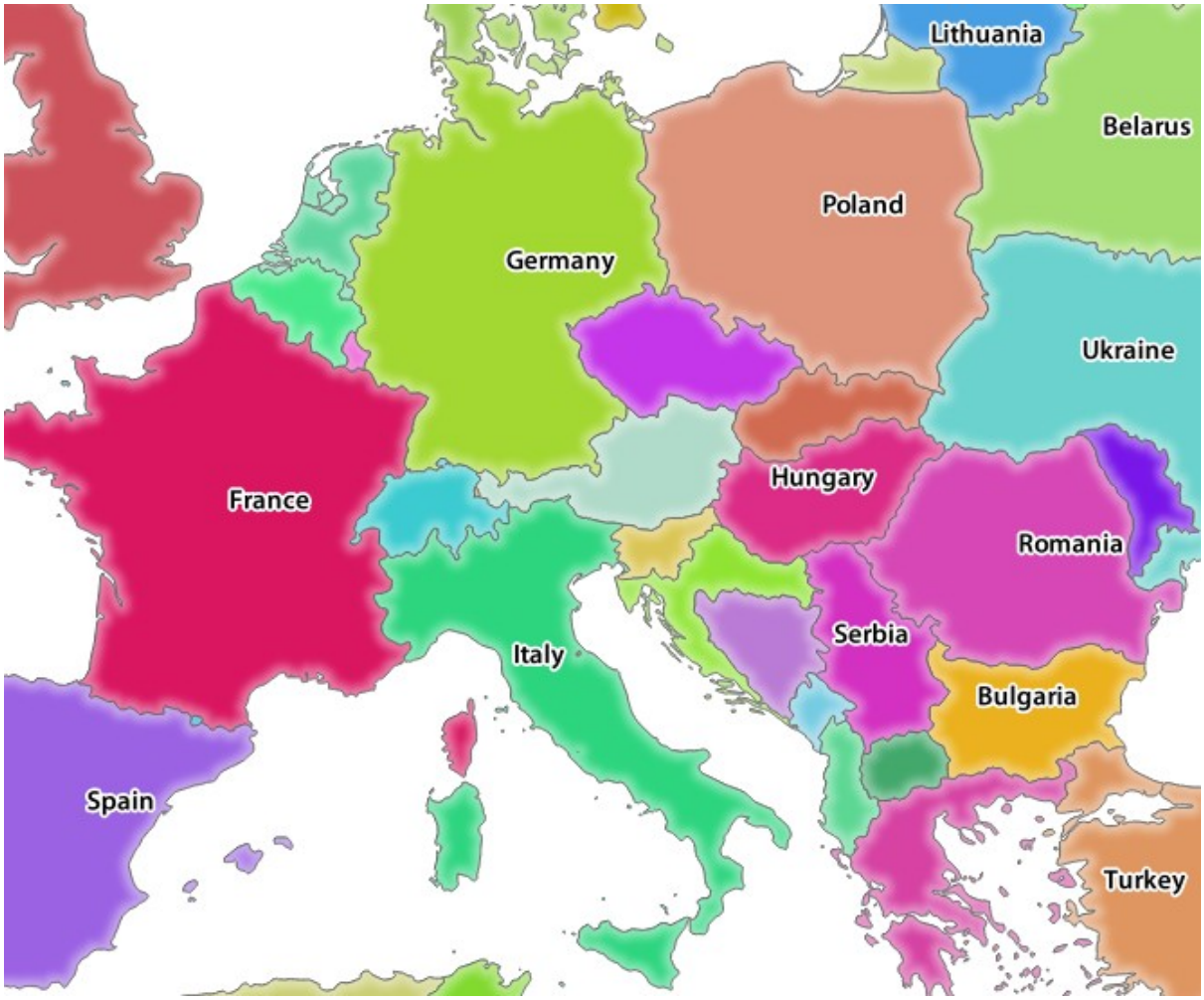


Abb. 11.3: Qgis Server response after a simple GetMap request

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:

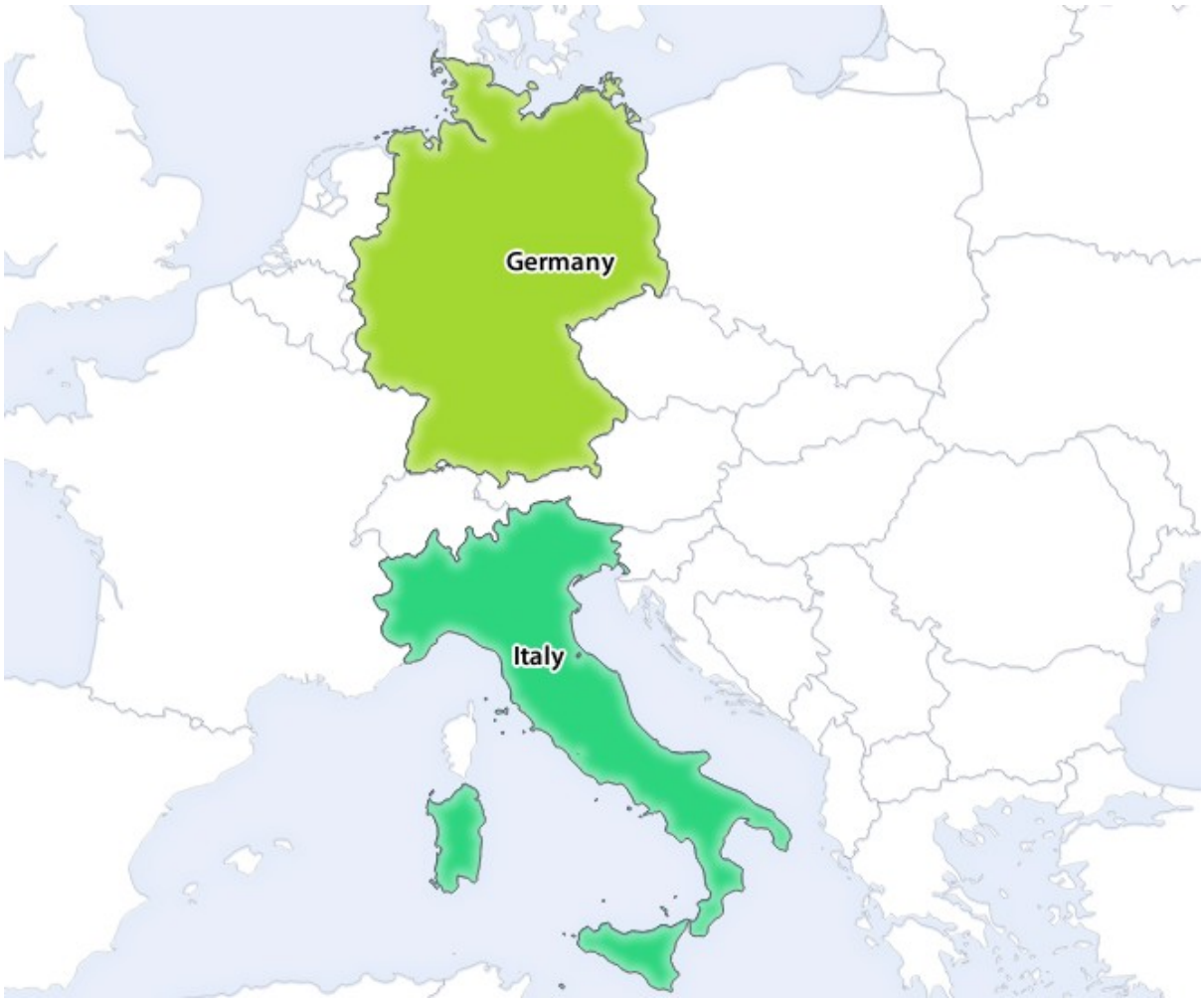


Abb. 11.4: Response to a GetMap request with FILTER and OPACITIES parameters

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:

Abb. 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
```

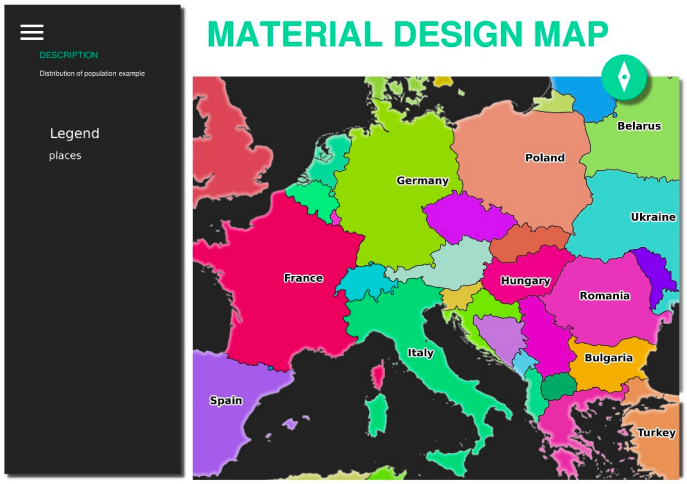


Abb. 11.6: Shows the pdf resulted from the above GetPrint request

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:

Abb. 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.

KAPITEL 12

Module: GRASS

GRASS (Geographic Resources Analysis Support System) ist ein bekanntes Open Source GIS mit einem breiten Feld an GIS Funktionen. Es wurde zuerst 1984 veröffentlicht und wurde seit dem vielfältig verbessert und in der Funktionalität erweitert. QGIS erlaubt es, die mächtigen GRASS Werkzeuge direkt einzusetzen.

12.1 Lesson: GRASS-Einrichtung

Die Nutzung von GRASS in QGIS erfordert etwas Umdenken in Bezug auf die Benutzerschnittstelle. Man arbeit nicht direkt in QGIS sonder in GRASS *über* QGIS. Stellen Sie daher sicher, dass QGIS Desktop mit GRASS Unterstützung installiert ist.

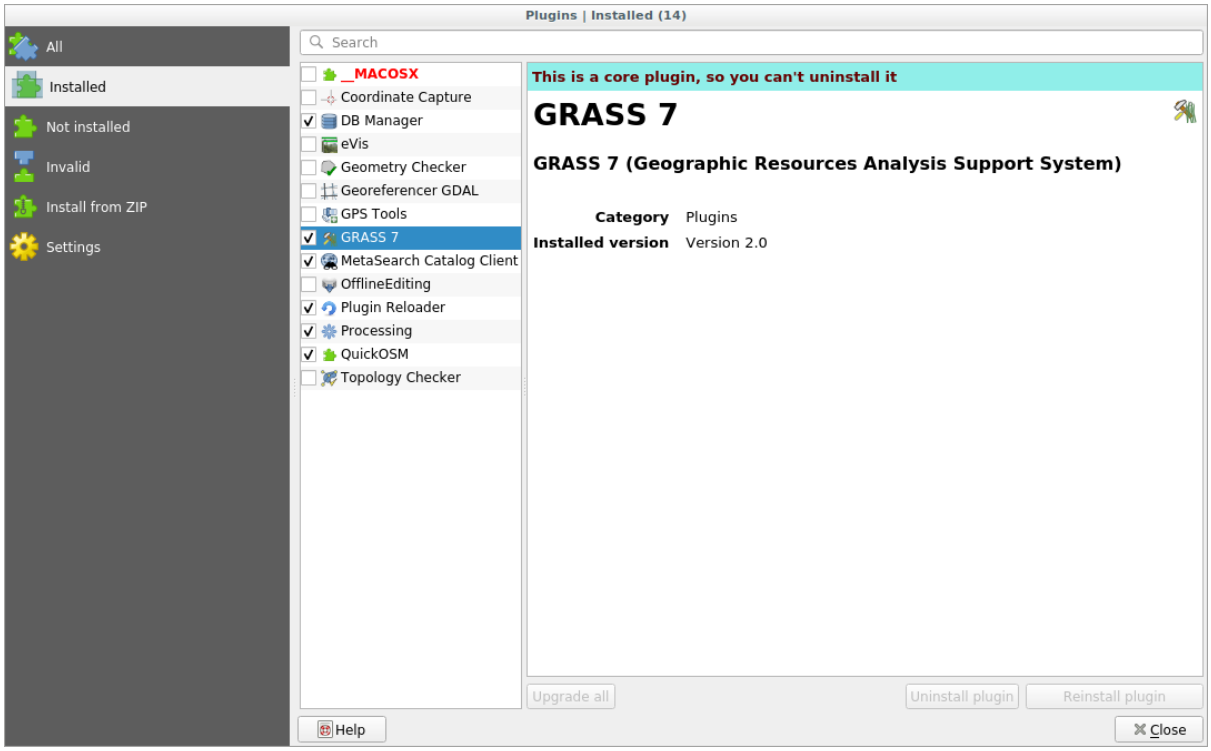
To open a QGIS session with GRASS available on Windows you have to click on the `QGIS Desktop with GRASS` icon.

Ziel dieser Übung: Erstellung eines GRASS Projektes in QGIS.

12.1.1 Follow Along: Start einer neuen GRASS Sitzung

To launch GRASS from within QGIS, you need to activate it as with any other plugin:

1. Öffne zuerst ein neues QGIS Projekt.
2. Wir schalten dazu im *Plugin Manager* die Erweiterung *GRASS* ein:



Die GRASS Werkzeugleiste und das GRASS Bedienfeld werden nun angezeigt:

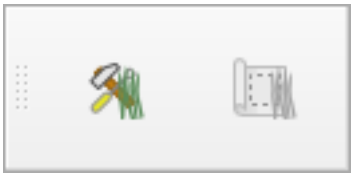


Abb. 12.1: GRASS Werkzeugleiste

Abb. 12.2: GRASS Bedienfeld

Das GRASS Bedienfeld ist noch nicht aktiv. Vorher muss ein `Mapset` erstellt werden. GRASS arbeit immer in einer Datenbankumgebung. Das bedeutet, dass die Daten die verwendet werden sollen, in die Datenbank importiert werden müssen.

Die GRASS Datenbank hat eine überschaubare Struktur, selbst wenn sie auf den ersten Blick sehr kompliziert wirkt. Am Wichtigsten ist es zu verstehen, dass die oberste Ebene der Datenbank die `Location` ist. Jede `Location` kann mehrere verschiedene `Mapset` enthalten: in **jedem** `Mapset` ist das `PERMANENT` Mapset enthalten. Es wird standardmäßig von GRASS erstellt. Jedes `Mapset` enthält die Daten (Raster, Vektor usw.) in einer bestimmten Struktur, aber keine Sorge, darum kümmert sich GRASS.

Just remember: `Location` contains `Mapset` that contains the data. For more information visit the GRASS website.

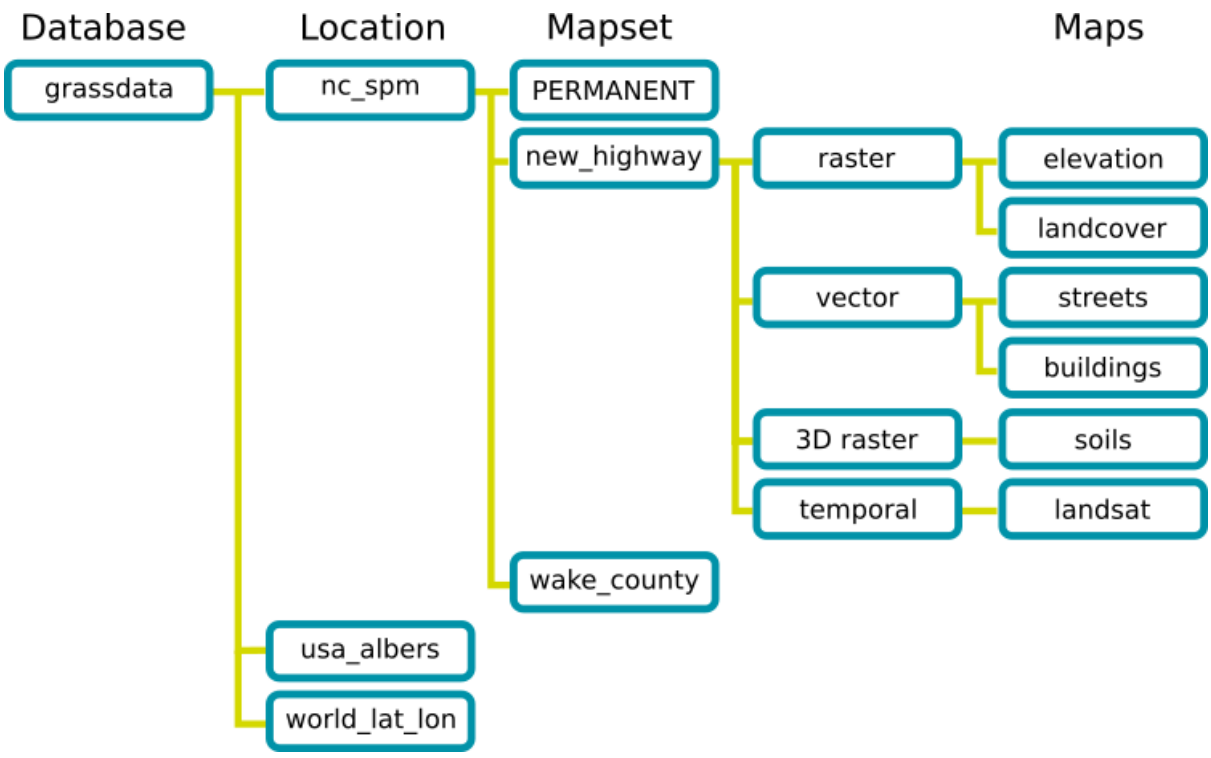
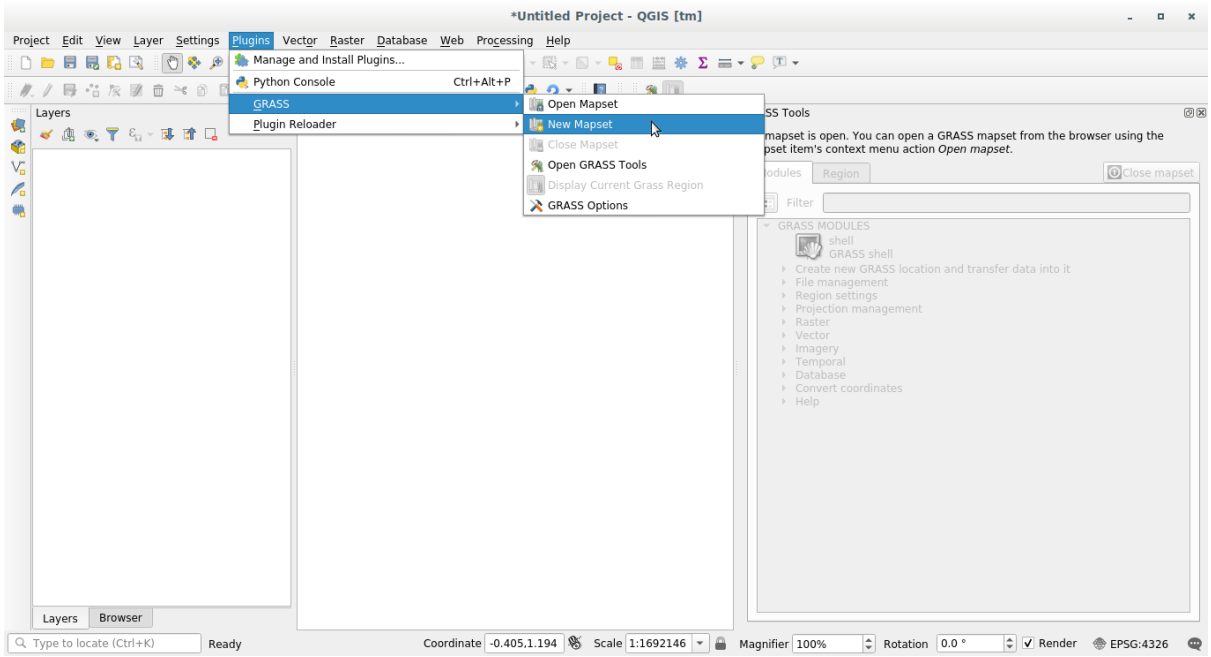


Abb. 12.3: Die GRASS Datenbankstruktur (aus der GRASS Hilfe)

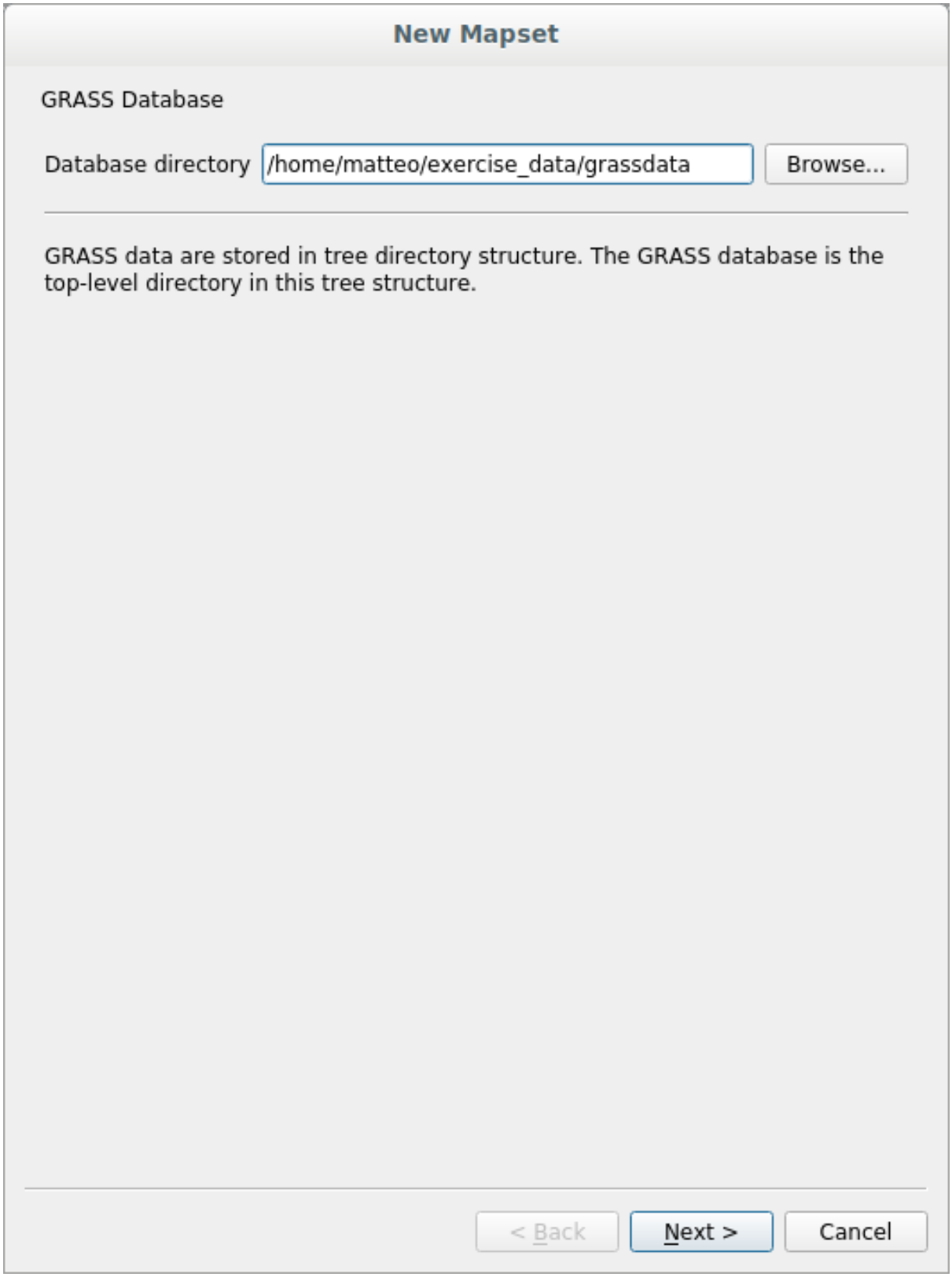
12.1.2 Follow Along: Starten eines neuen GRASS Projektes

1. Click on the *Plugins* ► *GRASS* ► *New Mapset* menu:



   Du wirst nun nach dem GRASS Datenbankverzeichnis gefragt.

2. Wir geben das Verzeichnis ein, das GRASS für die Einrichtung seiner Datenbank verwenden wird.



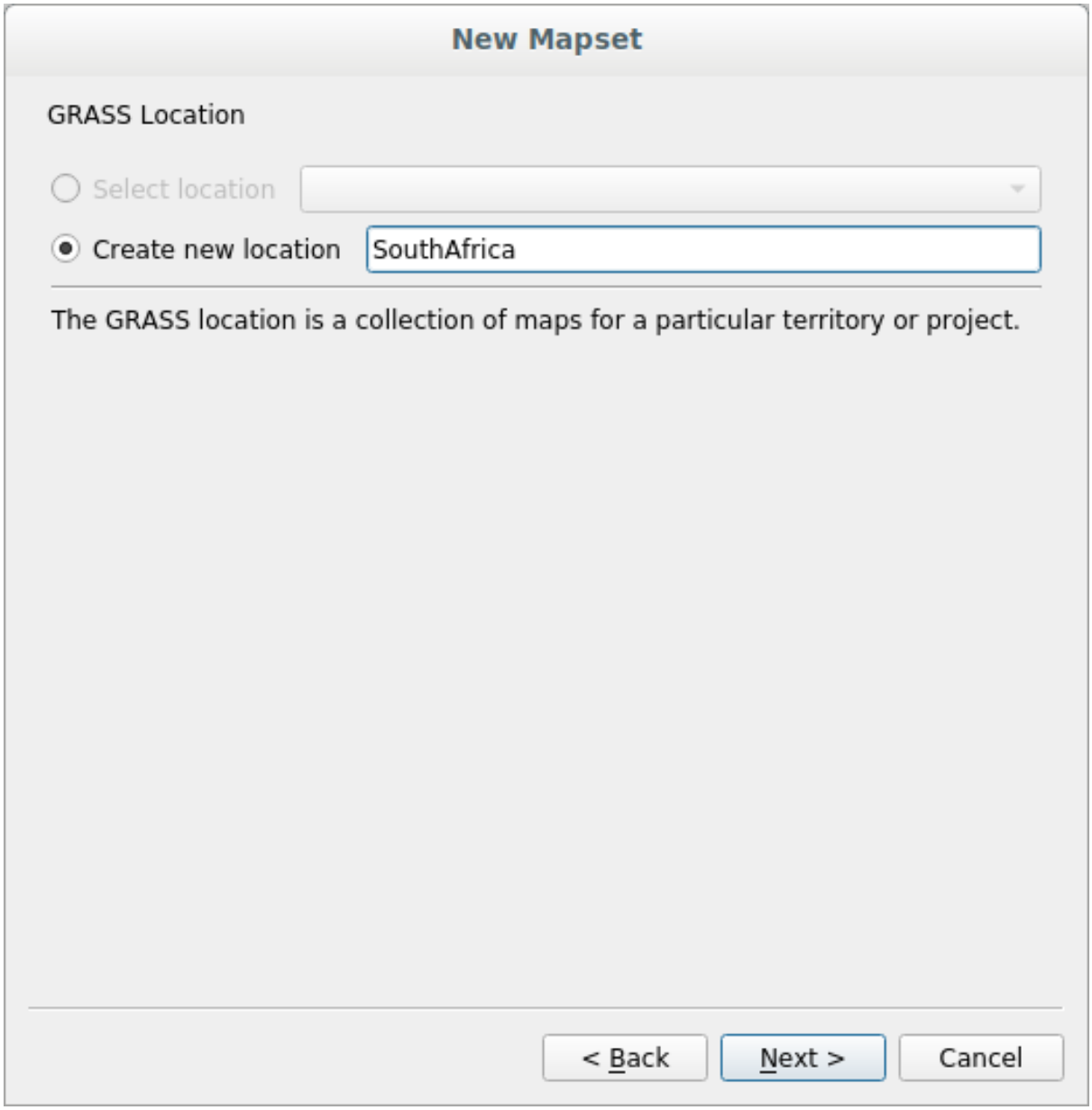
3. Klicke auf *Weiter*.

GRASS muss eine `Location` erstellen, die die maximale Ausdehnung des Arbeitsgebietes beschreibt und auch GRASS `Region` genannt wird.

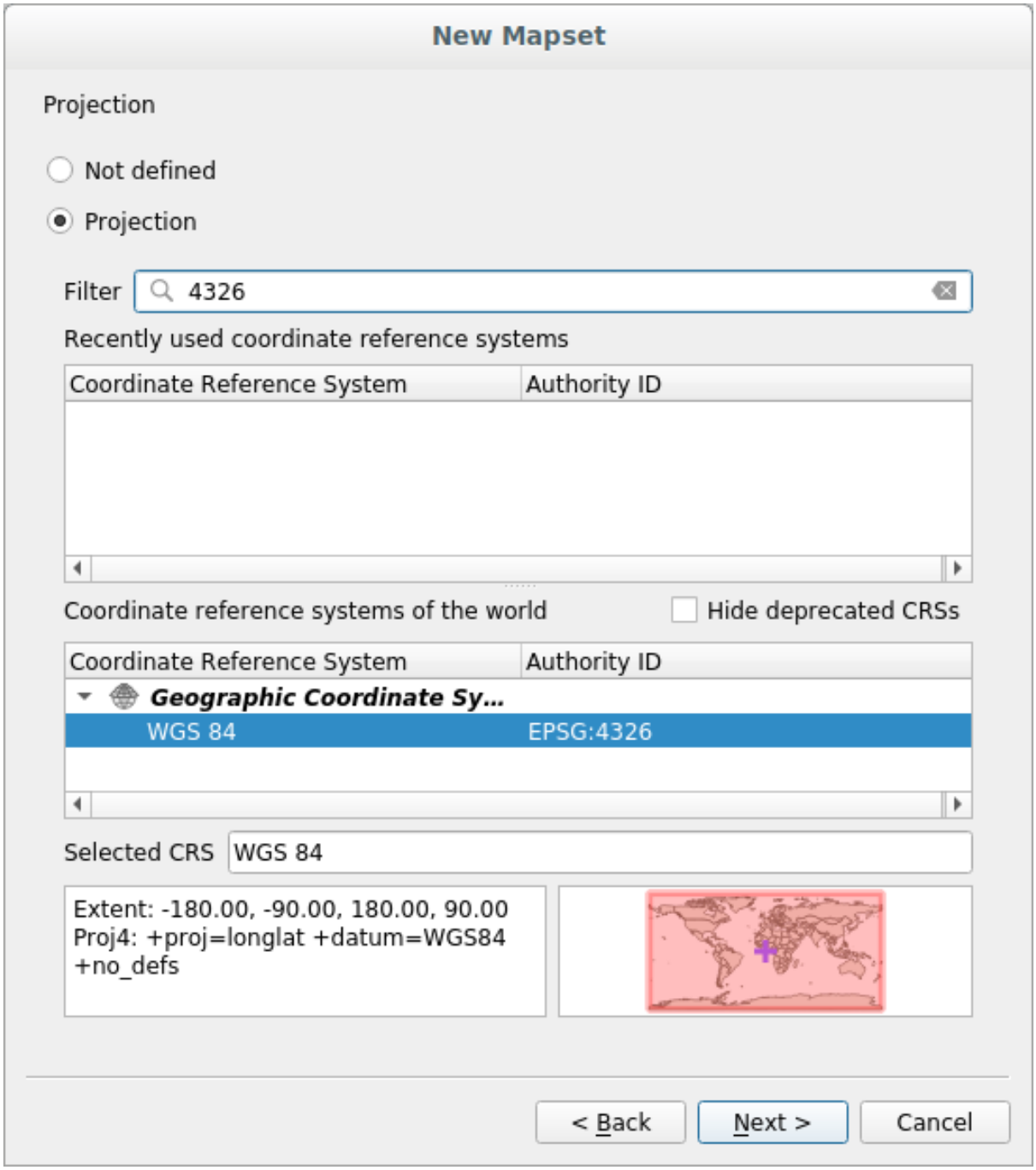
Bemerkung: Die Region ist extrem wichtig für GRASS, da sie das Gebiet beschreibt in dem alle Layer von GRASS

berücksichtigt werden. Alles außerhalb wird nicht berücksichtigt. Keine Sorge, man kann die Ausdehnung der GRASS Region immer auch nach der Erstellung ändern.

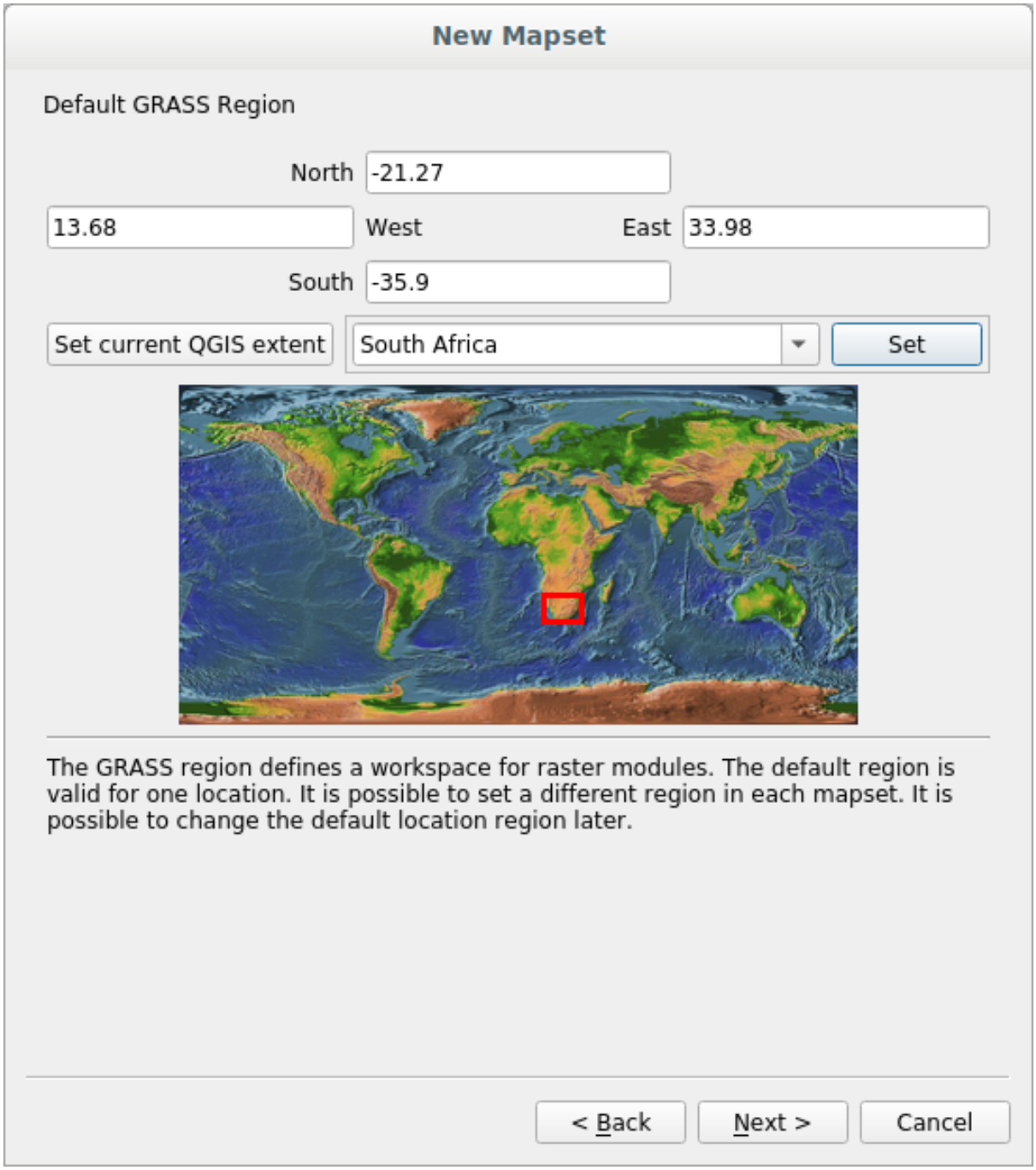
1. Wir nennen die neue location `SouthAfrica`:



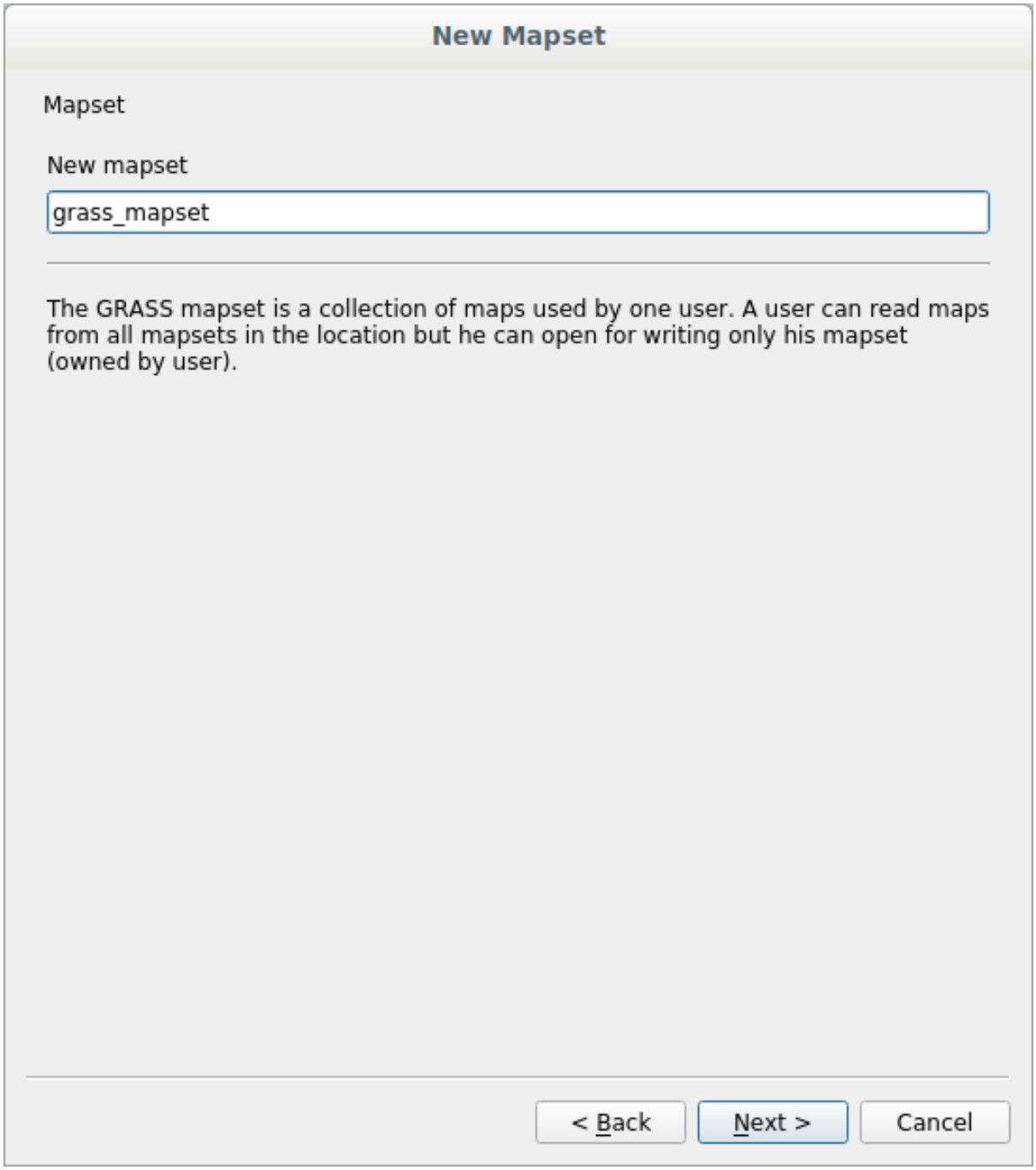
2. Klicke auf *Weiter*.
3. Wir werden mit `WGS 84` arbeiten, d.h. wir suchen und wählen dieses KBS:



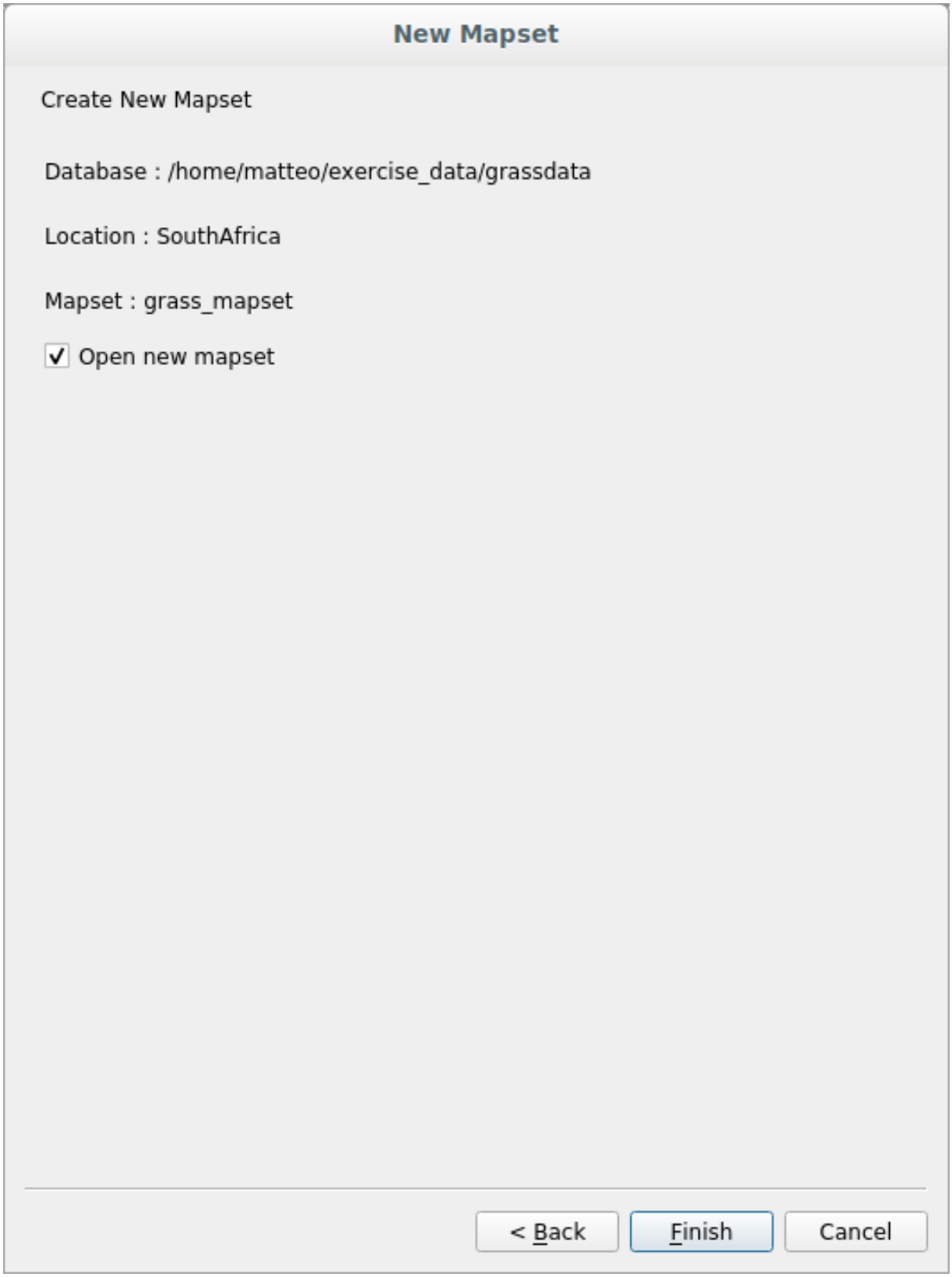
4. Klicke auf *Weiter*.
5. Wähle nun die Region *South Africa* aus dem dropdown Menü und klicke auf *Setzen*:



6. Klicke auf *Weiter*.
7. Wir erstellen ein mapset, d.h. die Kartendatei in der wir arbeiten.



Wenn wir fertig sind, wird ein Dialogfenster mit einer Zusammenfassung der eingegebenen Informationen angezeigt.



8. Klicke auf *Abschließen*.
9. Klicke in der Erfolgsmeldung auf *OK*.

Das GRASS Bedienfeld wird jetzt aktiviert und wir können alle GRASS Werkzeuge nutzen.

12.1.3 Follow Along: Vektordaten in GRASS laden

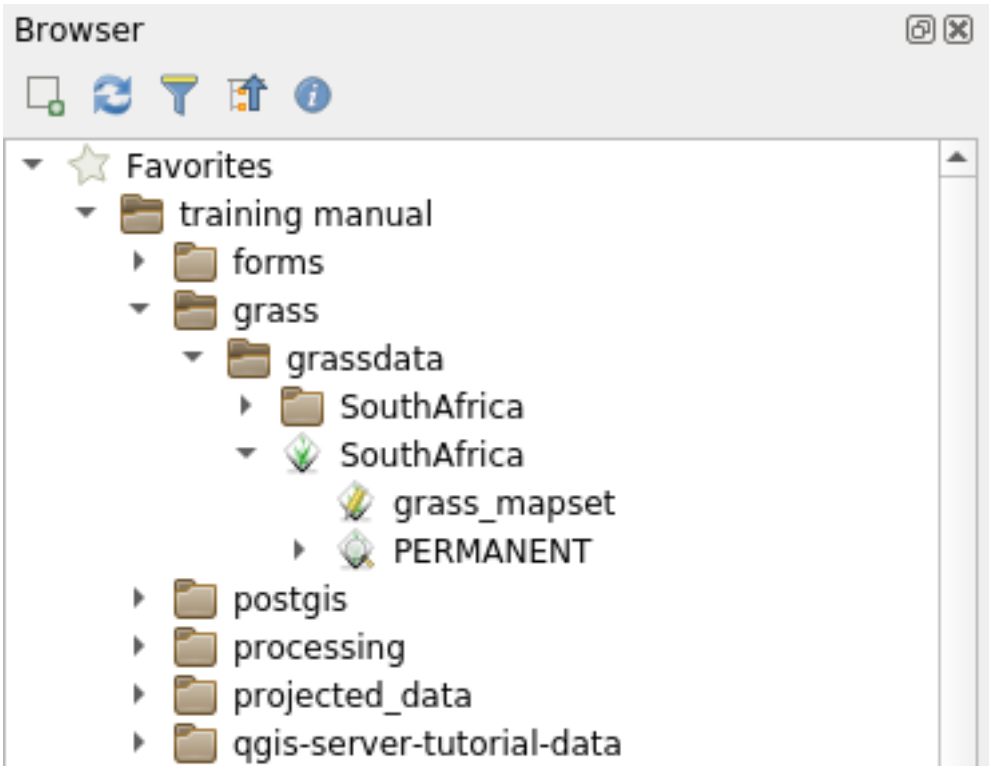
Wir haben jetzt eine leere Karte. Bevor wir die GRASS Werkzeuge nutzen können, müssen wir Daten in die GRASS Datenbank laden, d.h. in das `Mapset`. Man kann GRASS Werkzeuge nicht mit Layern verwenden, die nicht in einem GRASS `Mapset` geladen sind.

Es gibt verschiedene Möglichkeiten, Daten in die GRASS Datenbank zu laden. Wir beginnen mit der Ersten.

Follow Along: Daten mit Hilfe des QGIS Browsers laden

Im Kapitel *The Browser Panel* sahen wir, dass das Browser Bedienfeld der einfachste Weg ist, um Daten in QGIS zu laden.

GRASS Daten werden vom QGIS Browser als *echte* GRASS Daten erkannt. Dies erkennt man am GRASS Icon neben dem GRASS Mapset. Außerdem wird das Icon neben einem geöffneten Mapset angezeigt.

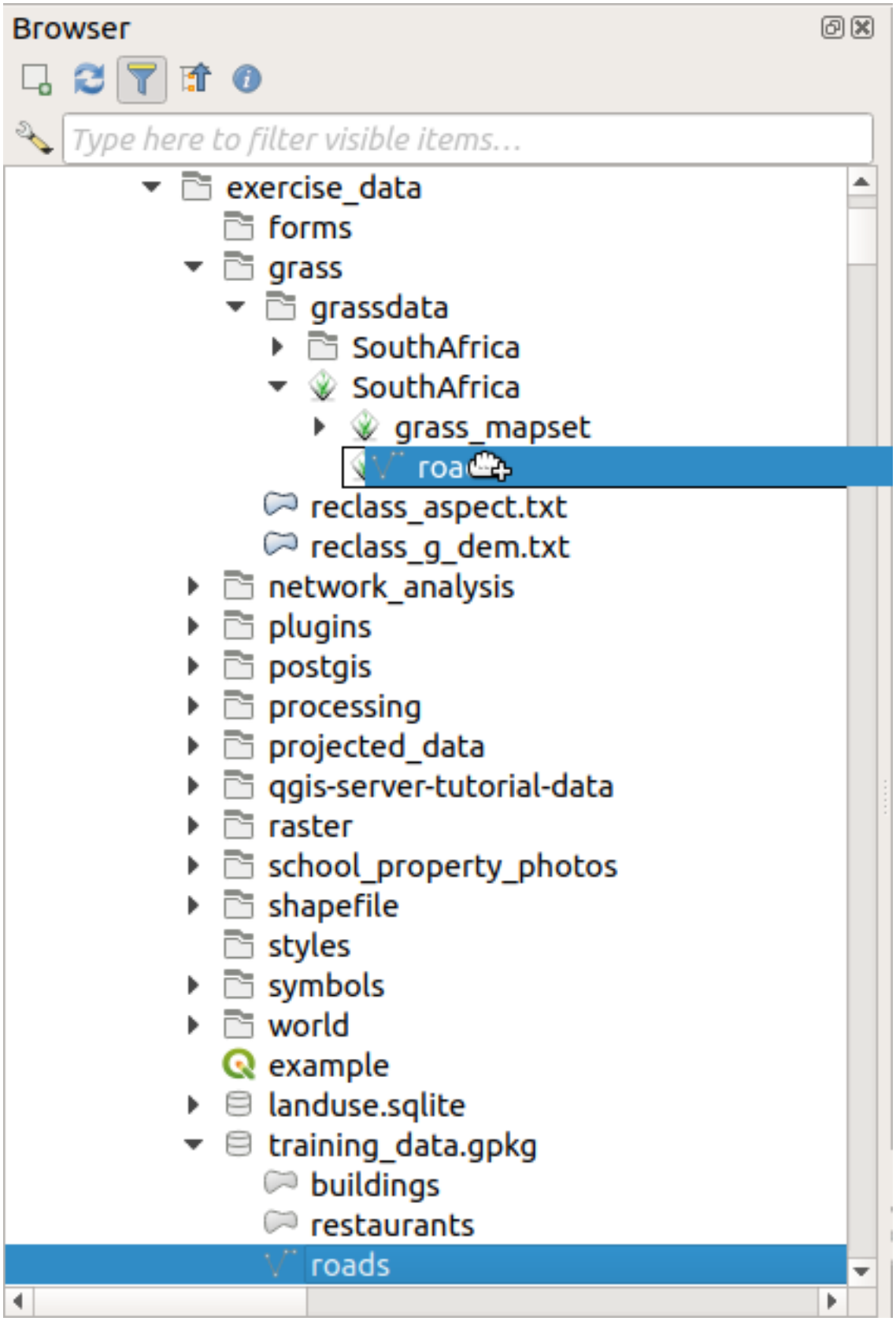


Bemerkung: Man sieht eine Kopie der GRASS Location als normalen Ordner: die GRASS `Mapset` Daten liegen im Ordner

Layer aus einem Ordner können mit **drag and drop** einfach in das GRASS Mapset gezogen werden.

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.



Das wars schon. Wenn man das Mapset erweitert, sieht man den importierten Layer `roads`. Man den importierten Layer nun wie jeden anderen Layer in QGIS laden.

Tipp: Man kann außerdem Layer direkt aus dem Layerinhaltsverzeichnis in ein Mapset im Browser Bedienfeld

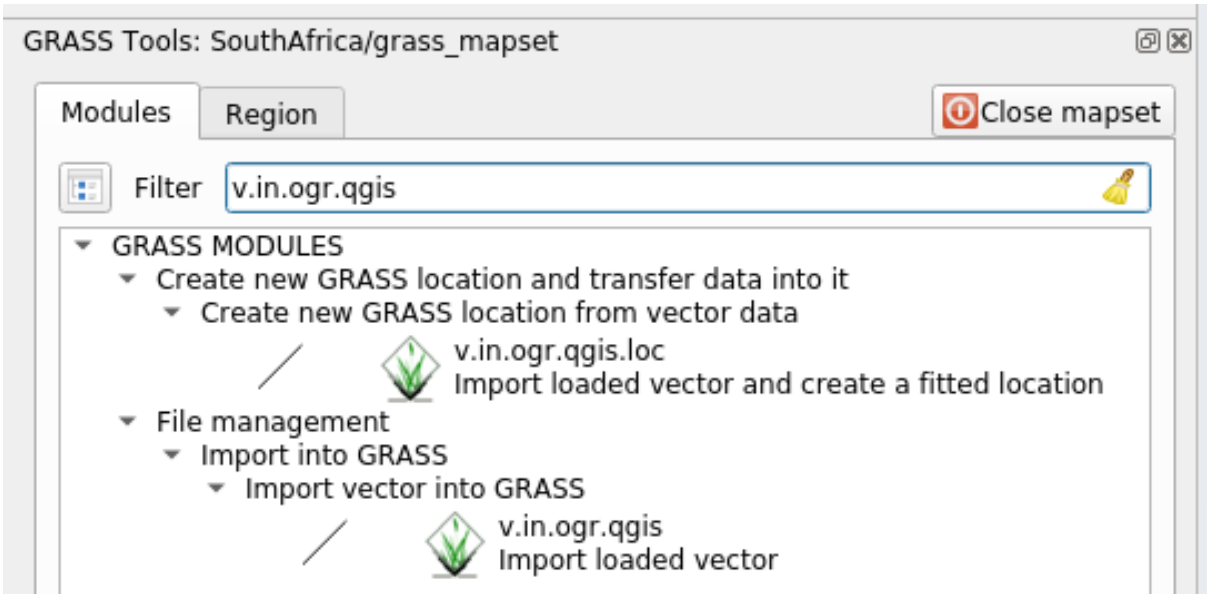
ziehen. Damit wird der Workflow stark vereinfacht!

Follow Along: Daten mit Hilfe des GRASS Bedienfeldes laden

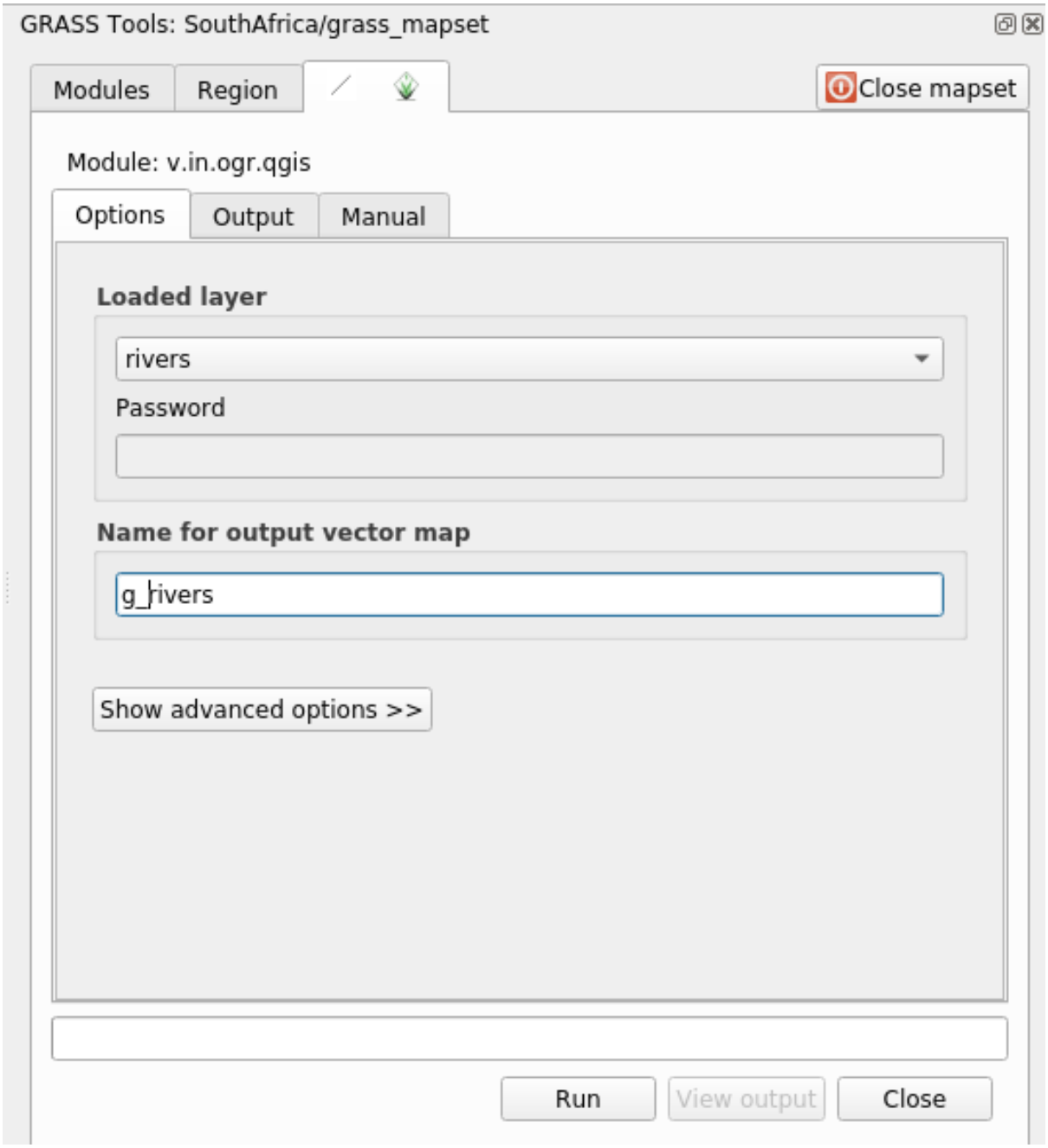
Wir werden nun die *lange* Methode zum laden des Layers `rivers.shp` in das Mapset nutzen.

1. Wir laden wie gewohnt Daten in QGIS und verwenden den Datensatz `rivers.shp` (zu finden im Ordner `exercise_data/shapefile/`).
2. Sobald die Daten in QGIS geladen sind, klicken wir auf den *Filter* Kasten im `GRASS Bedienfeld` und suchen das Vektorimportwerkzeug indem wir `v.in.ogr.qgis` im Filterkasten eingeben:

   Warnung: Es werden 2 ähnliche Werkzeuge angezeigt: `v.in.ogr.qgis` und `v.in.ogr.qgis.loc`. Wir verwenden das **erste** Werkzeug.



   Das `v` steht für *Vektor*, `in` steht für die Funktion Daten in die GRASS Datenbank zu importieren, `ogr` ist die Softwarebibliothek die zum Laden von Vektordaten verwendet wird und `qgis` bedeutet, dass das Werkzeug nach bereits in QGIS geladenen Layern sucht.
3. Nachdem wir das Werkzeug gefunden haben, klicken wir darauf, um es zu öffnen. Wählen Sie den Layer *rivers* aus der Auswahlbox *Geladener Layer* und vergeben für die Ausgabe den Namen `g_rivers`, um Verwechslungen vorzubeugen:



Bemerkung: Beachten Sie, dass mit dem Knopf *Fortgeschrittene Optionen einblenden* weitere Optionen verfügbar sind. Hier ist Möglichkeit der Einschränkung des Datenimports mit Hilfe einer WHERE Klausel in der zum Import genutzten SQL Abfrage vorhanden.

---

4. Klicken Sie auf *Starten* um den Import zu beginnen.
5. Drücken Sie nach Abschluss des Imports auf den Knopf *Ergebnis visualisieren*, um den in GRASS importierten Layer in der Karte anzuzeigen.
6. Schließen Sie zuerst das Importwerkzeug (Klick auf den Knopf *Schließen* im rechten Bereich neben dem Knopf *Ergebnis visualisieren*) und danach das *GRASS Werkzeuge* Fenster.
7. Entfernen Sie das Original des Layers *rivers*.

In unserer QGIS Karte wird jetzt nur noch der in GRASS importierte Layer angezeigt.

12.1.4 Follow Along: Rasterdaten in GRASS laden

Rasterdaten importiert man auf dieselbe Weise wie Vektordaten.

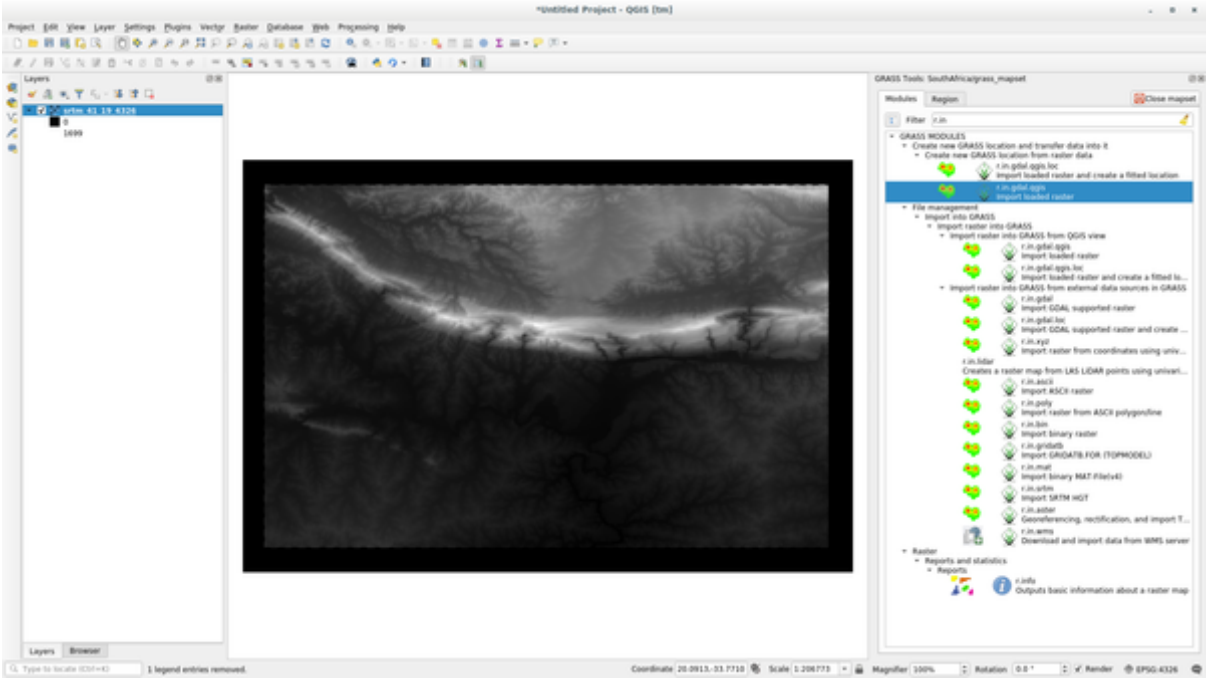
Wir werden in das GRASS Mapset den Layer `srtm_41_19_4326.tif` importieren.

Bemerkung: Der Rasterlayer liegt schon im richtigen KBS `WGS 84` vor. Layer mit abweichendem KBS müssen vorher in das KBS des GRASS Mapsets reprojiziert weden.

---

1. Laden Sie den Layer `srtm_41_19_4326.tif` in QGIS
2. Öffnen Sie erneut das Fenster *GRASS Werkzeuge*.
3. Klicken Sie auf den Reiter *Module*.
4. Suchen Sie nach `r.in.gdal.qgis` und klicken doppelt auf das Werkzeug, um sein Dialogfenster zu öffnen.
5. Geben Sie für den Eingabelayer `srtm_41_19_4326.tif` und für den Ausgabelayer `g_dem` vor.

6. Klicken Sie auf *Starten*.
7. Klicken Sie nach dem Abschluss des Prozesses auf *Ergebnis visualisieren*.
8. *Schließen* Sie den Reiter und nocheinmal mit *Schließen* den gesamten Dialog.



9. Wir können den originalen Layer `srtm_41_19_4326.tif` nun entfernen.

12.1.5 Try Yourself Layer zum Mapset hinzufügen

Versuchen Sie die Vektorlayer `water.shp` und `places.shp` aus dem Ordner `exercise_data/shapefile/` in das GRASS Mapset zu importieren. Benennen Sie die Layer wie beim vorherigen Import mit `g_water` und `g_places` um einer Verwechselung vorzubeugen.

Answer

You can add layers (both vector and raster) into a GRASS Mapset by drag and drop them in the Browser (see *Follow Along: Daten mit Hilfe des QGIS Browsers laden*) or by using the `v.in.gdal.qgis` for vector and `r.in.gdal.qgis` for raster layers.

12.1.6 Öffnen eines vorhandenen Mapsets

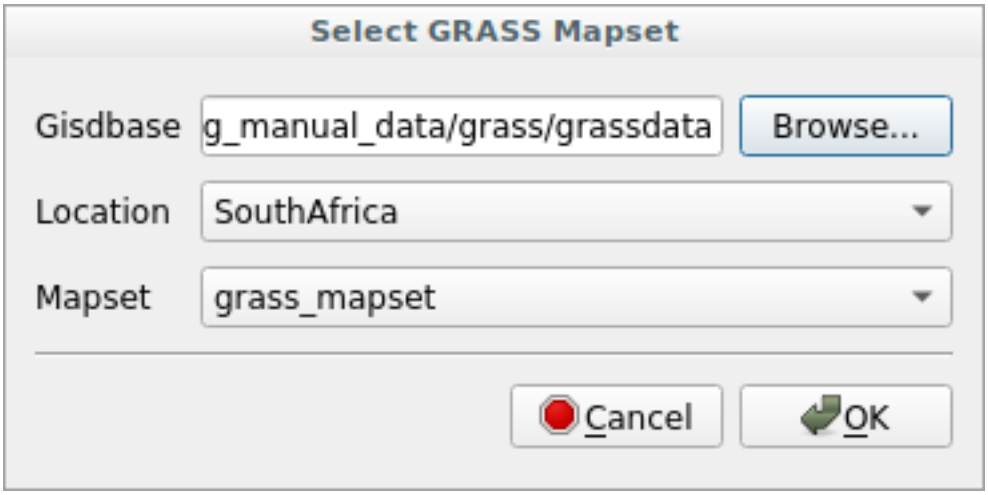
Wenn bereits ein GRASS Mapset vorliegt, können Sie es einfach in einer anderen QGIS-Sitzung öffnen.

Es gibt verschiedene Methoden, ein GRASS Mapset zu öffnen. Wir werden nun einige untersuchen.

Wir shcließen zuerst das Mapset durch Klick auf die Schaltfläche *Schließe Mapset* im Fenster *GRASS-Werkzeuge*.

Follow Along: Benutzung des GRASS Plugins

1. Klicken Sie im Menü auf *Erweiterungen –> GRASS –> Mapset öffnen* über dem Menüeintrag *Plugins –> GRASS –> Neues Mapset* den wir in der vorherigen Übung verwendet haben.
2. Gehen Sie zum Ordner mitt der GRASS Datenbank: Seien Sie vorsichtig! Man muss den übergeordneten Ordner auswählen, nicht den Ordner mit dem GRASS Mapset. GRASS liest alle `Locations` der Datenbank und alle `Mapsets` einer `Location` ein:



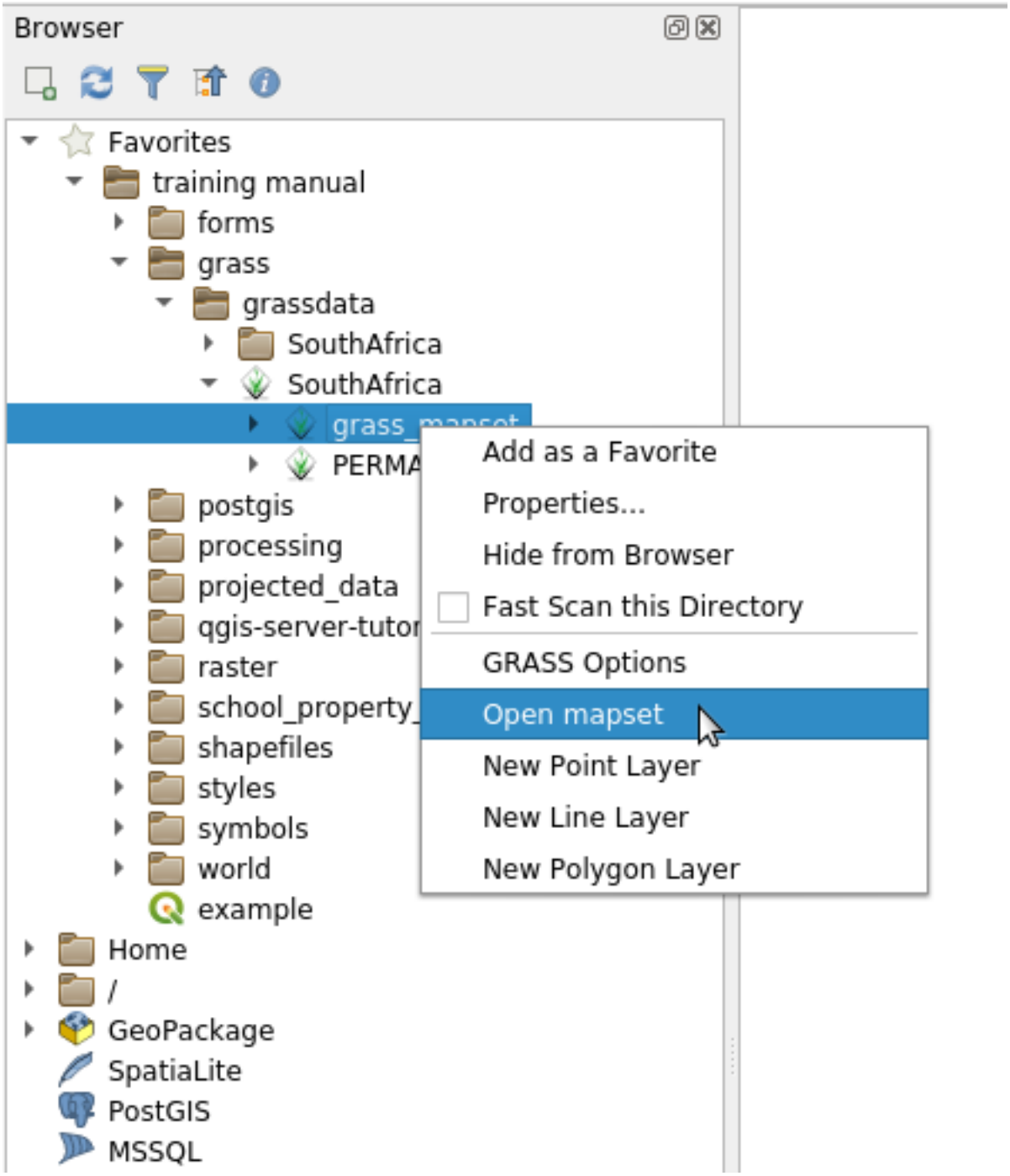
3. Wählen Sie die `Location` *SouthAfrica* und das `Mapset` *grass_mapset* das wir vorher erstellt haben.

Das war's. Das GRASS Bedienfeld wird aktiv, was bedeutet, dass das Mapset korrekt geladen wurde.

Follow Along: Verwendung des QGIS Browsers

Noch einfacher kann ein `Mapset` mit Hilfe des QGIS Browsers geöffnet werden:

1. Schließen Sie das Mapset (wenn es noch geöffnet ist) durch Klick auf die Schaltfläche *Schließe Mapset* im *GRASS-Werkzeuge* Fenster.
2. Gehen Sie im QGIS Browser zum Ordner der GRASS Datenbank.
3. Klicken Sie mit der rechten Maustaste auf das Mapset (das Mapset hat das GRASS Icon neben dem Namen). Es werden einige Optionen angezeigt.
4. Klicke Sie auf *Mapset öffnen*:



Das Mapset ist nun geöffent und kann verwendet werden!

Tipp: Nach Klick mit rechten Maustaste auf ein Mapset werden viele verschiedene Optionen abgeboten. Untersuchen Sie auch die anderen Optionen.

12.1.7 In Conclusion

Der Arbeitslauf zum Hinzufügen von Daten in GRASS ist etwas abweichend von der QGIS Methode, weil GRASS seine Daten in eine räumliche Datenbank lädt. Trotzdem kann QGIS als Frontend die Verwaltung eines GRASS Mapsets durch die Verwendung von QGIS Layern als Datenquellen für QGIS vereinfachen.

12.1.8 What's Next?

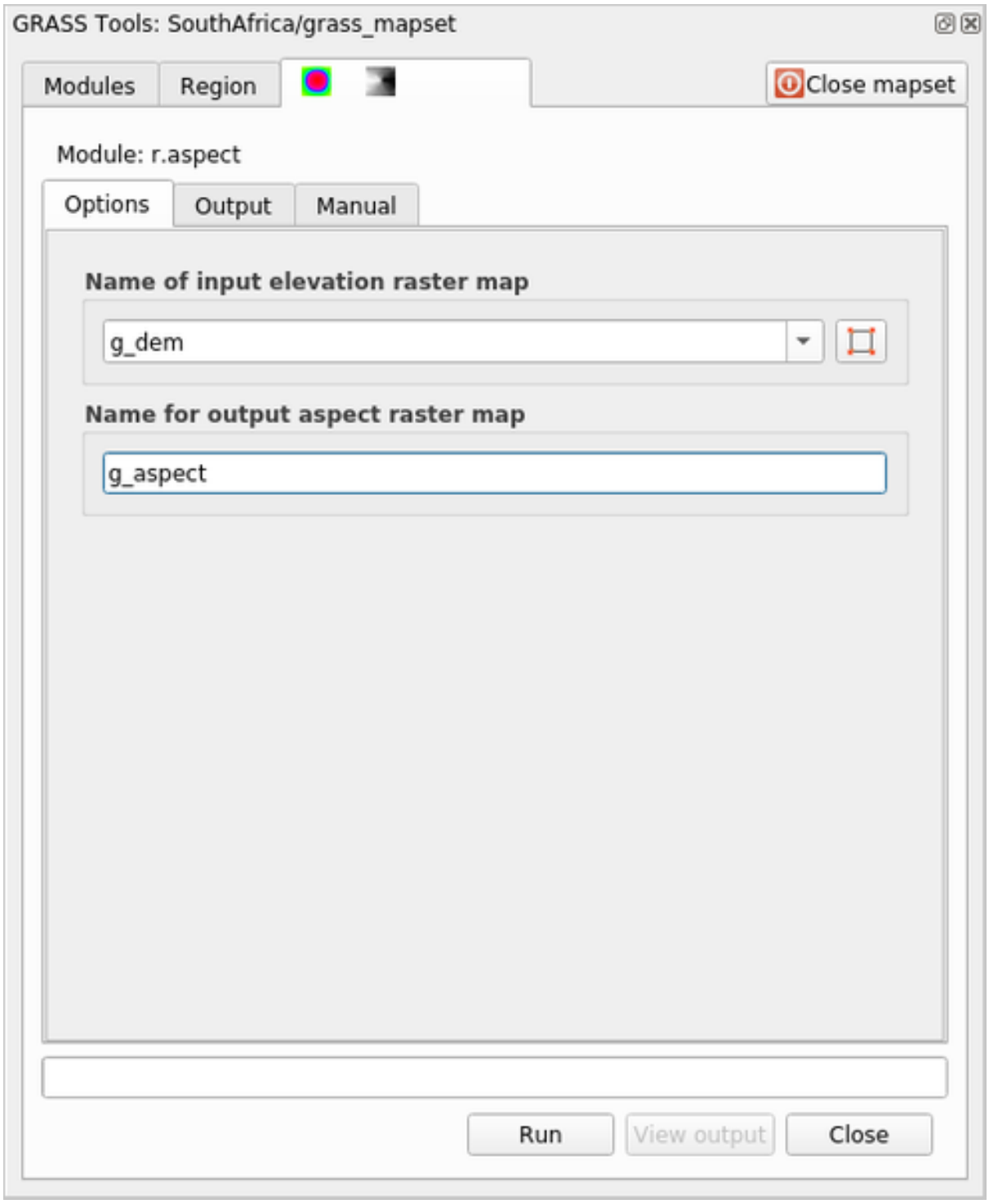
Jetzt wo die Daten in GRASS importiert sind, können wir uns die erweiterten Analysemöglichkeiten ansehen, die GRASS bietet.

12.2 Lesson: GRASS-Werkzeuge

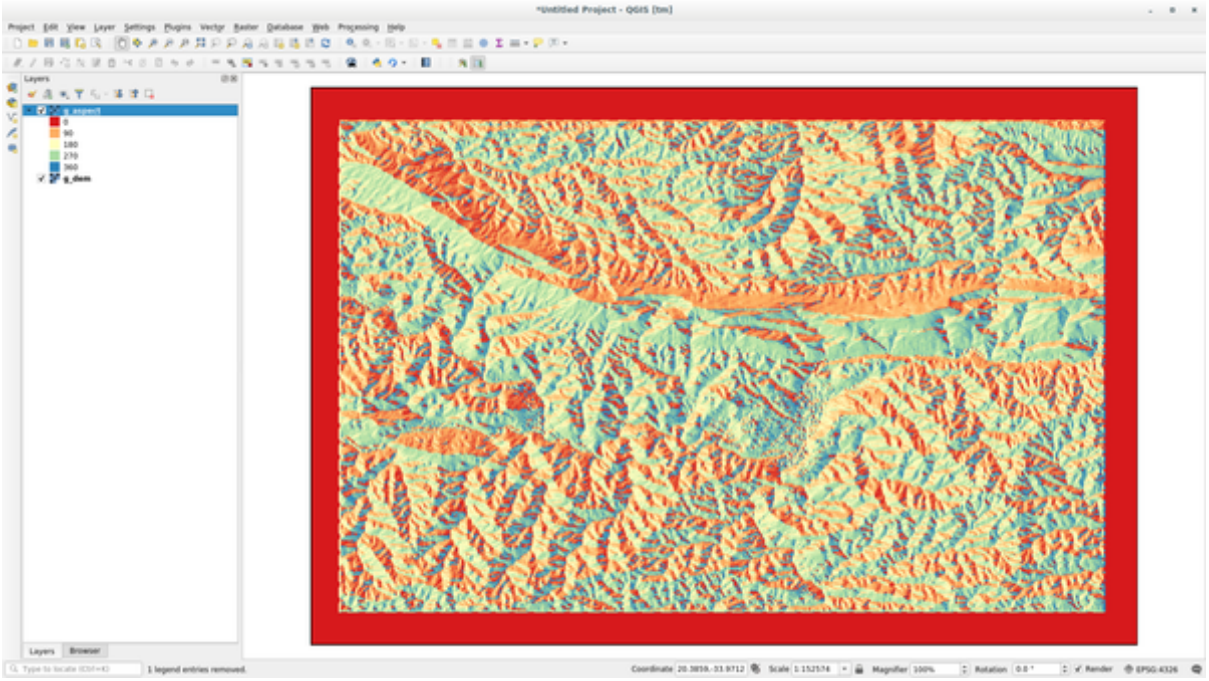
In dieser Übung werden wir eine Auswahl an Werkzeugen vorstellen, um einen Eindruck der Fähigkeiten von GRASS zu erhalten.

12.2.1 Follow Along: Erstellung einer Karte der Expostition

1. Öffnen Sie das Dialogfenster *GRASS-Werkzeuge*
2. Laden Sie den Rasterlayer `g_dem` aus dem Mapset *grass_mapset*.
3. Suchen Sie das Modul *r.aspect* durch Eingabe in das *Filter* Feld im *Module* Reiter.
4. Öffnen Sie das Werkzeug, tragen sie die Werte wie hier dargestellt ein und klicken Sie auf die Schaltfläche *Starten*:



5. Klicken Sie nach Abschluss des Prozesses auf *Ergebnis visualisieren* um den erzeugten Layer im Arbeitsbereich anzuzeigen:

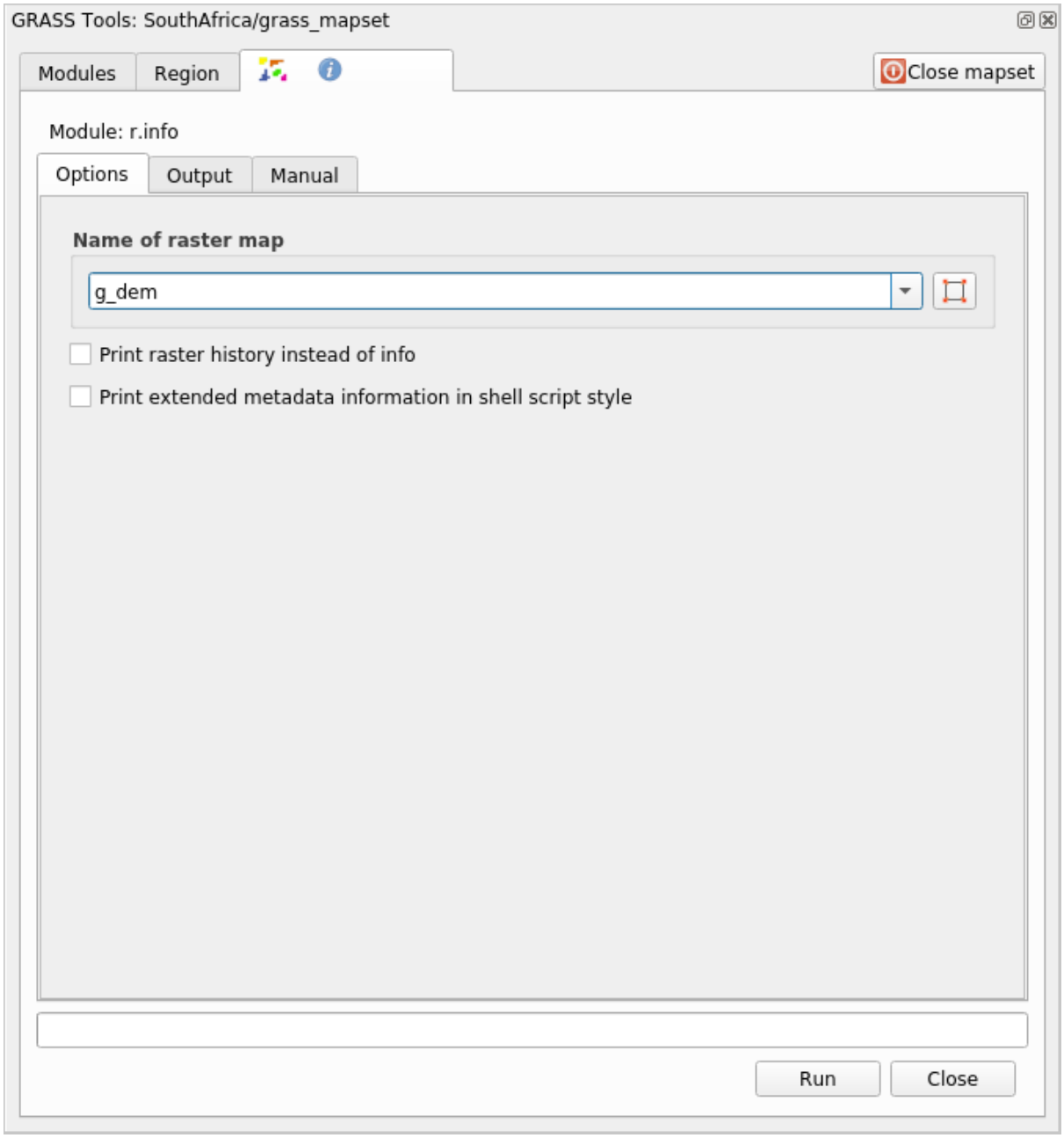


Der Layer `g_aspect` wird im Mapset *grass_mapset* gespeichert, so dass man den Layer aus der Karte entfernen und neu einladen kann, wann immer er benötigt wird.

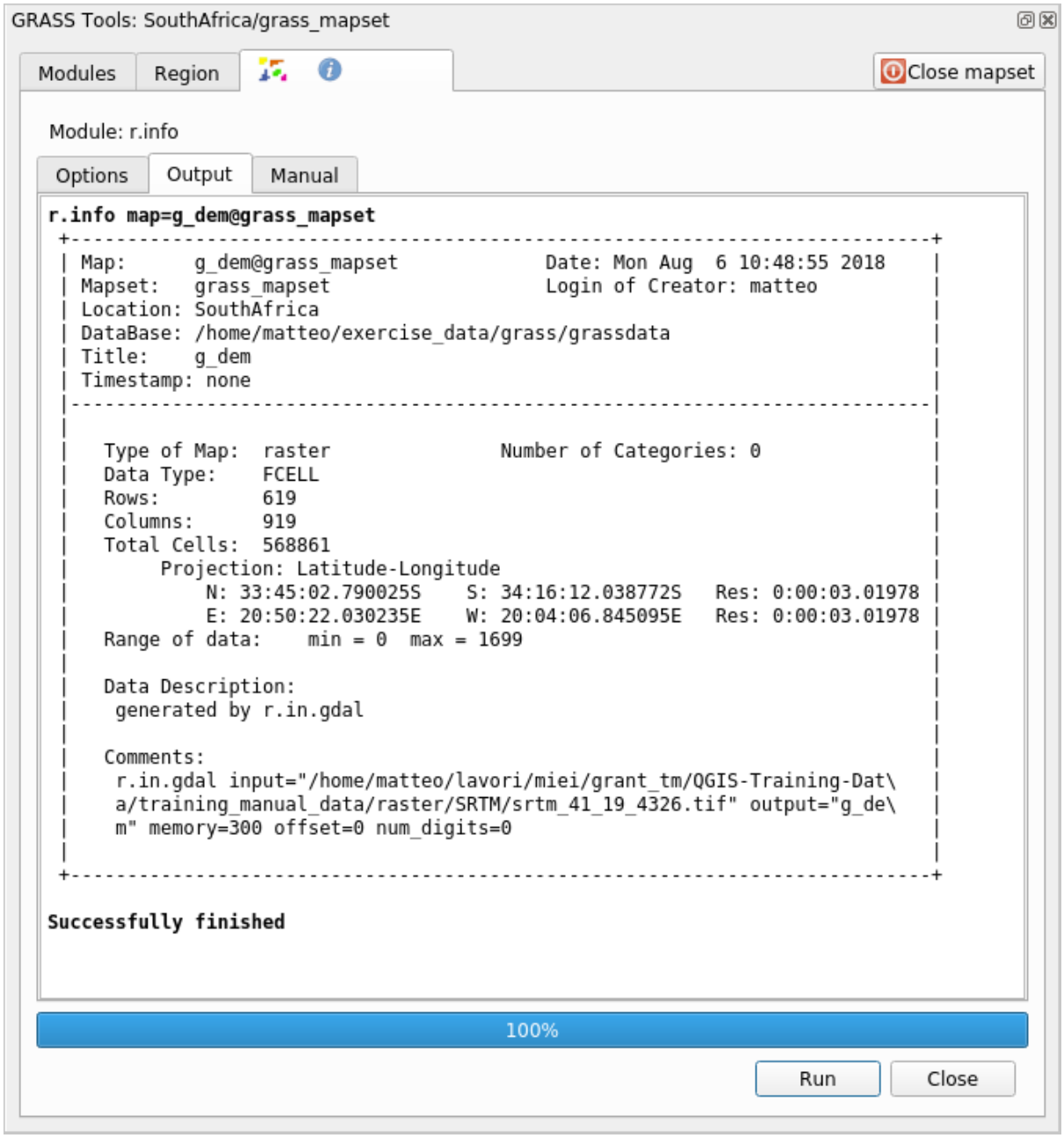
12.2.2 Follow Along: Eine einfache Statistik über den Rasterlayer erstellen

Wir möchten einige einfache Statistiken über den Rasterlayer `g_dem` erstellen.

1. Öffnen Sie das Dialogfenster *GRASS-Werkzeuge*
2. Laden Sie den Rasterlayer `g_dem` aus dem Mapset *grass_mapset*.
3. Suchen Sie nach dem Modul *r.info* durch Eingabe in das *Filter* Feld im *Module* Reiter.
4. Tragen Sie die Werte wie hier dargestellt ein klicken auf *Starten*:



5. Im Ergebnis Reiter sehen wir einige Informationen zum Rasterlayer wie z.B. der Pfad der Datei, die Anzahl an Spalten und Zeilen und andere nützliche Informationen:



12.2.3 Follow Along: Das Reklassifizierungswerkzeug

Die Reklassifizierung eines Rasterlayers kann sehr nützlich sein. Wir haben gerade den Layer `g_aspect` aus dem Layer `g_dem` erstellt. Der Wertebereich geht von 0 (Nord) über 90 (Ost), 180 (Süd), 270 (West) bis zu 360 (wieder Nord). Wir können den Layer `g_aspect` reklassifizieren, so dass er nur noch 4 **Kategorien** entsprechend folgender *Regeln* enthält: (Nord = 1, Ost = 2, Süd = 3 und West = 4).

Das Grass Werkzeug zur Reklassifizierung verwendet eine `txt` Datei mit vorgegebenen Regeln. Das Schreiben der Regeln ist sehr einfach und im GRASS Handbuch ist eine sehr gute Beschreibung dazu enthalten.

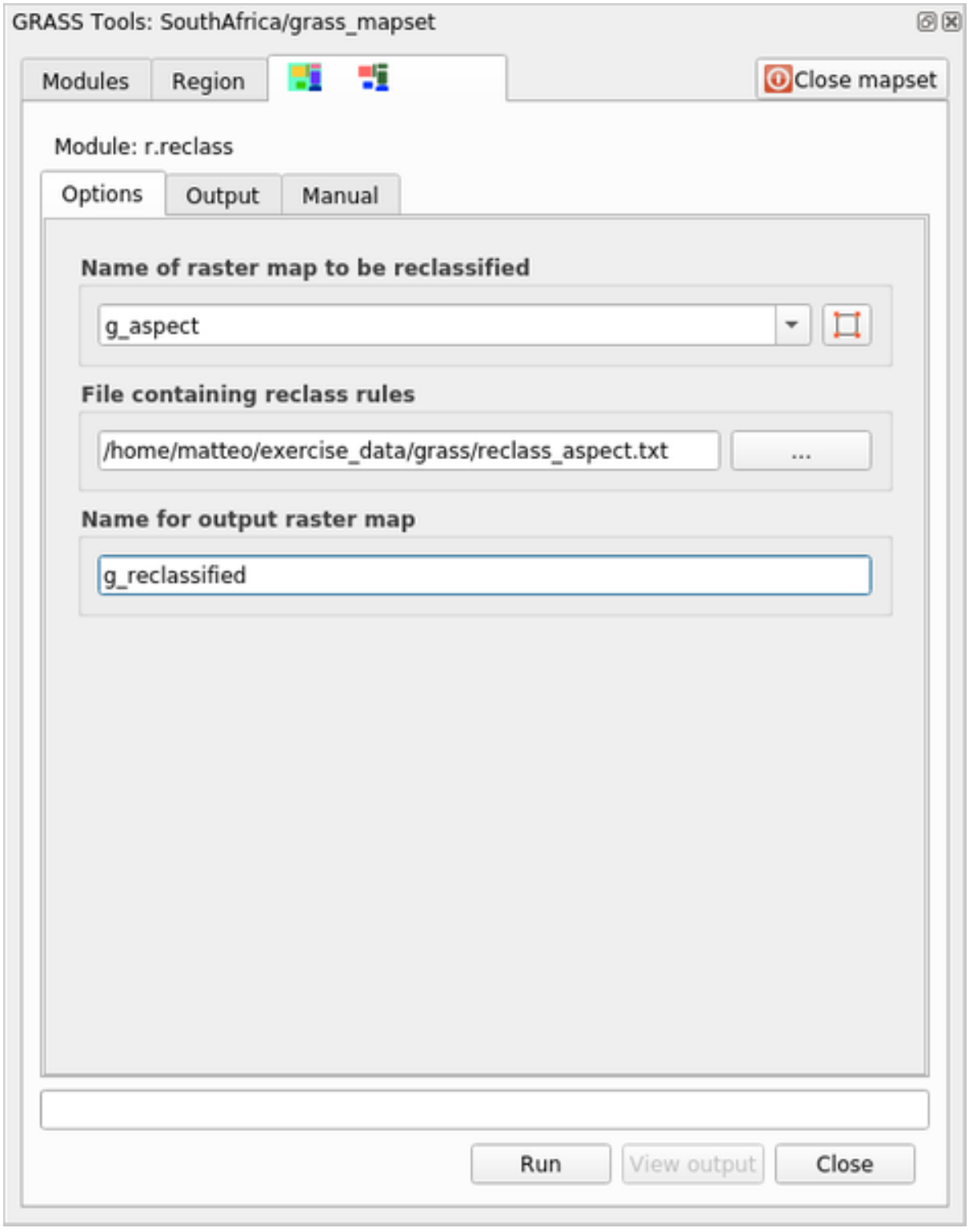
Tipp: Jedes GRASS Werkzeug hat seinen eigenen Handbuch Reiter. Nehmen Sie sich die Zeit, die Beschreibung des Werkzeuges zu lesen, um nicht einige nützliche Parameter zu übersehen.

---

1. Laden Sie den Layer `g_aspect` oder falls Sie ihn noch nicht erstellt haben, gehen Sie zum Abschnitt *Follow*

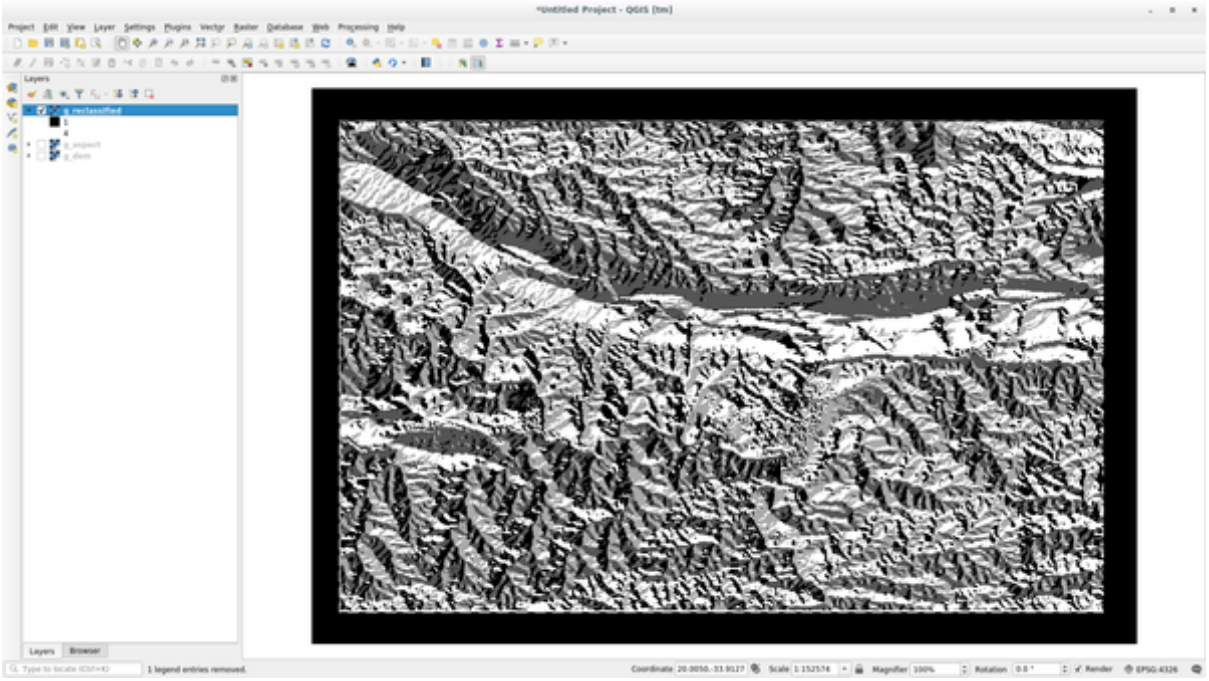
Along: Erstellung einer Karte der Expostition.

2. Suchen Sie nach dem Modul *r.reclass* durch Eingabe in das *Filter* Feld im *Module* Reiter.
3. Öffnen Sie das Werkzeug und tragen Sie die Werte wie im folgenden Bild dargestellt ein. Die Datei mit den Regeln liegt im Ordner `exercise_data/grass/` und hat den Namen `reclass_aspect.txt`.
4. Klicken Sie auf *Starten* und warten bis der Prozess abgeschlossen ist:



5. Klicken Sie auf *Ergebnis visualisieren*, um den reklassifizierten Rasterlayer im Arbeitsbereich anzuzeigen.

Der neue Layer enthält nur 4 Werte (1, 2, 3 und 4) und ist einfacher handzuhaben und zu verarbeiten.



Tipp: Öffnen Sie die Datei `reclass_aspect.txt` mit einem Texteditor, um sich mit den Regeln vertraut zu machen. Sehen Sie sich auch das GRASS Handbuch an. Hier sind viele verschiedene Beispiele tiefergehend erklärt.

12.2.4 Try Yourself Reklassifizieren mit eigenen Regeln

Versuchen Sie den Layer `g_dem` in 3 neue Kategorien zu reklassifizieren:

- von 0 bis 1000: neuer Wert = 1
- von 10000 bis 1400: neuer Wert =2
- von 1400 bis zum maximalen Rasterwert: neuer Wert = 3

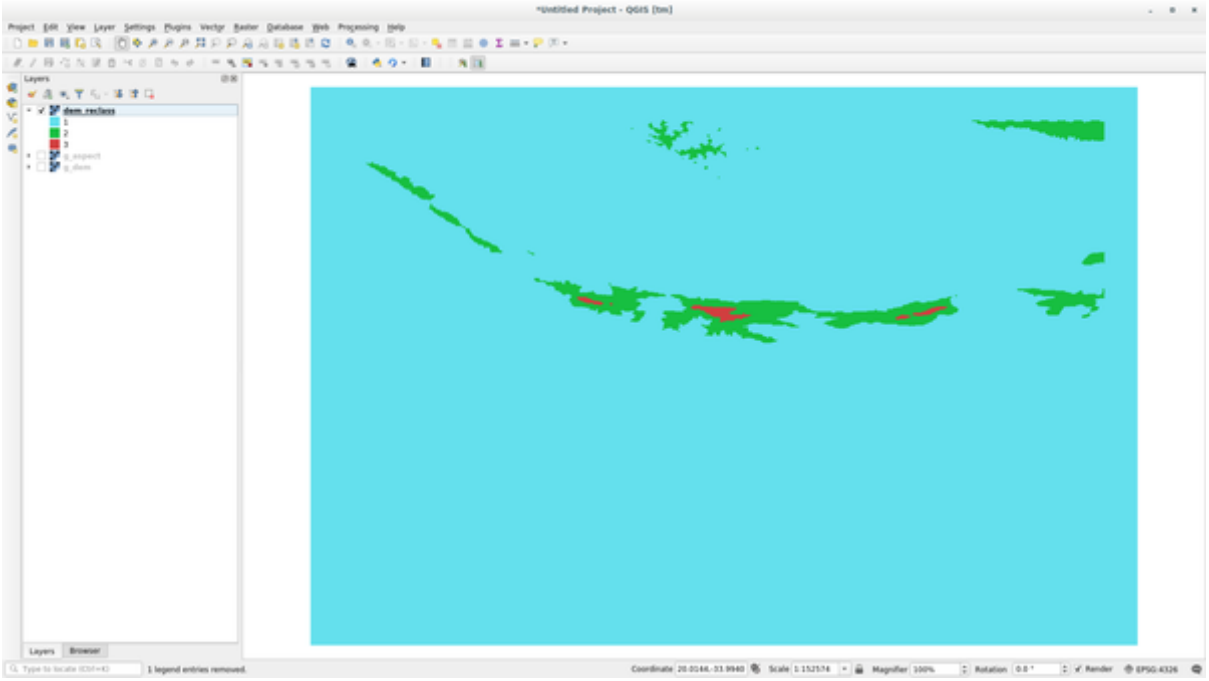
Answer

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: Das Mapcalc Werkzeug

Das Mapcalc Werkzeug ist vergleichbar mit dem Rasterrechner in QGIS. Man führt mathematische Operationen auf einem oder mehreren Rasterlayern aus und erhält im Ergebnis einen neuen Layer mit den berechneten Werten.

Das Ziel der nächsten Übung ist es, Werte größer als 1000 aus dem Rasterlayer `g_dem` zu extrahieren.

1. Suchen Sie nach dem Modul *r.mapcalc* durch Eingabe in das *Filter* Feld des Reiters *Module*.
2. Starten Sie das Werkzeug.

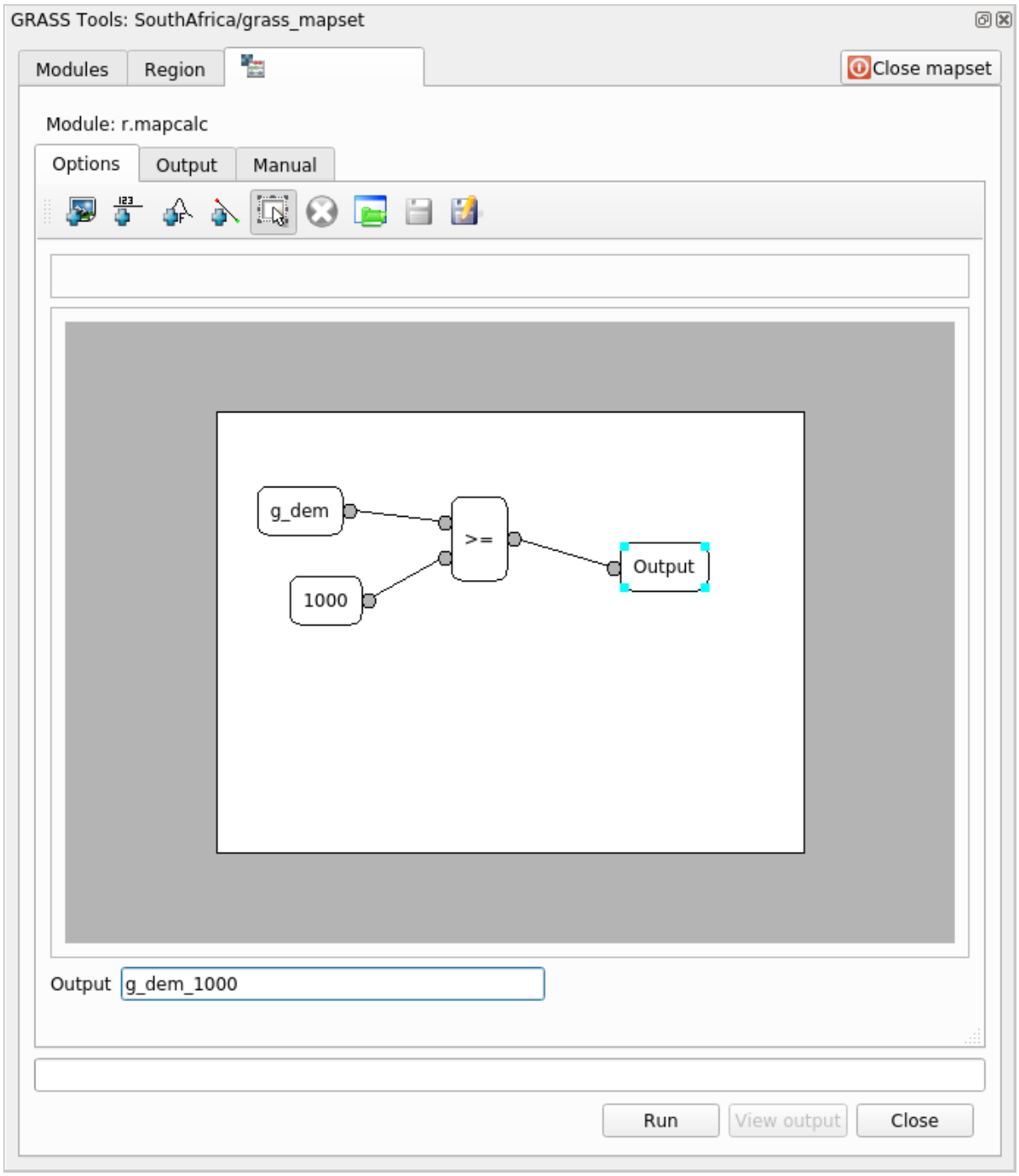
   Der:guilabel:*Mapcalc* Dialog erlaubt es, eine Abfolge von Analysen für ein Raster oder mehrere Raster zu erstellen. Wir verwenden dazu folgende Werkzeuge:

   Dies sind in geordneter Form:

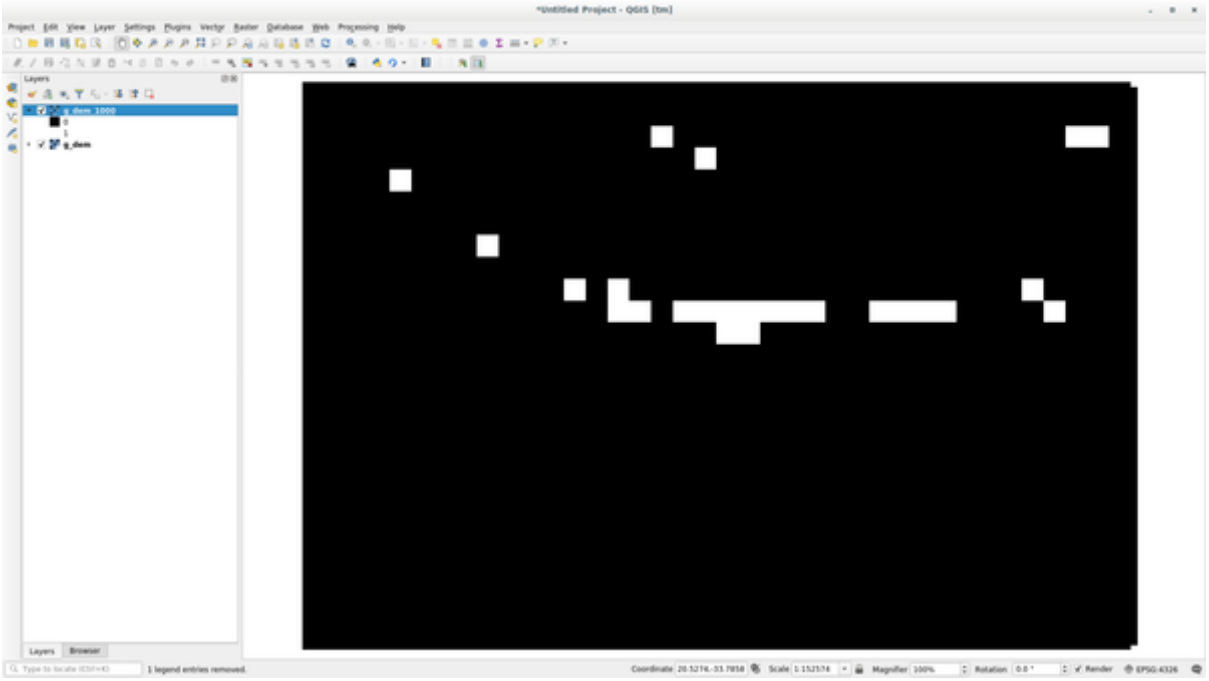
   - *Add map*: Eine Rasterdatei aus dem aktuellen GRASS Mapset hinzufügen.
   - *Add constant value*: Eine Kostante für die Nutzung in Funktionen hinzufügen, in diesem Fall 1000
   - *Add operator or function*: Eine Operation oder Funktion hinzufügen, die mit Eingaben und Ausgaben verbunden ist; wir werden den Operator `greater equals than` verwenden.
   - *Add connection*: Verbindet Elemente. Das Werkzeug wird verwendet, indem man den roten Punkt eines Elements mit dem roten Punkte eines anderen Elements durch klicken und ziehen verbindet. Punkte die korrekt mit einer Linie verbunden sind, ändern ihre Farbe zu grau. Wenn die Linie oder ein Punkt rot ist, so ist die Verbindung nicht korrekt!
   - *Select item*: Auswahl eines Elements und Verschieben eines ausgewählten Elements.
   - *Delete selected item*: Entfernt das ausgewählte Element aus dem aktuellen Mapcalc Blatt, aber nicht aus dem Mapset (wenn es ein vorhandenes Raster ist).

- *Open*: Öffnet eine vorhandene Datei mit der vorgegebenen Operation
- *Save*: Sichert alle Verarbeitungsoperationen in einer Datei.
- *Save as*: Sichert alle Verarbeitungsoperationen in einer neuen lokalen Datei.

3. Erstellen Sie mit Hilfe dieser Werkzeuge den folgenden Algorithmus:



4. Klicken Sie auf *Starten* und dann auf *Ergebnis visualisieren*, um die Ausgabe in der Karte anzuzeigen:



Es werden Flächen dargestellt, in denen das Gelände eine Höhe größer als 1000 Meter erreicht.

Tipp: Sie können die Formel, die Sie erstellt haben auch durch Klick auf die letzte Schaltfläche speichern und in einem anderen QGIS Projekt laden.

12.2.6 In Conclusion

In dieser Lektion haben wir nur einige der vielen Werkzeuge behandelt, die GRASS bietet. Um die Möglichkeiten von GRASS selbst zu erkunden, öffnen Sie den Dialog *GRASS Werkzeuge* und scrollen Sie in der *Modulliste* nach unten. Oder für eine strukturiertere Herangehensweise schauen Sie unter dem Reiter *Modulbaum* nach, in dem die Werkzeuge nach Typ geordnet sind.

KAPITEL 13

Module: Bewertung

Verwenden Sie Ihre eigenen Daten für diesen Abschnitt. Sie benötigen:

- einen Punktvektor-Datensatz von Sonderzielen, mit Punktnamen und mehreren Kategorien
- ein Linienvektordatensatz von Straßen
- ein Polygon-Vektordatensatz der Landnutzung (unter Verwendung von Grundstücksgrenzen)
- ein Bild im visuellen Spektrum (z. B. ein Luftbild)
- ein DEM (herunterladbar von the CGIAR-CSI falls Sie kein eigenes haben)

13.1 Erstellen Sie eine Basiskarte

Vor jeder Datenanalyse benötigen Sie eine Basiskarte, die das Ergebnis Ihrer Analyse in einen Kontext setzt.

13.1.1 Fügen Sie den Punktlayer hinzu

- Fügen Sie den Punktlayer hinzu. Führen Sie je nach Niveau, auf dem Sie den Kurs absolvieren, nur das aus, was im entsprechenden Abschnitt unten aufgeführt ist:

- Beschriften Sie die Punkte nach einem eindeutigen Attribut, z. B. Ortsnamen. Verwenden Sie eine kleine Schriftart und halten Sie die Beschriftungen unauffällig. Die Informationen sollten verfügbar sein, aber kein Hauptmerkmal der Karte sein.
- Klassifizieren Sie die Punkte selbst anhand einer Kategorie in verschiedene Farben. Kategorien könnten beispielsweise „Touristenziel“, „Polizeiwache“ und „Stadtzentrum“ umfassen.

- Gehen Sie genauso vor wie bei Sektion.
- Klassifizieren Sie die Punktgröße nach Wichtigkeit: Je signifikanter ein Merkmal ist, desto größer ist sein Punkt. Überschreiten Sie jedoch nicht die Punktgröße von `2.00`.
- Weisen Sie Features, die sich nicht an einem einzelnen Punkt befinden (z. B. Namen von Provinzen/Regionen oder Städtenamen in großem Maßstab), überhaupt keinen Punkt zu.

- Verwenden Sie überhaupt keine Punktsymbole, um den Layer zu symbolisieren. Verwenden Sie stattdessen Beschriftungen, die über den Punkten zentriert sind; die Punktsymbole selbst sollten keine Größe haben.
- Verwenden Sie *Data defined settings*, um die Beschriftungenin aussagekräftige Kategorien einzuteilen.
- Fügen Sie bei Bedarf geeignete Spalten zu den Attributdaten hinzu. Erstellen Sie dabei keine fiktiven Daten, sondern verwenden Sie den *Field Calculator*, um die neuen Spalten basierend auf den entsprechenden vorhandenen Werten im Datensatz zu füllen.

13.1.2 Fügen Sie einen Linien Layer hinzu

- Fügen Sie den Straßen-Layer hinzu und ändern Sie dann seine Symbologie. Straßen nicht beschriften.

- Ändern Sie die Straßensymbologie in eine helle Farbe mit einer breiten Linie. Machen Sie es auch etwas transparent.

- Erstellen Sie ein Symbol mit mehreren Symbol-Layern. Das resultierende Symbol sollte wie eine echte Straße aussehen. Sie können dafür ein einfaches Symbol verwenden; zum Beispiel eine schwarze Linie mit einer dünnen weißen durchgezogenen Linie, die in der Mitte verläuft. Es kann auch aufwändiger sein, aber die resultierende Karte sollte nicht zu unruhig aussehen.
- Wenn Ihr Dataset in dem Maßstab, in dem Sie die Karte anzeigen möchten, eine hohe Straßendichte aufweist, sollten Sie zwei Straßen-Layer haben: das aufwändige straßenähnliche Symbol und ein einfacheres Symbol in kleineren Maßstäben. (Verwenden Sie die maßstabsbasierte Sichtbarkeiten, damit die Darstellung bei geeigneten Maßstäben wechselt.)
- Alle Symbole sollten mehrere Symbolebenen haben. Verwenden Sie Symbole, damit sie richtig angezeigt werden.

- Gehen Sie genauso vor wie im Abschnitt oben.
- Außerdem sollten Straßen klassifiziert werden. Bei der Verwendung realistischer straßenähnlicher Symbole sollte jeder Straßentyp ein geeignetes Symbol haben; Beispielsweise sollte eine Autobahn so aussehen, als hätte sie zwei Fahrspuren in beide Richtungen.

13.1.3 Fügen Sie den Polygonlayer hinzu

- Fügen Sie den Landnutzungs-Layer hinzu und ändern Sie seine Symbologie.

- Klassifizieren Sie die Schicht nach Landnutzung. Verwenden Sie Volltonfarben.

- Klassifizieren Sie die Schicht nach Landnutzung. Integrieren Sie gegebenenfalls Symbol-Layer, verschiedene Symboltypen usw. Lassen Sie die Ergebnisse jedoch gedämpft und einheitlich aussehen. Denken Sie daran, dass dies Teil eines Hintergrunds sein wird!

- Verwenden Sie die regelbasierte Klassifizierung, um die Landnutzung in allgemeine Kategorien wie „städtisch“, „ländlich“, „Naturschutzgebiet“ usw. einzuteilen.

13.1.4 Erstellen Sie den Rasterhintergrund

- Erstellen Sie eine Schummerung aus dem DEM und verwenden Sie sie als Überlagerung für eine klassifizierte Version des DEM selbst. Sie könnten auch das Plugin *Relief* verwenden (wie in der Lektion über Plugins gezeigt).

13.1.5 Stellen Sie die Basiskarte fertig

- Erstellen Sie mit den oben genannten Ressourcen eine Basiskarte mit einigen oder allen Layern. Diese Karte sollte alle grundlegenden Informationen enthalten, die zur Orientierung des Benutzers erforderlich sind, sowie visuell einheitlich / „einfach“ sein.

13.2 Analysieren Sie die Daten

- Sie suchen eine Immobilie, die bestimmte Kriterien erfüllt.
- Sie können eigene Kriterien festlegen, die Sie dokumentieren möchten.
- Für diese Kriterien gibt es einige Richtlinien:
  - das Zielgrundstück sollte (einer) bestimmten Art(en) der Landnutzung angehören
  - es sollte innerhalb einer bestimmten Entfernung von Straßen liegen oder von einer Straße gekreuzt werden
  - Es sollte sich innerhalb einer bestimmten Entfernung von einer bestimmten Kategorie von Punkten befinden, z. B. einem Krankenhaus

13.2.1 /

- Beziehen Sie Rasteranalysen in Ihre Ergebnisse ein. Berücksichtigen Sie mindestens eine abgeleitete Eigenschaft des Rasters, z. B. Ausrichtung oder Neigung.

13.3 Endgültige Karte

- Verwenden Sie das *Print Layout*, um eine endgültige Karte zu erstellen, die Ihre Analyseergebnisse enthält.
- Nehmen Sie diese Karte zusammen mit Ihren dokumentierten Kriterien in ein Dokument auf. Wenn die Karte aufgrund der hinzugefügten Ebene(n) visuell zu unübersichtlich geworden ist, deaktivieren Sie die Ebenen, die Ihrer Meinung nach am wenigsten wichtig sind.
- Ihre Karte muss einen Titel und eine Legende enthalten.

KAPITEL 14

Module: Anwendung in der Forstwirtschaft

In den Modulen 1 bis 13 haben Sie bereits einiges über QGIS und die Arbeit damit gelernt. Sind Sie daran interessiert, ein paar grundlegende Anwendungsbereiche von GIS in der Forstwirtschaft kennenzulernen? Im folgenden Modul werden Sie das bisher Gelernte anwenden und Ihnen werden einige nützliche neue Werkzeuge vorgestellt.

Die Entwicklung dieses Moduls wurde von der Europäischen Union finanziert.

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

Bemerkung: 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.

Warnung: 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

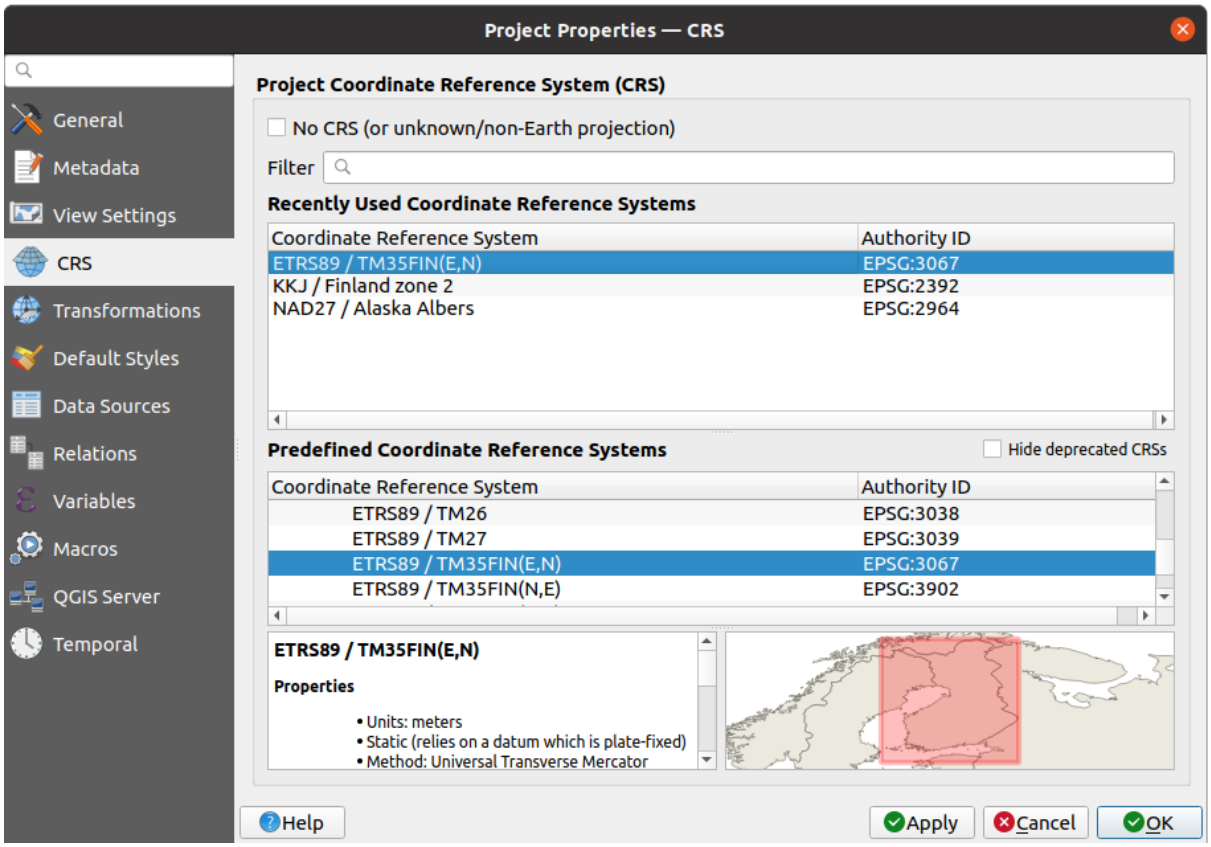
The first task you will have to do is to scan your map. If your map is too big, then you can scan it in different parts but keep in mind that you will have to repeat preprocessing and georeferencing tasks for each part. So if possible, scan the map in as few parts as possible.

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

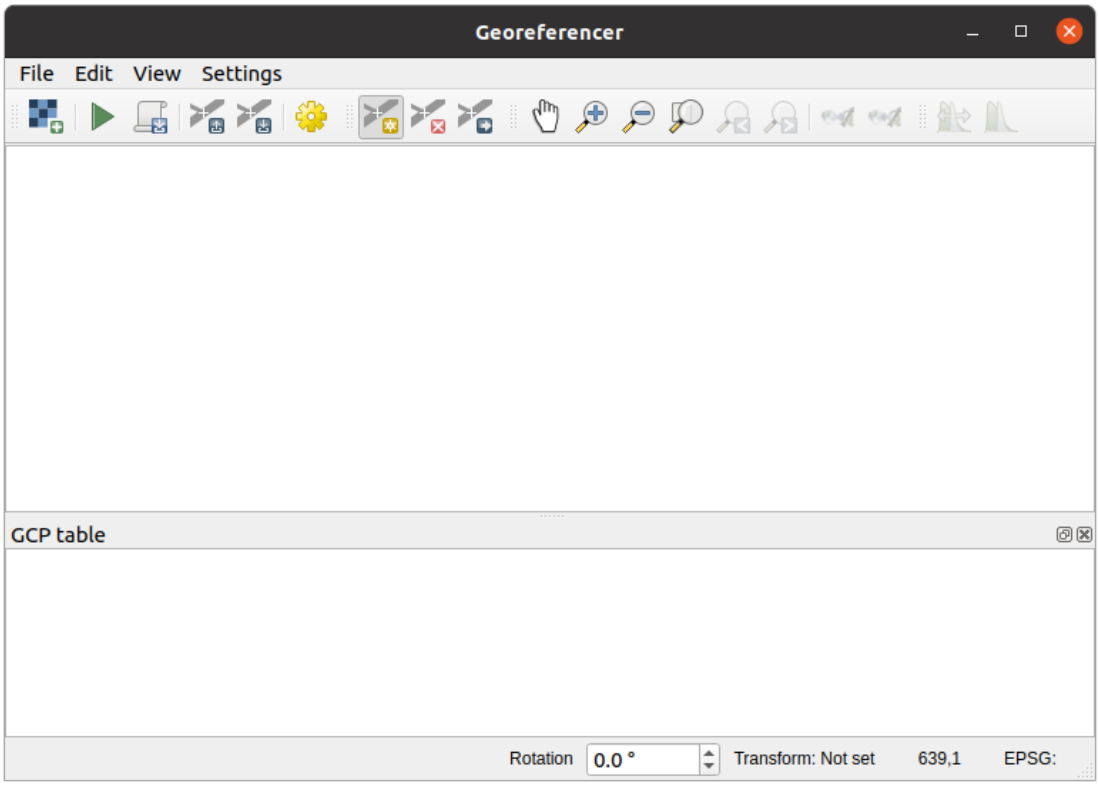
1. Open QGIS and set the project's CRS to `EPSG:3067 - ETRS89 / TM35FIN(E,N)` in *Project* ► *Properties* ► *CRS*, which is the currently used CRS in Finland.



2. Save the QGIS project as `map_digitizing.qgs`.

You will use the georeferencing tool from QGIS named *Georeferencer*. To georeference the map:

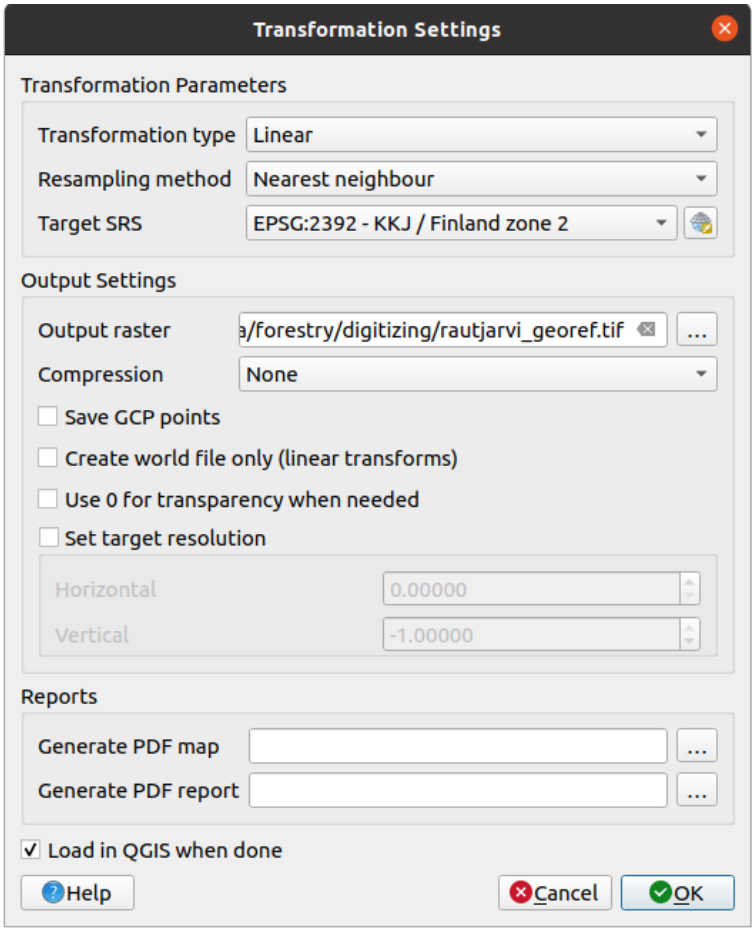
1. Open the georeference tool, *Layer* ► *Georeferencer*....



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.

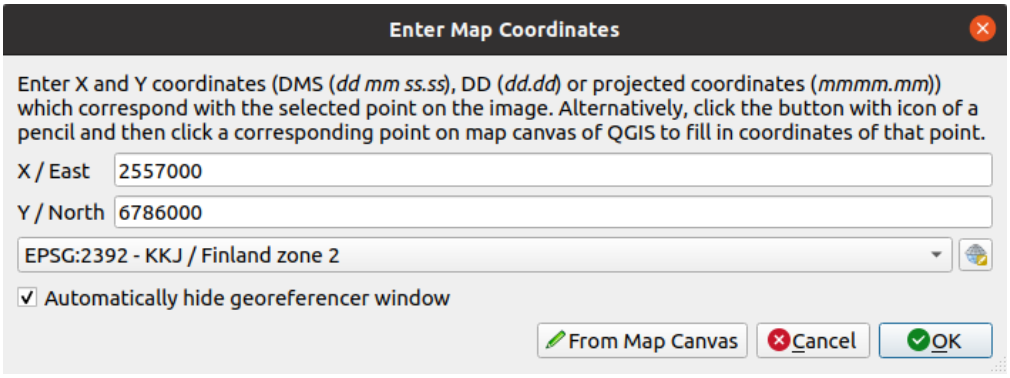


7. Click *OK*.

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`)



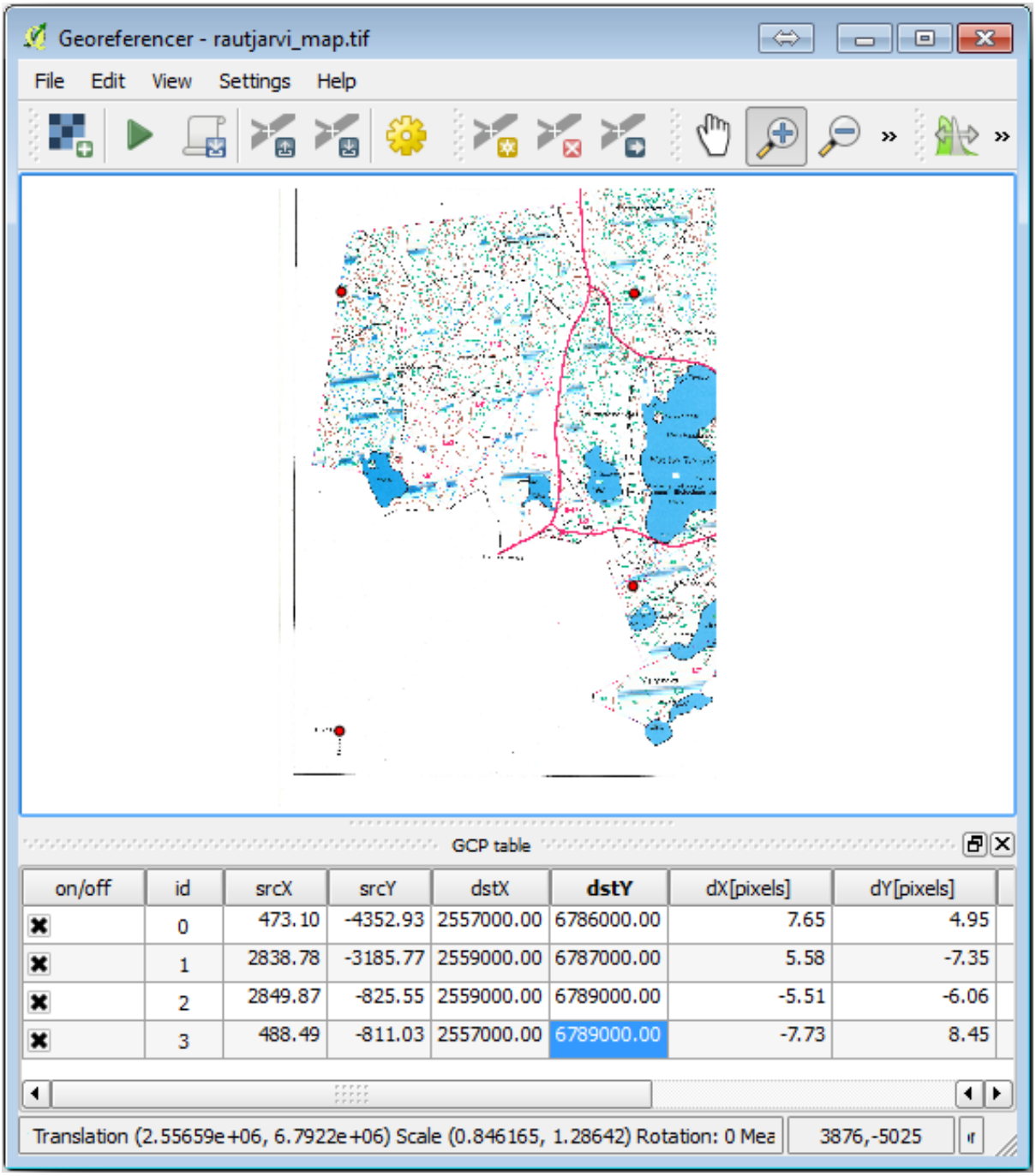
4. Click *OK*.

   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:

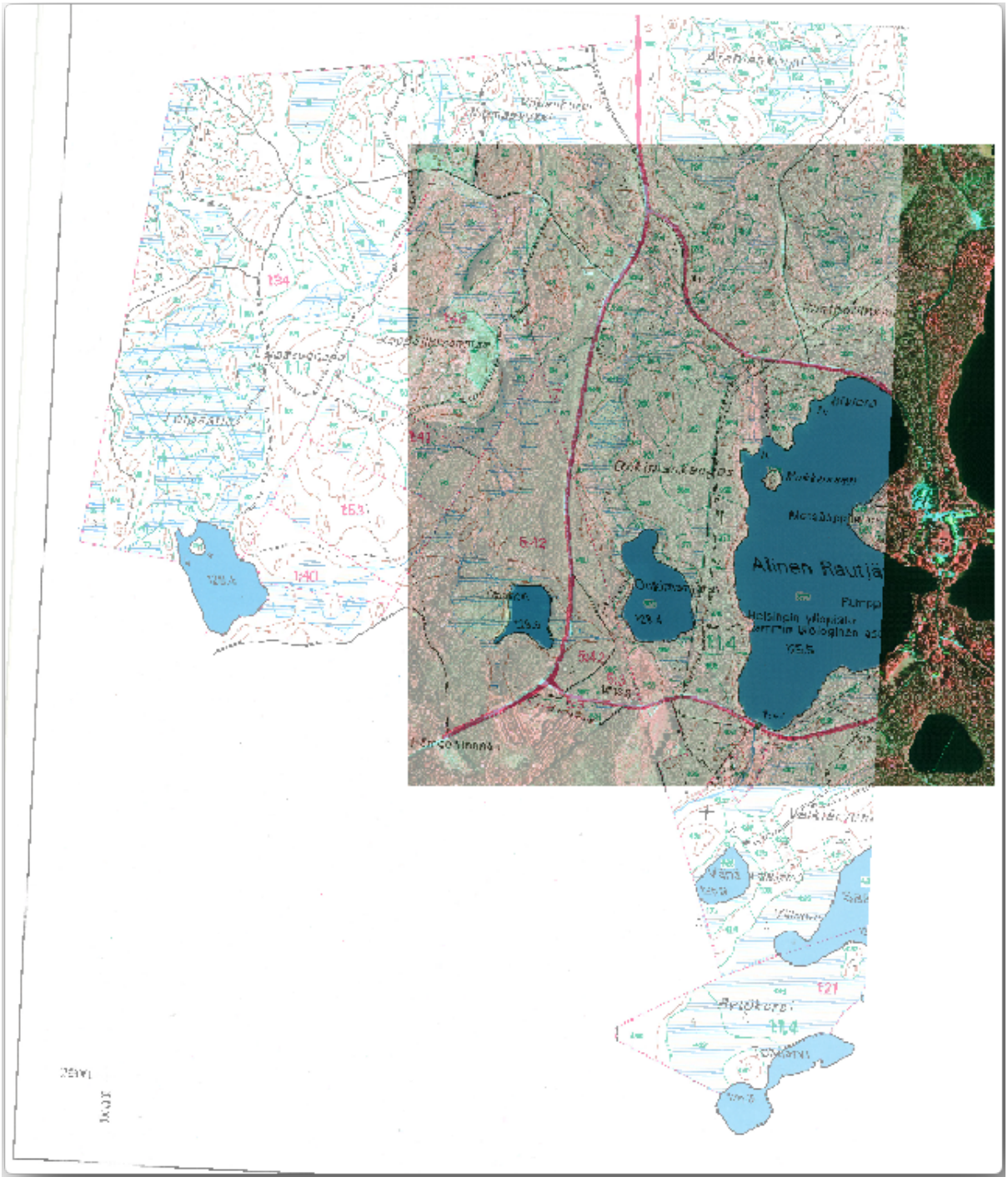
1. Go to *File* ► *Start georeferencing*.

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 *Lesson: Creating a New Vector Dataset*, 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

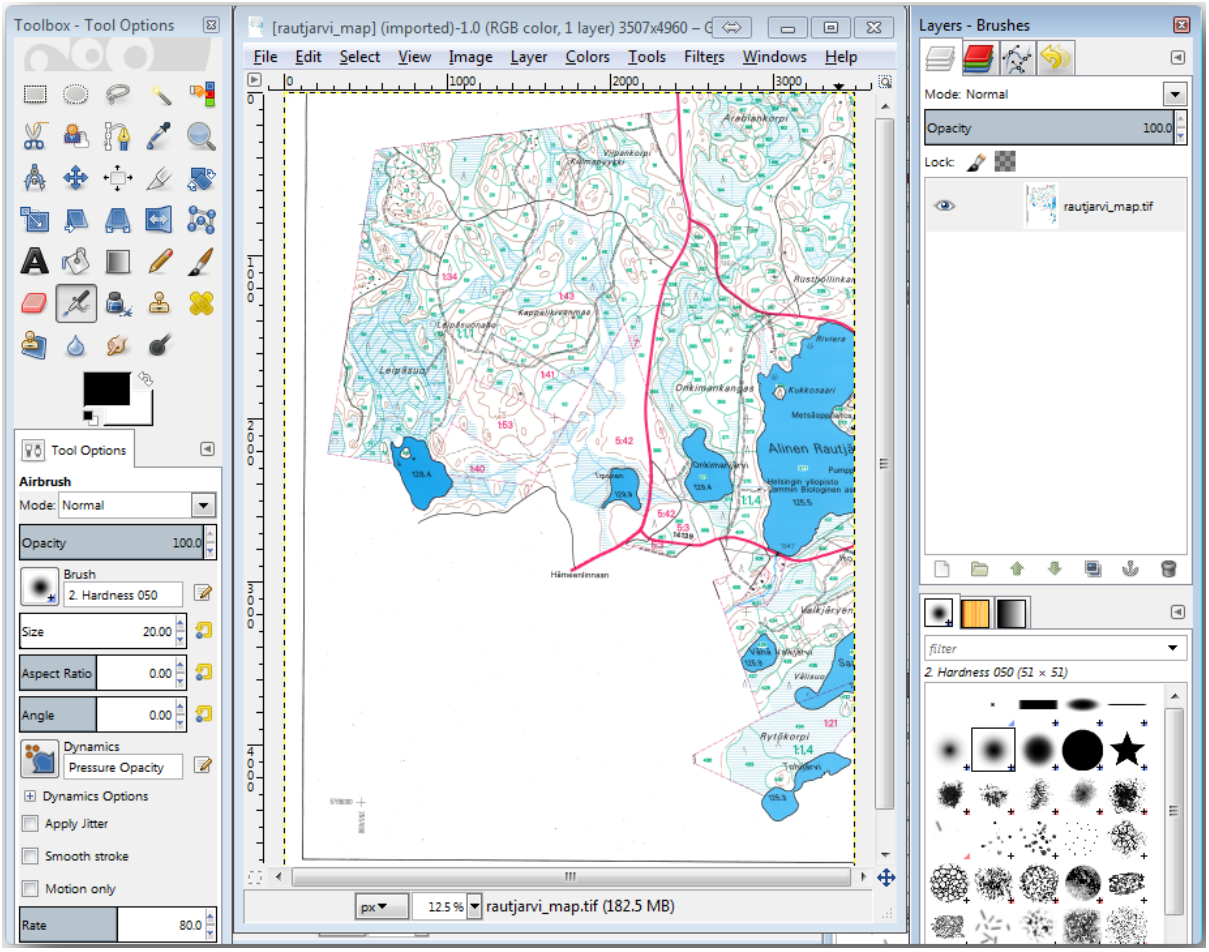
Open your `map_digitizing.qgs` project in QGIS, that you saved from the previous lesson.

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.

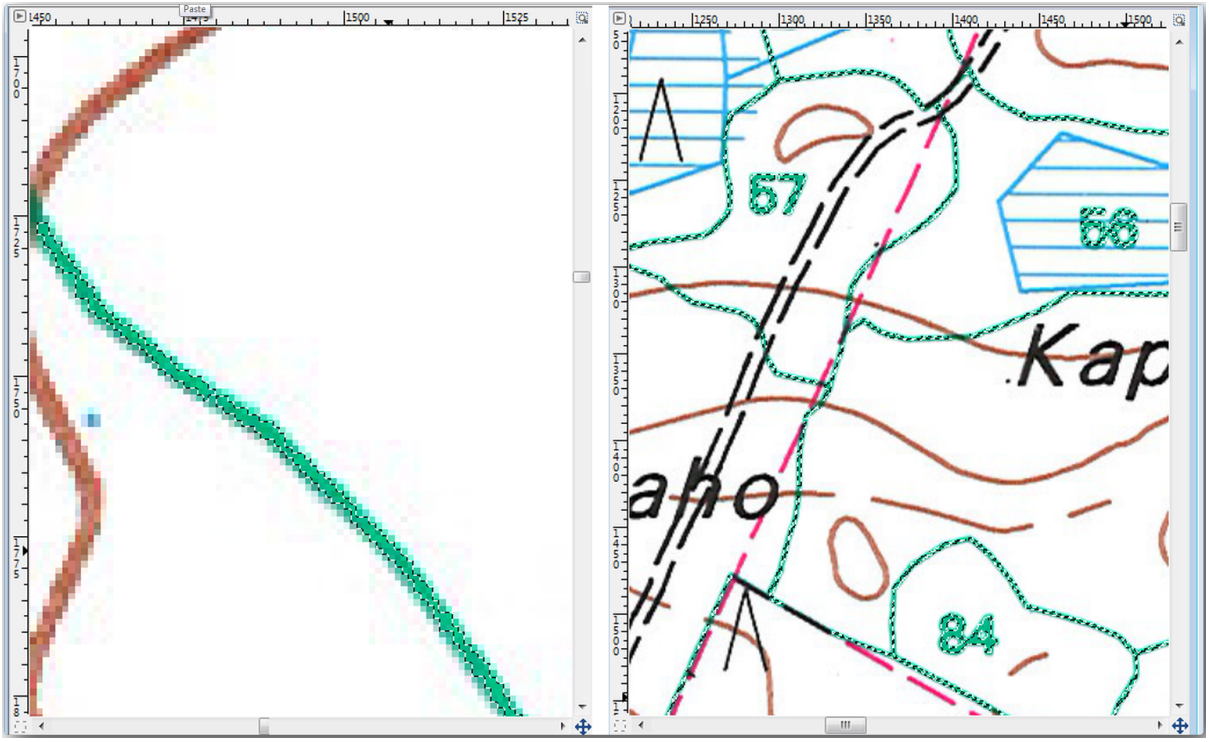
If what you are using to digitize is a good map, as it is in our case, it is likely that the information is clearly displayed as lines with different colors for each type of element. Those colors can be relatively easy extracted as individual images using an image processing software like GIMP. Such 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:

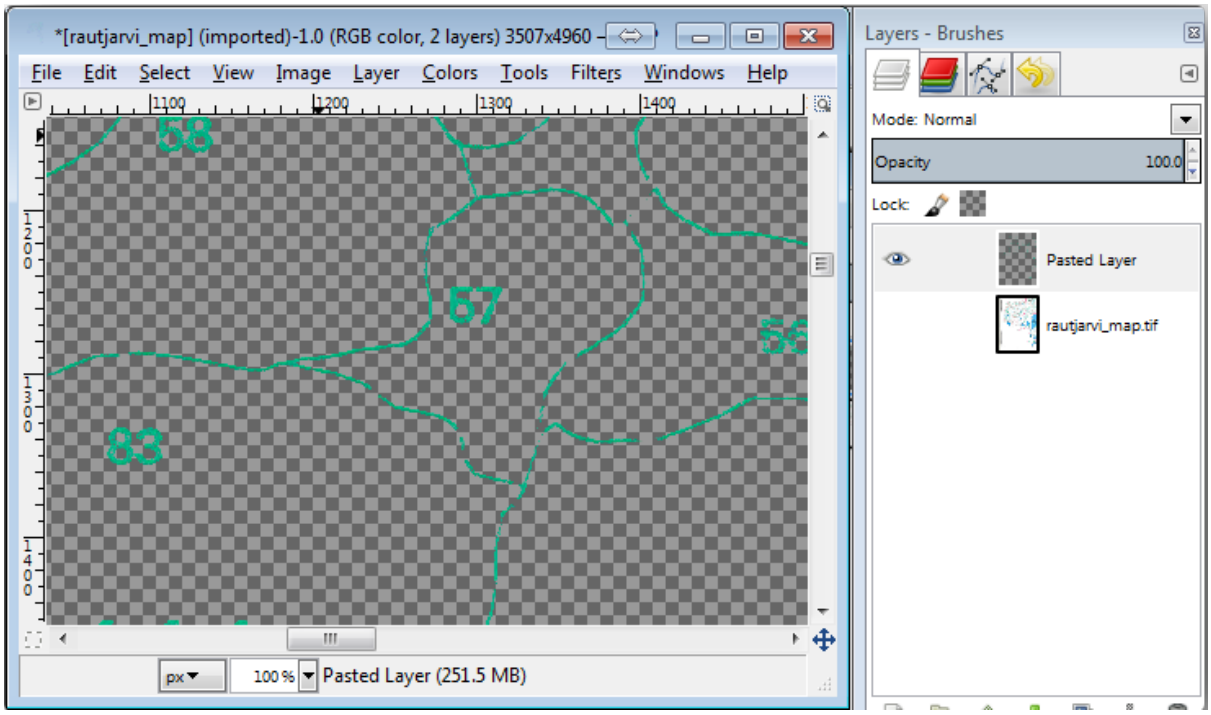
1. Open GIMP (if you don't have it installed yet, download it from the internet or ask your teacher).
2. Open the original map image, *File* ► *Open*, `rautjarvi_map.tif` in the `exercise_data/forestry` folder. Note that the forest stands are represented as green lines (with the number of the stand also in green inside each polygon).



3. Now you can select the pixels in the image that are making up the forest stands' borders (the greenish pixels):
    1. Open the tool *Select* ► *By color*.
    2. 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.
    3. Click and drag the mouse cursor in the middle of the line so that the tool will collect several pixel color values.
    4. 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.
    5. Zoom out to see how the greenish pixels have been selected throughout the image.
    6. If you are not happy with the result, repeat the click and drag operation.
    7. Your pixel selection should look something like the right image below.



4. 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:
    1. Copy (`Ctr+C`) the selected pixels.
    2. 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)*.
    3. Right click that temporary layer and select *To New Layer*.
    4. Click the „eye“ icon next to the original image layer to switch it off, so that only the *Pasted Layer* is visible:



5. 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 anymore 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:

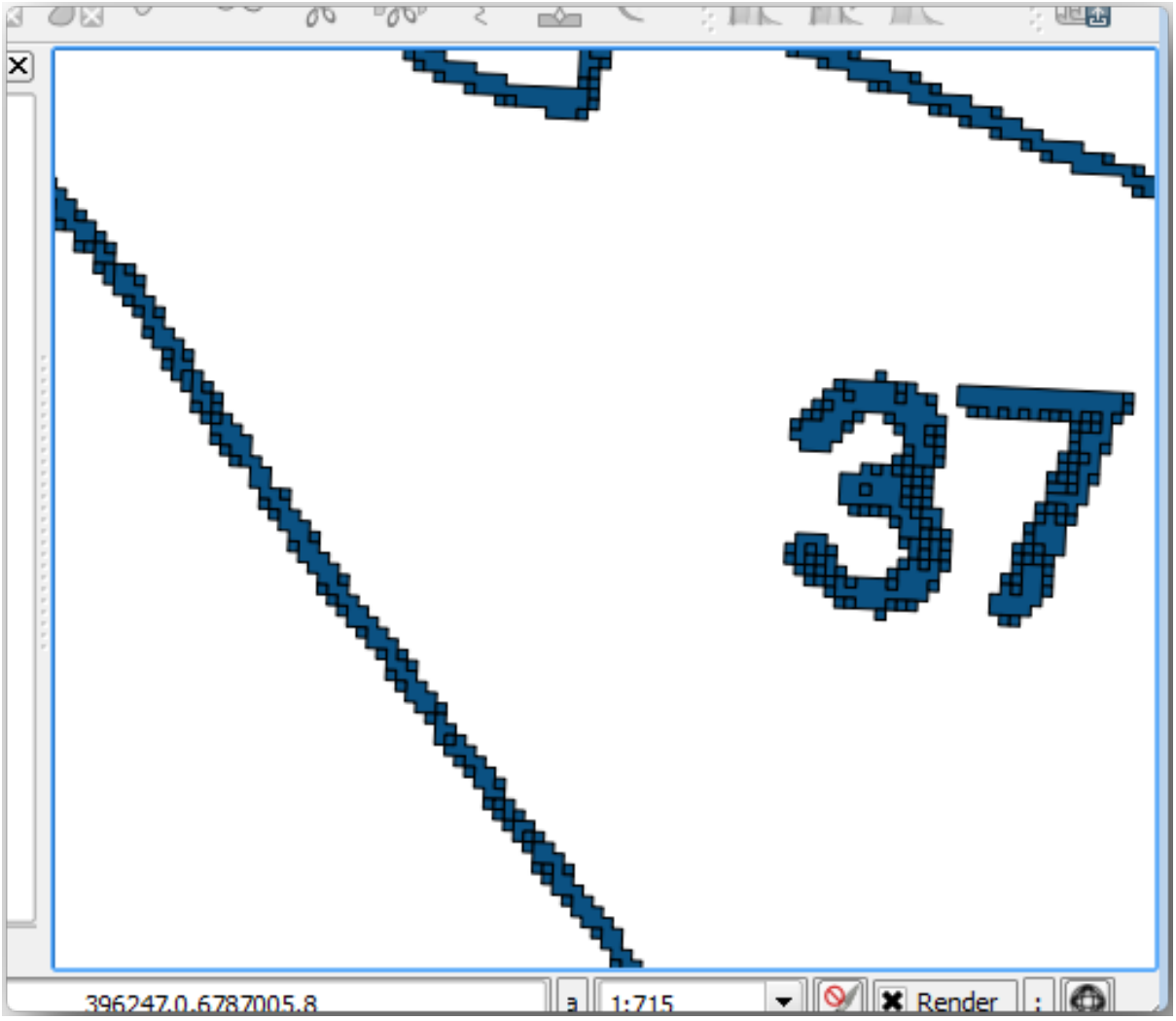
- This image is also, of course, in `KKJ / Finland zone 2` CRS.
- You should use the ground control points you saved, *File* ► *Load GCP points*.
- Remember to review the *Transformation settings*.
- Name the output raster as `rautjarvi_green_georef.tif` in the `digitizing` folder.

Check that the new raster is fitting nicely with the original map.

14.3.3 Follow Along: Creating Supporting Points for Digitizing

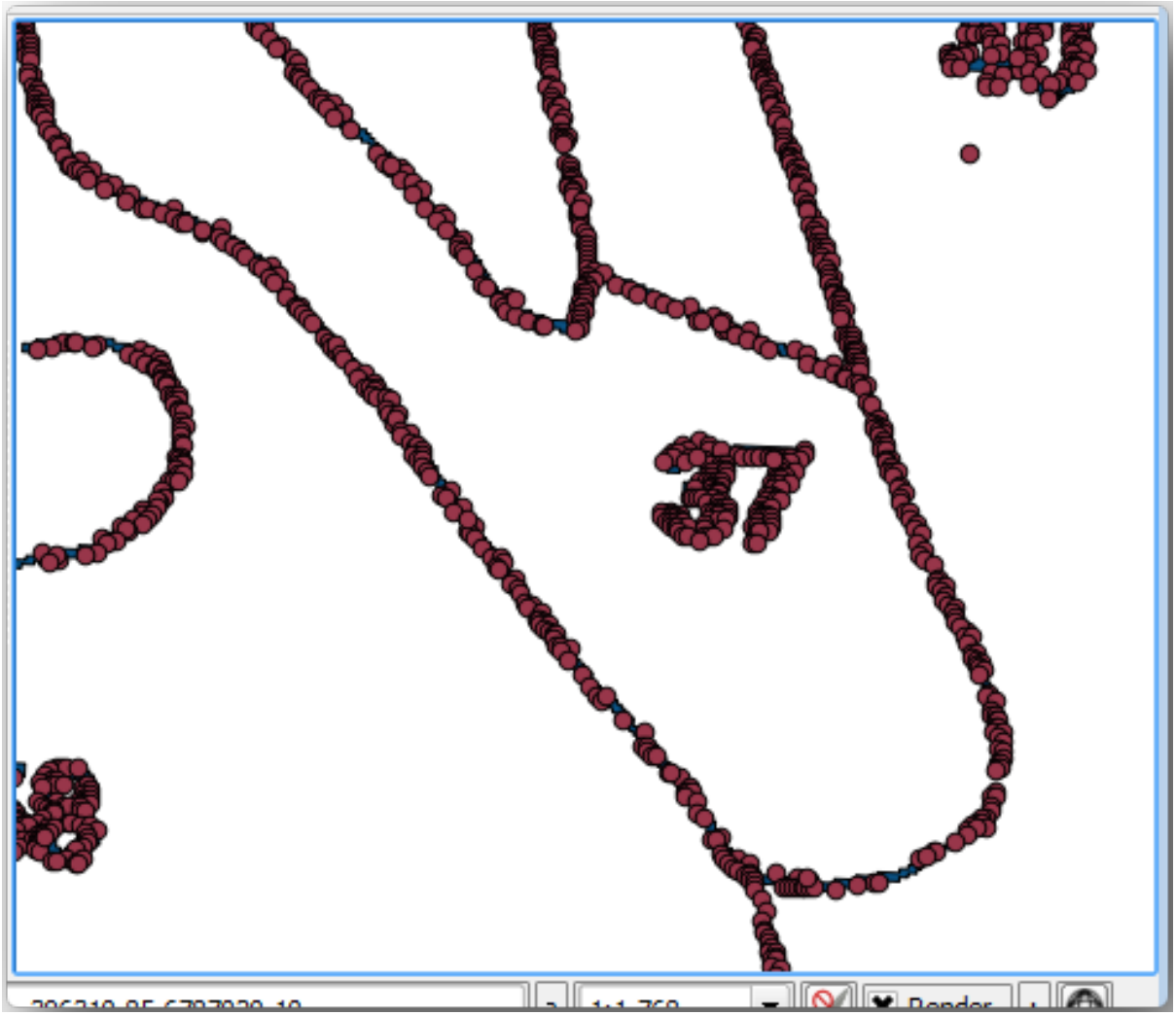
Having in mind the digitizing tools in QGIS, you might already be thinking that it would be helpful to snap to those green pixels while digitizing. That is precisely what you are going to do next create points from those pixels to use them later to help you follow the forest stands' borders when digitizing, by using the snapping tools available in QGIS.

1. 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 *Lesson: Raster to Vector Conversion*.
2. Save as `rautjarvi_green_polygon.shp` inside the `digitizing` folder.
3. Zoom in and see what the polygons look like. You will get something like this:



4. Next option to get points out of those polygons is to get their centroids:
   1. Open *Vector* ► *Geometry Tools* ► *Centroids…*.
   2. Set *Input Layer* to *rautjarvi_green_polygon* (the polygon layer you have just created)
   3. Set *Centroids* output to `green_centroids.shp` file within the folder `digitizing`
   4. Check *Open output file after running algorithm*

5. Press *Run*. This will calculate the centroids for the polygons as a new layer and add it to the project.

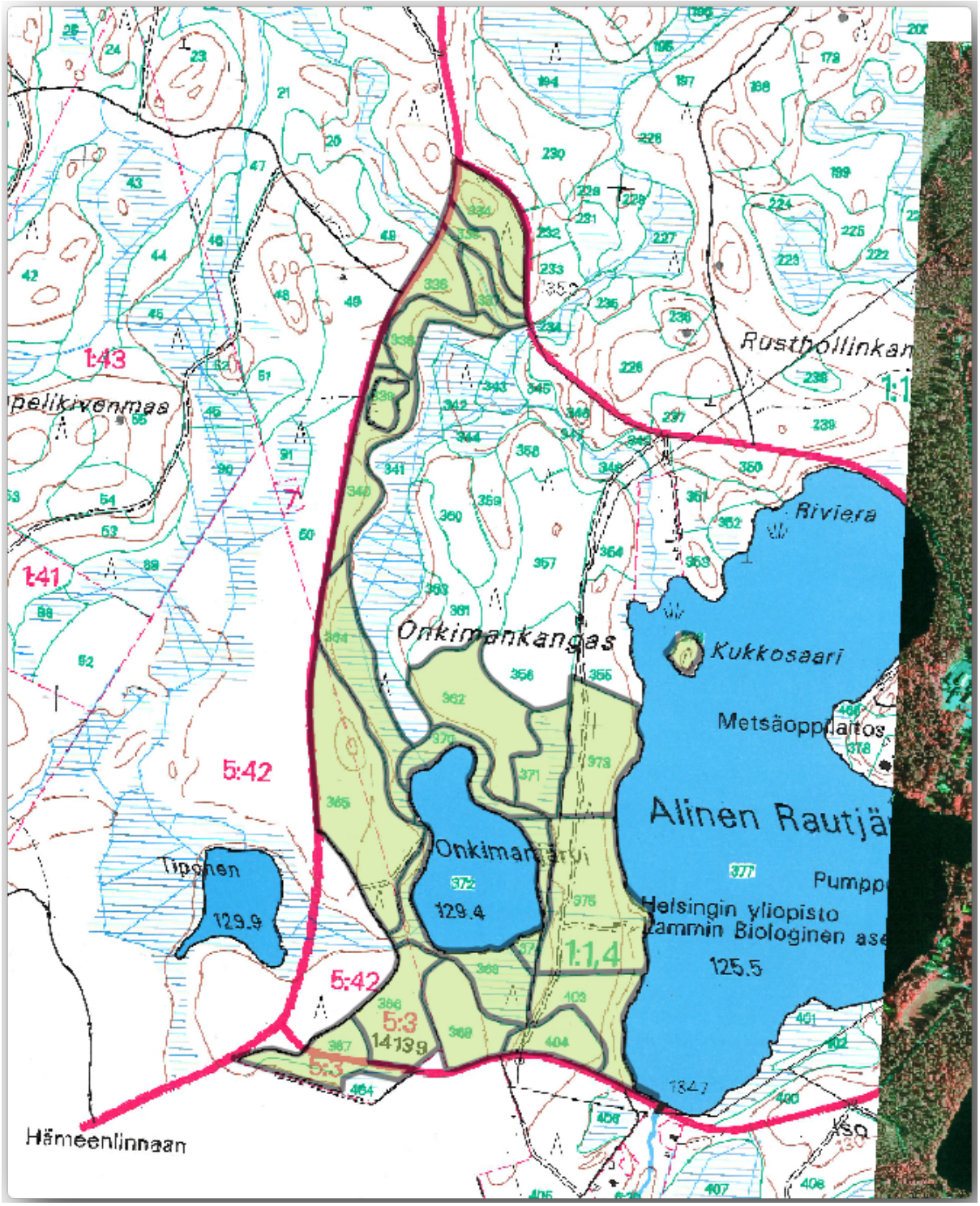


5. Now you can remove the `rautjarvi_green_polygon` layer from the TOC.
6. Change the symbology of the centroids layer as follows:
    1. Open the *Layer Properties* for `green_centroids`.
    2. Go to the *Symbology* tab.
    3. Set *Size* to `1.00` and choose *Map Units*

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

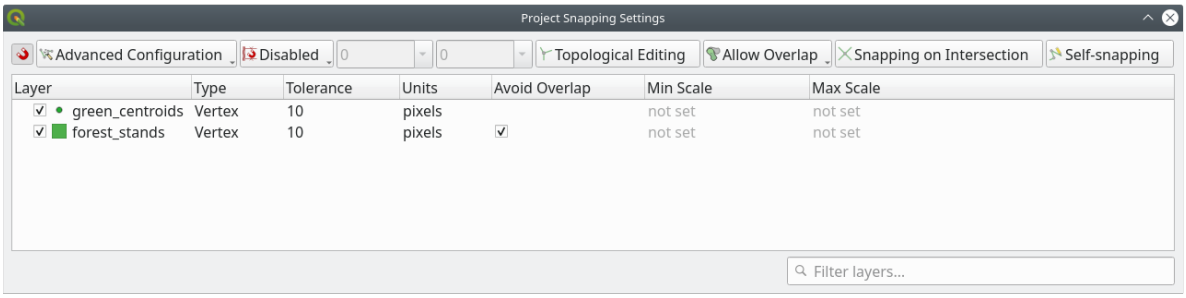
Now you are ready to start with the actual digitizing work. You would start by creating a vector file of *polygon type*, but for this exercise, there is a shapefile with part of the area of interest already digitized. You will just finish digitizing the half of the forest stands that are left between the main roads (wide pink lines) and the lake:



1. Go to the `digitizing` folder using your file manager browser.
2. Drag and drop the `forest_stands.shp` vector file to your map.
3. Change the new layer's symbology so that it will be easier to see the polygons that have already been digitized.
    1. Set *Fill color* to green - and change the *Opacity* to `50%`.
    2. Select *Simple Fill* and set *Stroke width* to `1.00 mm`.

Now, if you remember past modules, we have to set up and activate the snapping options:

1. Go to *Project* ► *Snapping options…*
2. Press Enable Snapping and select *Advanced Configuration*
3. Check the *green_centroids* and *forest_stands* layers
4. Set *Type* for each layer to *Vertex*
5. Set *Tolerance* for each layer to `10`
6. Set *Units* for each layer to *pixels*
7. Check *Avoid Overlap* for the *forest_stands* layer
8. Press *Topological editing*
9. Choose *Follow Advanced Configuration*
10. Close the pop-up



    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 square will appear on the point that will be snapped to.

11. Finally, turn off the visibility of all the layers except *forest_stands* and *rautjarvi_georef*. Make sure that the map image has not transparency any more.

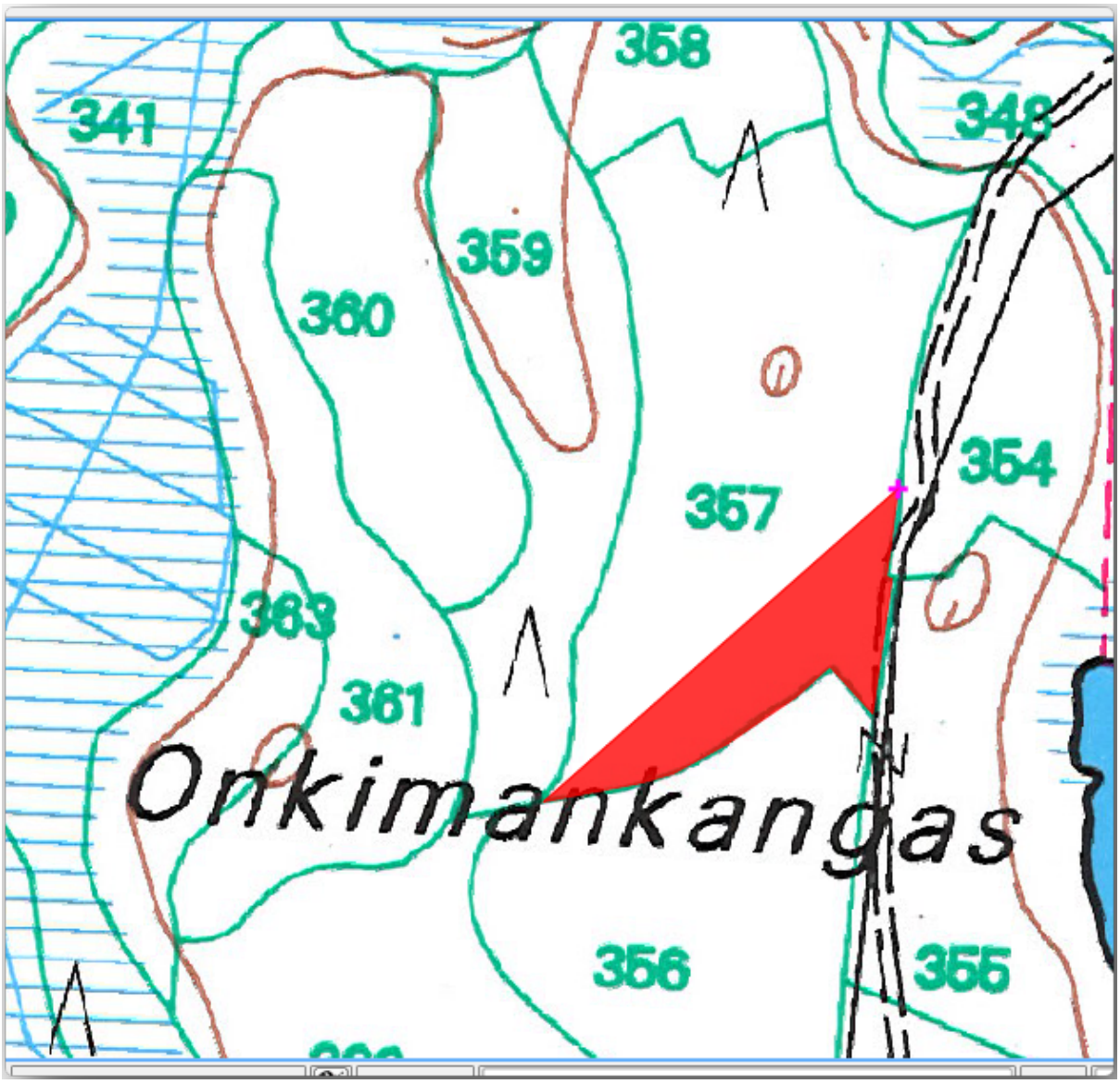
A few important things to note before you start digitizing:

- 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:

1. Locate the forest stand number `357` in the map window.

2. Select the `forest_stands` layer.
3. Click the Toggle Editing button to enable editing
4. Select Add Polygon Feature tool.
5. Start digitizing the stand `357` by connecting some of the dots. Note the pink crosses indicating the snapping.

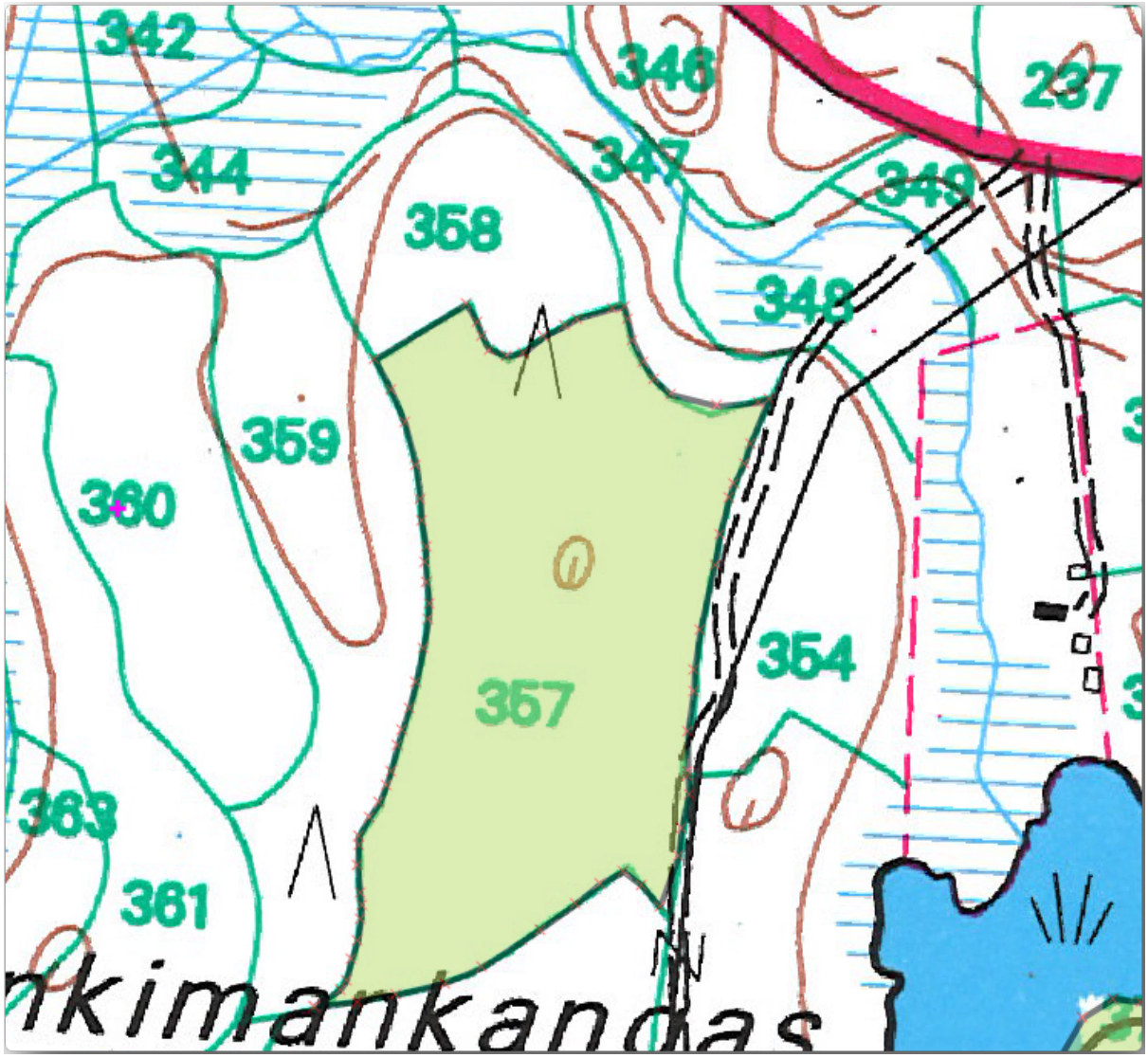


When you are done:

1. Right click to end digitizing that polygon.
2. Enter the forest stand *ID* within the form (in this case `357`).
3. Click *OK*.

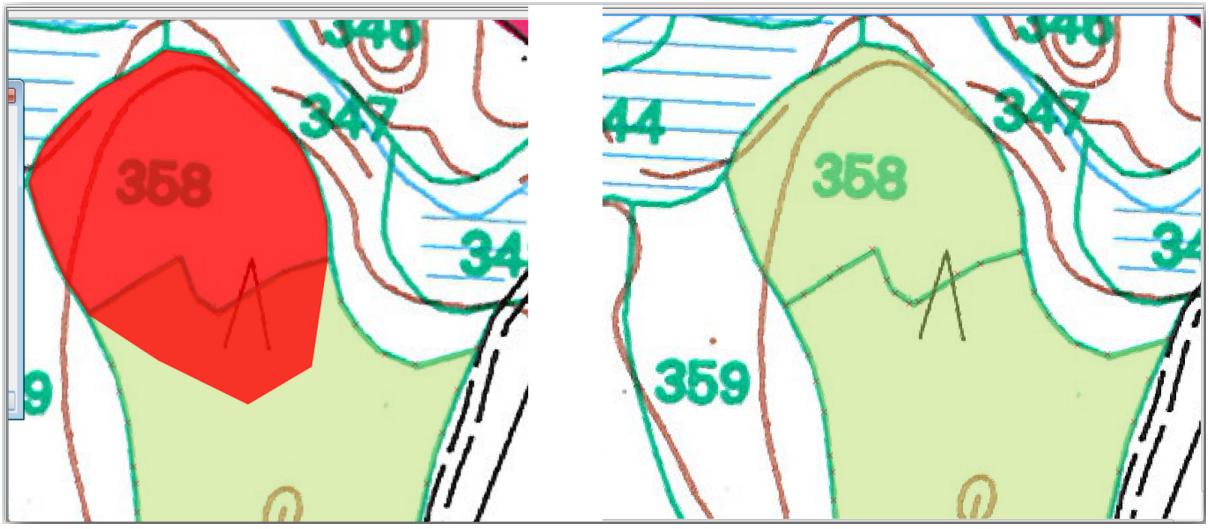
If a form did not appear when you finished digitizing the polygon, go to *Settings* ► *Options* ► *Digitizing* and make sure that the *Suppress attribute form pop-up after feature creation* is not checked.

Your digitized polygon will look like this:



Now for the second polygon, pick up the stand number 358. Make sure that *Avoid Overlap* is checked for the *forest_stands* layer (as shown above). This option ensures polygons do not overlap. So, if you digitize over an existing polygon, the new polygon will be trimmed to meet the border of the existing polygons. You can use this option to automatically obtain a common border.

1. Begin digitizing the stand 358 at one of the common corners with the stand 357.
2. Continue normally until you get to the other common corner for both stands.
3. Finally, digitize a few points inside polygon 357 making sure that the common border is not intersected. See left image below.
4. Right click to finish editing the forest stand 358.
5. Enter the *ID* as `358`.
6. Click *OK*. Your new polygon should have a common border with the stand 357 as you can see in the image below.



The part of the polygon that was overlapping the existing polygon has been automatically trimmed and you are left with a common border - as you intended it to be.

14.3.5 Try Yourself Finish Digitizing the Forest Stands

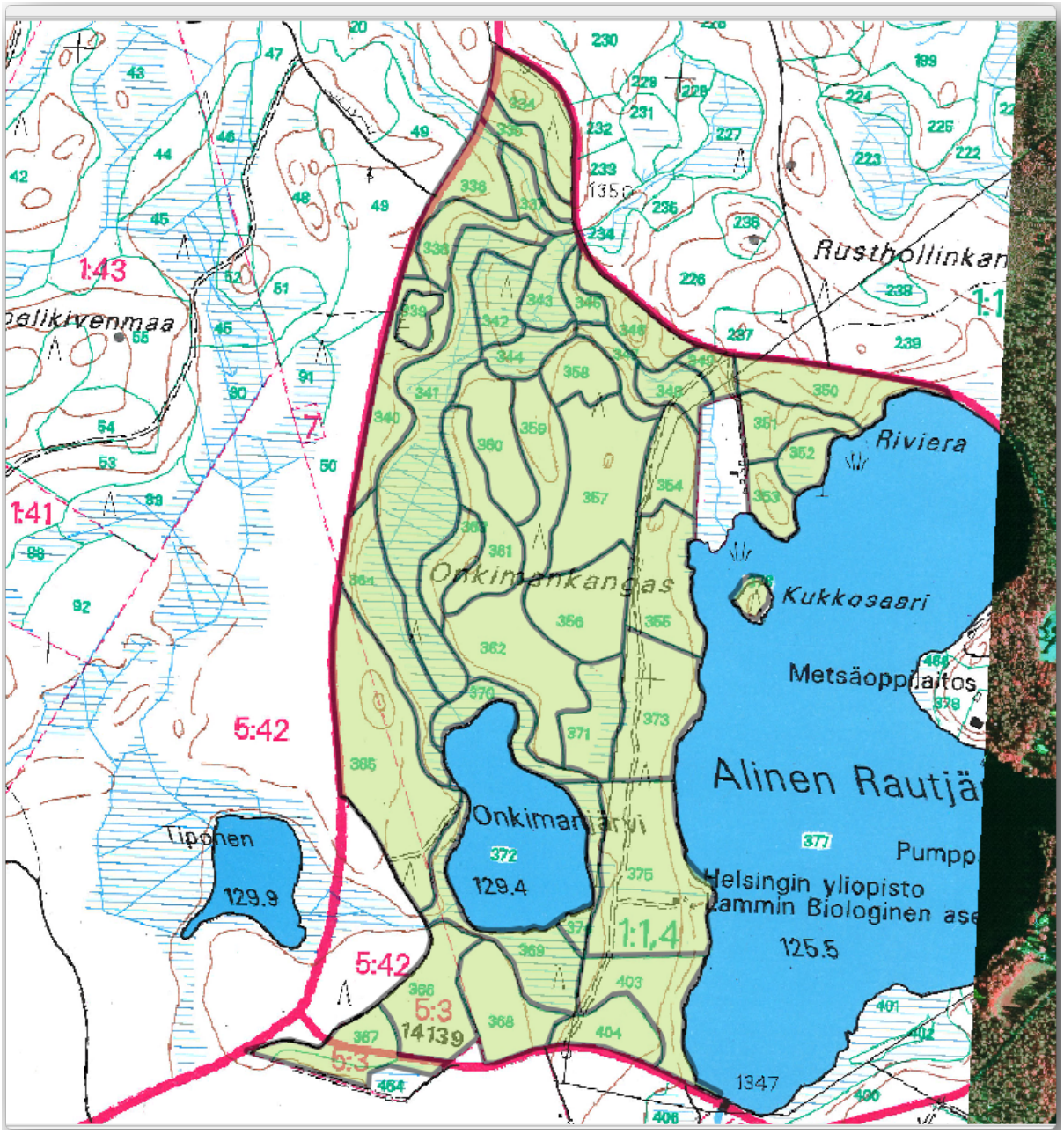
Now you have two forest stands ready. And a good idea on how to proceed. Continue digitizing on your own until you have digitized all the forest stands that are limited by the main road and the lake.

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 *Lesson: Funktionstopologie*, 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:

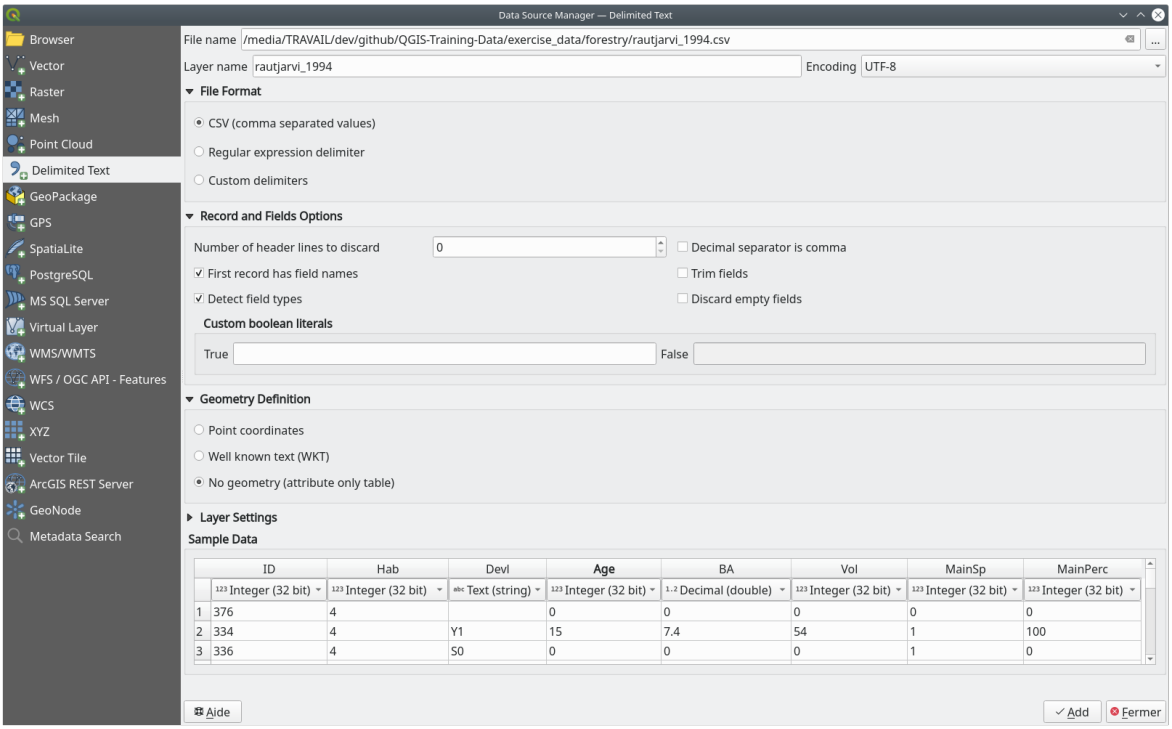


14.3.6 Follow Along: Joining the Forest Stand Data

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.

1. 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.
2. Now add this file into the project:

1. Use the Add Delimited Text Layer tool. This is accessed via *Layer* ► *Add Layer* ► *Add Delimited Text Layer*….
2. Set details in the dialog as follows:



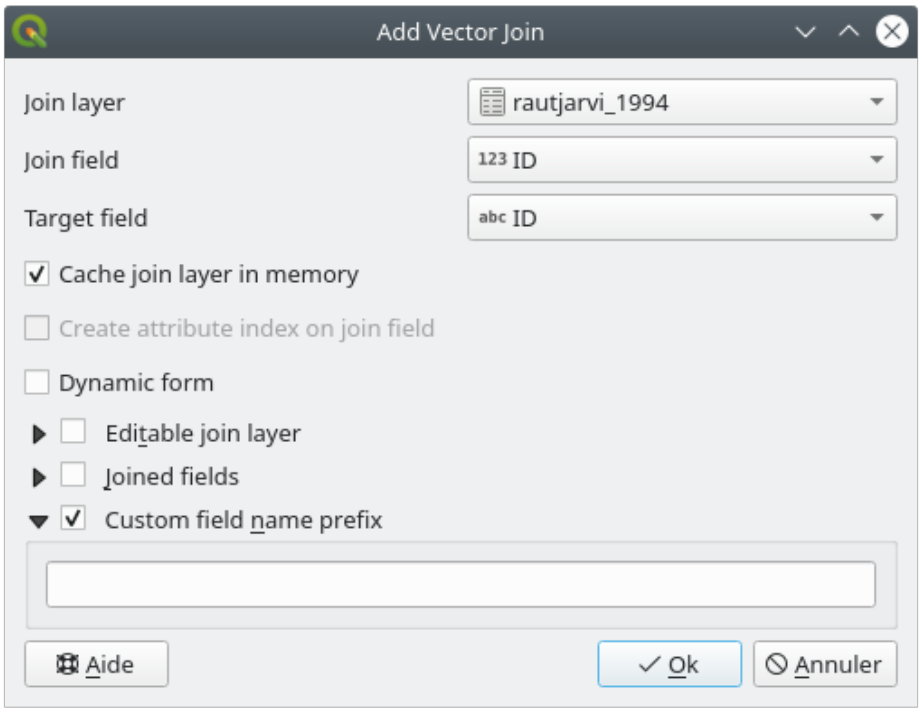
3. Press *Add* to load the formatted `csv` file in the project.

3. To link the data from the `.csv` file with the digitized polygons, create a join between the two layers:
   1. Open the Layer Properties for the `forest_stands` layer.
   2. Go to the *Joins* tab.
   3. Click Add new join on the bottom of the dialog box.
   4. Select *rautjarvi_1994.csv* as the *Join layer*
   5. Set the *Join* field to *ID*
   6. Set the *Target* field to *ID*
   7. Click *OK* two times.

The data from the text file should be now linked to your vector file. To see what has happened, select the `forest_stands` layer and use Open Attribute Table. You can see that all the attributes from the inventory data file are now linked to your digitized vector layer.

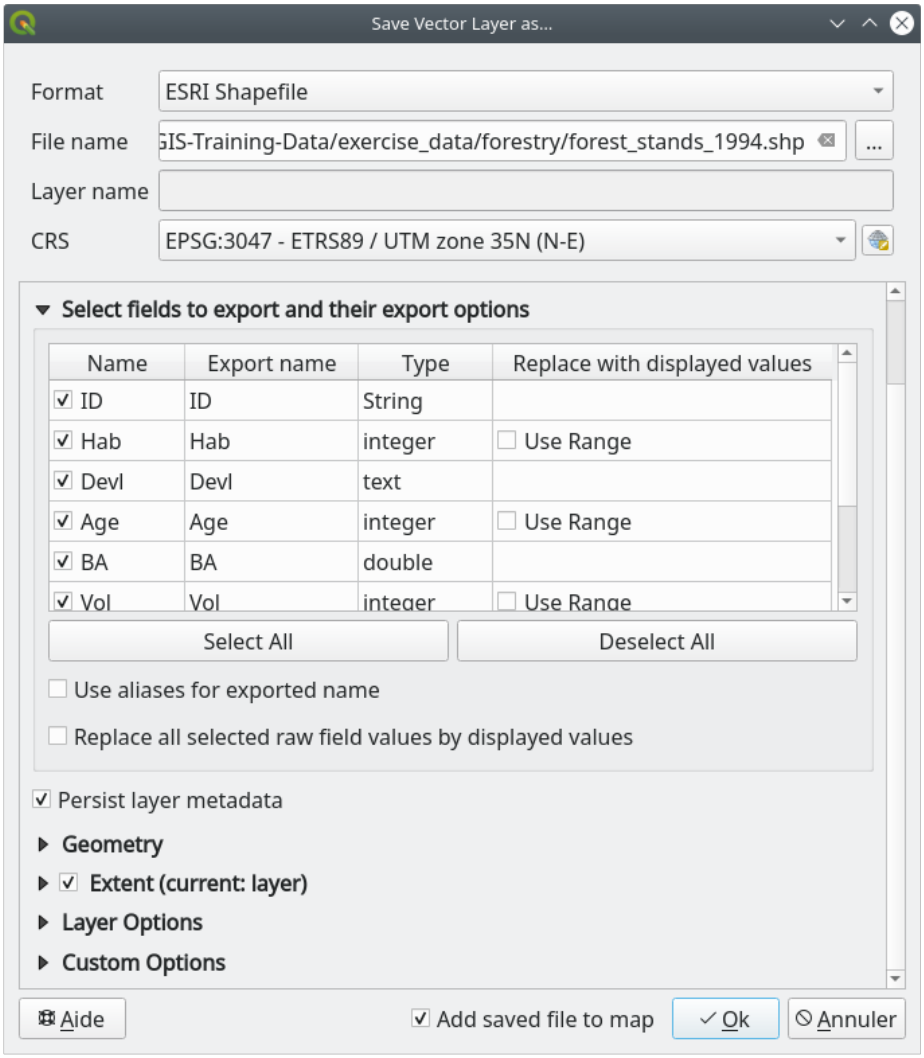
You will see that the field names are prefixed with `rautjarvi_1994_`. To change this:

1. Open the Layer Properties for the `forest_stands` layer.
2. Go to the *Joins* tab.
3. Select *Join Layer rautjarvi_1994*
4. Click the Edit selected join button to enable editing
5. Under *Custom field name prefix* remove the prefix name



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. To do this:

1. Right click on `forest_stands` layer
2. Choose *Export* ► *Save Features As…*
3. Set *Format* to *ESRI Shapefile*
4. Set file name to `forest_stands_1994.shp` under the `forestry` folder
5. To include the new file as a layer in the project, check *Add saved file to map*



14.3.7 Try Yourself Adding Area and Perimeter

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 *Lesson: Supplementary Exercise*. 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. You could also do the same for the perimeter.

Now your `forest_stands_1994` layer is ready and packed with all the available information.

Save your project to keep the current map layers 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 is to create the inventory of the current state of the forest.

You will digitize new forest stands using an aerial photo. As with the previous lesson, you will use an aerial Color Infrared (CIR) photograph. This type of imagery, where the infrared light is recorded instead of the blue light, is widely used to study vegetated areas.

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 datasets.

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 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).

1. Open QGIS and set the project's CRS to *ETRS89 / ETRS-TM35FIN* in *Project* ► *Properties…* ► *CRS*
2. Add the CIR image `rautjarvi_aerial.tif` to the project:
    1. Go to the `exercise_data\forestry\` folder using your file manager browser
    2. Drag and drop the file `rautjarvi_aerial.tif` onto your project
3. 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.

1. Add the `forest_stands_1994.shp` layer created in the previous lesson:
    1. Go to the `exercise_data\forestry\` folder using your file manager browser
    2. Drag and drop the file `forest_stands_1994.shp` onto your project
2. Set the symbology for the layer so that you can see through your polygons:
    1. Right click *forest_stands_1994*
    2. Select *Properties*
    3. Go to the *Symbology* tab
    4. Set *Fill color* to transparent fill
    5. Set *Stroke color* to purple

6. Set *Stroke width* to `0.50 mm`

3. Review how the old forest stands follow (or not) what you might visually interpret as an homogeneous forest.
4. 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.

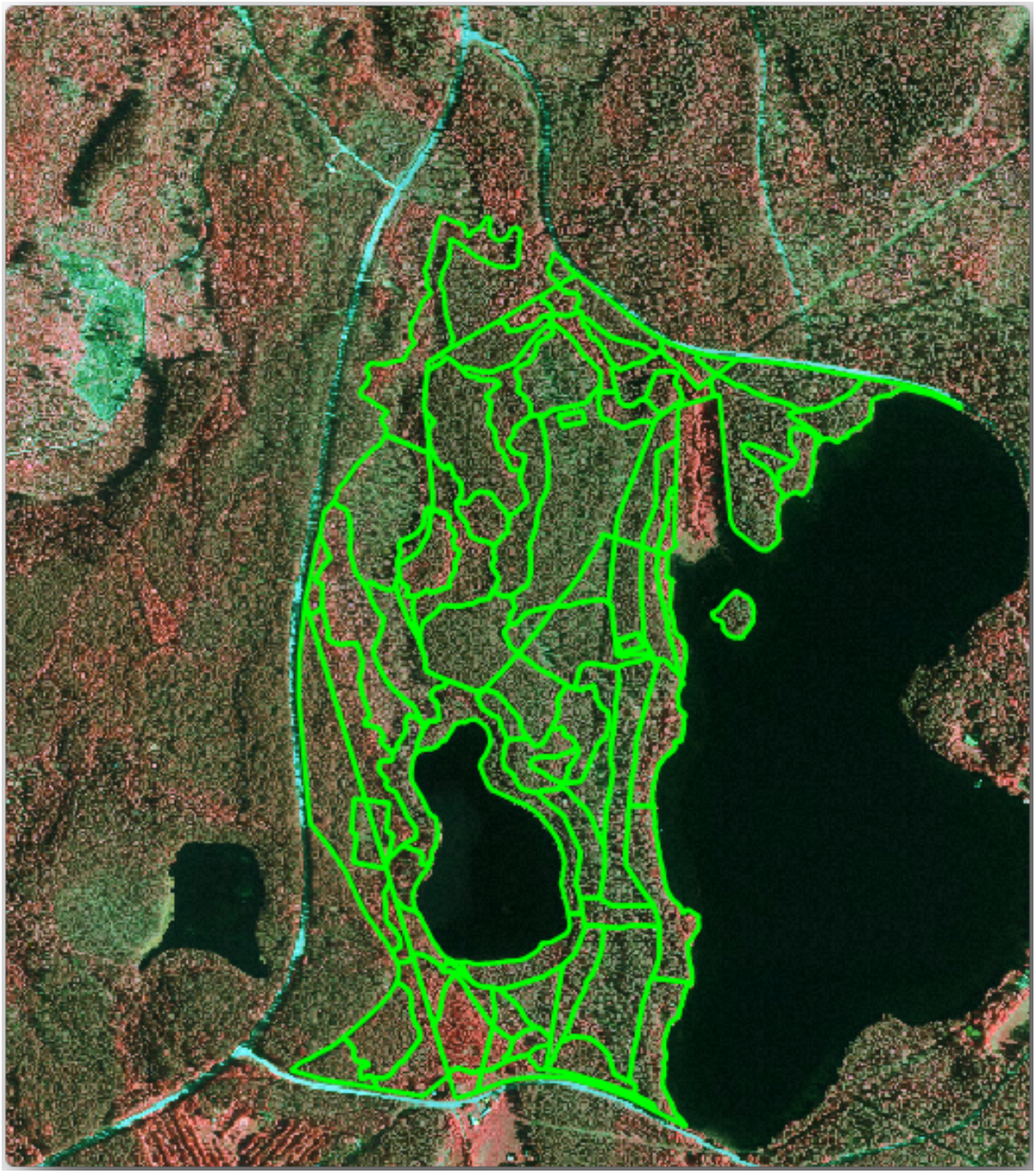
   This is a normal situation, as some 20 years have passed and different forest operations have been carried out (harvesting, thinning…). It is also possible that the forest stands looked homogeneous back in 1992 to the person who digitized them but as time has passed some forest has developed in different ways. It is also possible that that forest inventory priorities back then were different from those of 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.

1. Remove the layer *forest_stands_1994*
2. Add the file `exercise_data\forestry\forest_stands_2012.shp` to the project
3. Set the styling of this layer so that the polygons have no fill and the borders are visible
   1. Open *Properties* dialog of the `forest_stands_2012` layer
   2. Go to the *Symbology* tab
   3. Set *Fill color* to transparent fill
   4. Set *Stroke color* to green
   5. Set *Stroke width* to `0.50 mm`



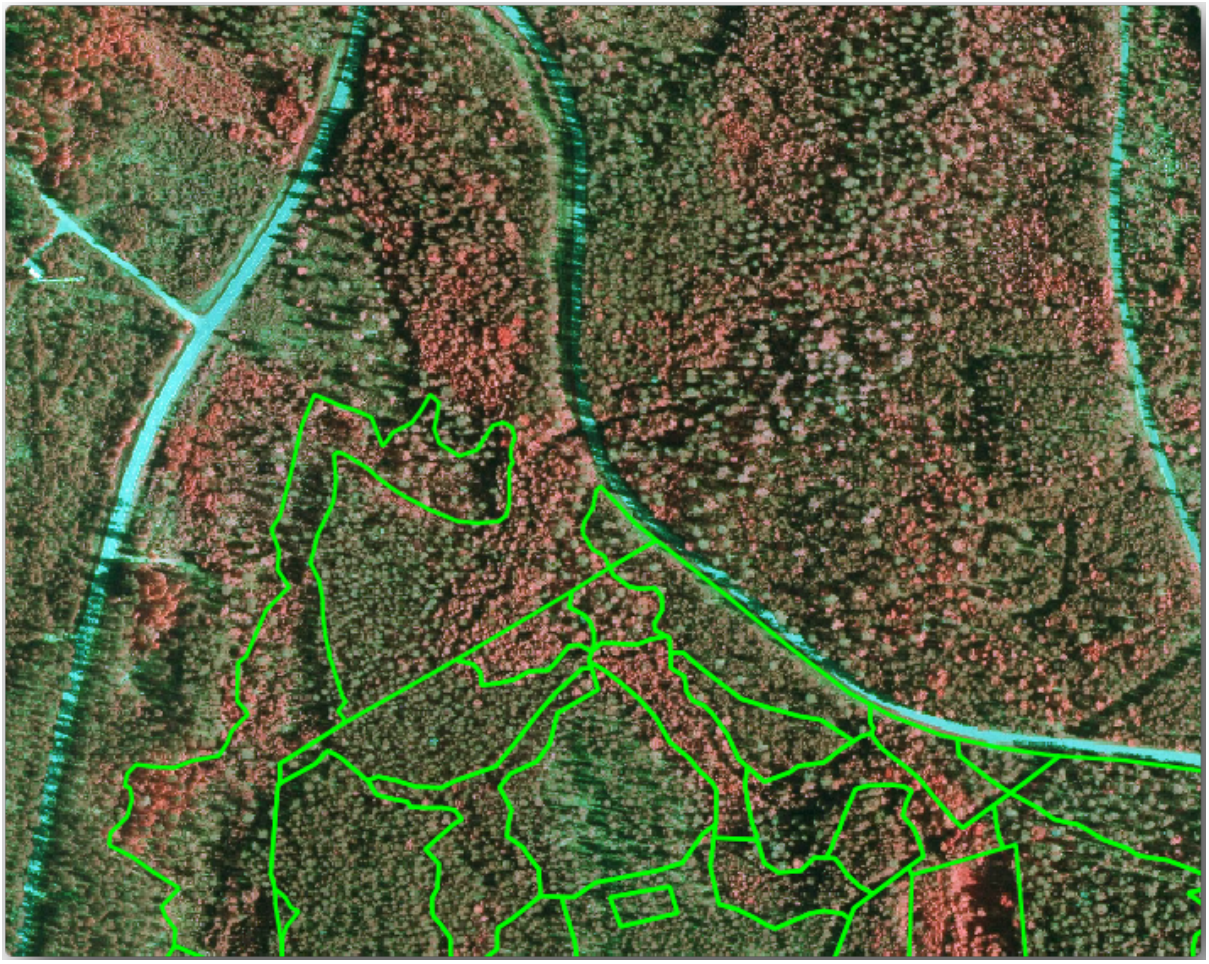
You can see that the northern section of the inventory area is still missing. Your task is to digitize the missing forest stands.

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 points to consider:

- Which forests have deciduous species (in Finland these are mostly birch forests) and which ones have conifers (in this area these are pine or spruce)? In CIR images, deciduous species usually show up as a bright red color whereas conifers show as a dark green color.

- How old is the forest? The size of the tree crowns can be identified in the imagery.
- How dense are the different forest stands? A forest stand where a thinning operation has recently been done would 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):



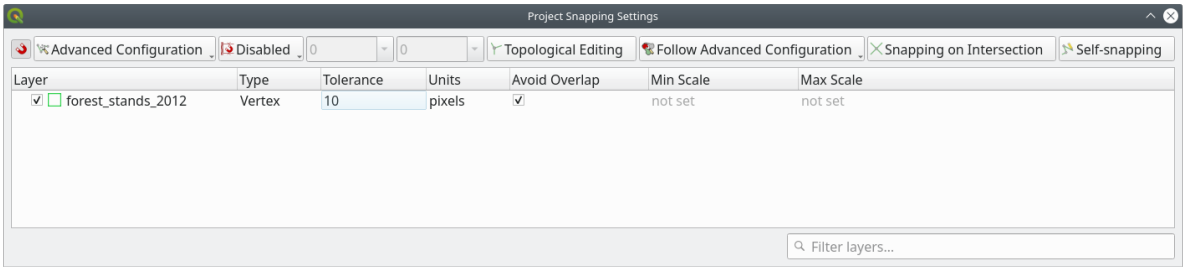
14.4.3 Try Yourself Digitizing Forest Stands from CIR Imagery

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 - and 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).

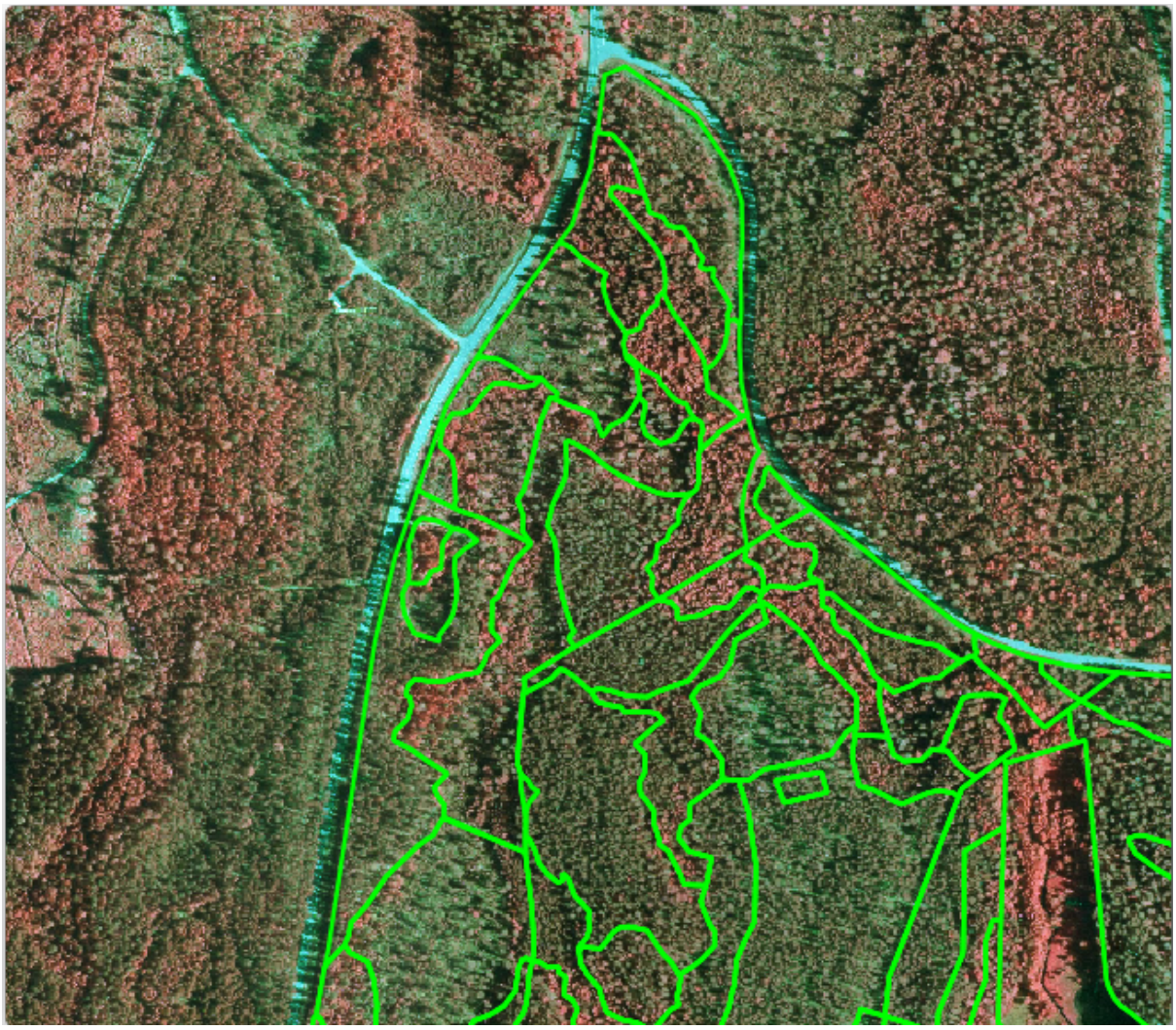
With these points in mind, you can now digitize the missing forest stands.

1. Set up the snapping and topology options:
    1. Go to *Project* ► *Snapping options…*
    2. Press Enable Snapping and select *Advanced Configuration*
    3. Check the *forest_stands_2012* layer
        1. Set *Type* to *Vertex*

2. Set *Tolerance* to `10`
3. Set *Units* to *pixels*
4. Check the box under *Avoid Overlap*
5. Press *Topological editing*
6. Choose *Follow Advanced Configuration*
7. Close the pop-up



2. Select the *forest_stands_2012* layer on the Layers list
3. Click the Toggle Editing button to enable editing
4. Start digitizing using the same techniques as in the previous lesson. The only difference is 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 `StandID` field with numbers starting at `901`.
5. When you are finished your layer should look something like:



Now you have a new set of polygons showing the different forest stands in 2012 - as interpreted from the CIR images. However, you are missing the forest inventory data. For that you will 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.

You can add some extra information about conservation regulations that need to be taken into account for this area.

14.4.4 Follow Along: Updating Forest Stands with Conservation Information

For the area you are working in, there are some conservation regulations that must be taken into account when 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 that is 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 a vector file containing the information about the squirrel locations and another containing the digitized stream running from the North area towards the lake.

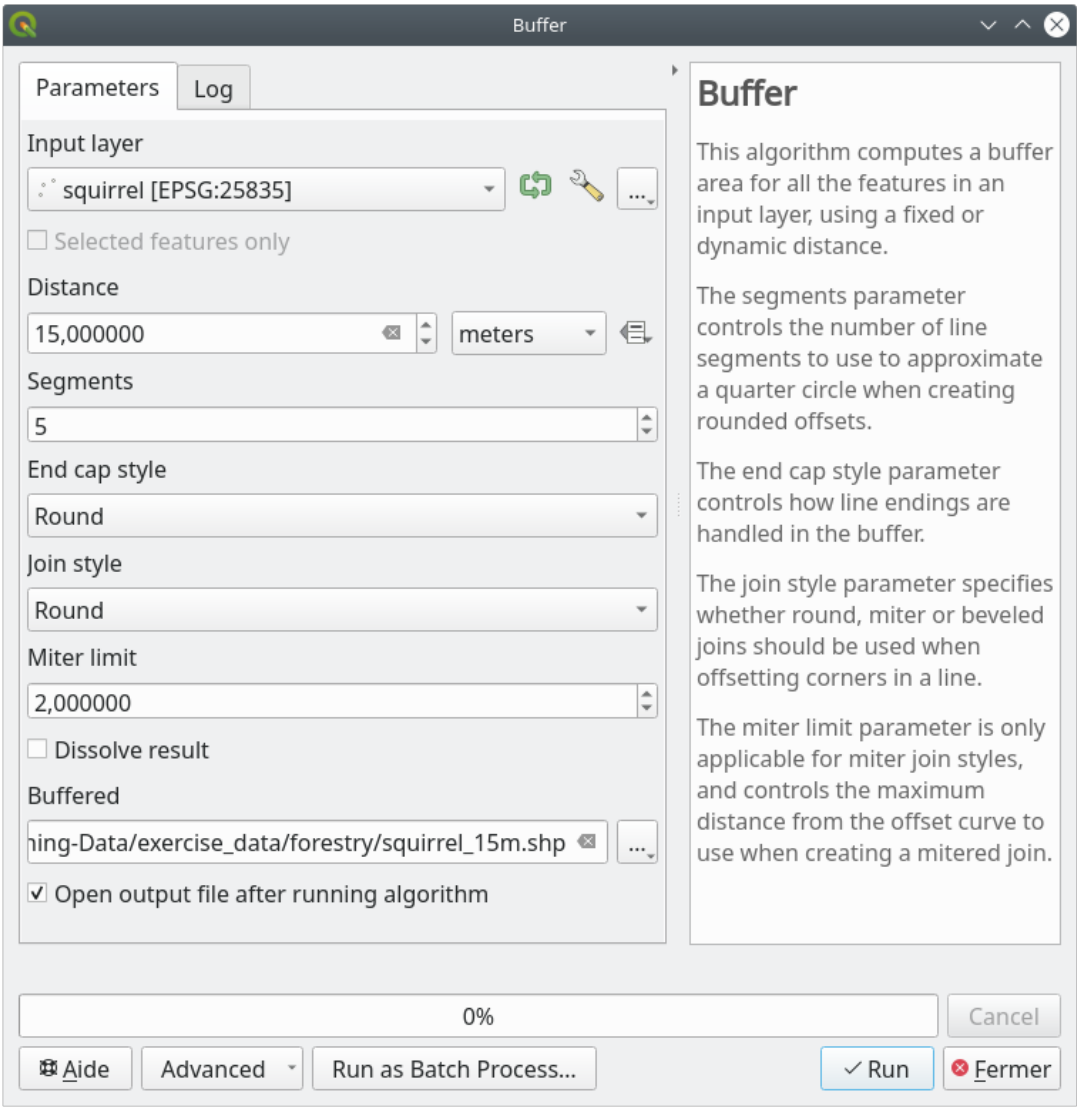
1. From the `exercise_data\forestry\` folder, add the `squirrel.shp` and `stream.shp` files to the project.

2. Use the Open Attribute Table tool to view the `squirrel` layer

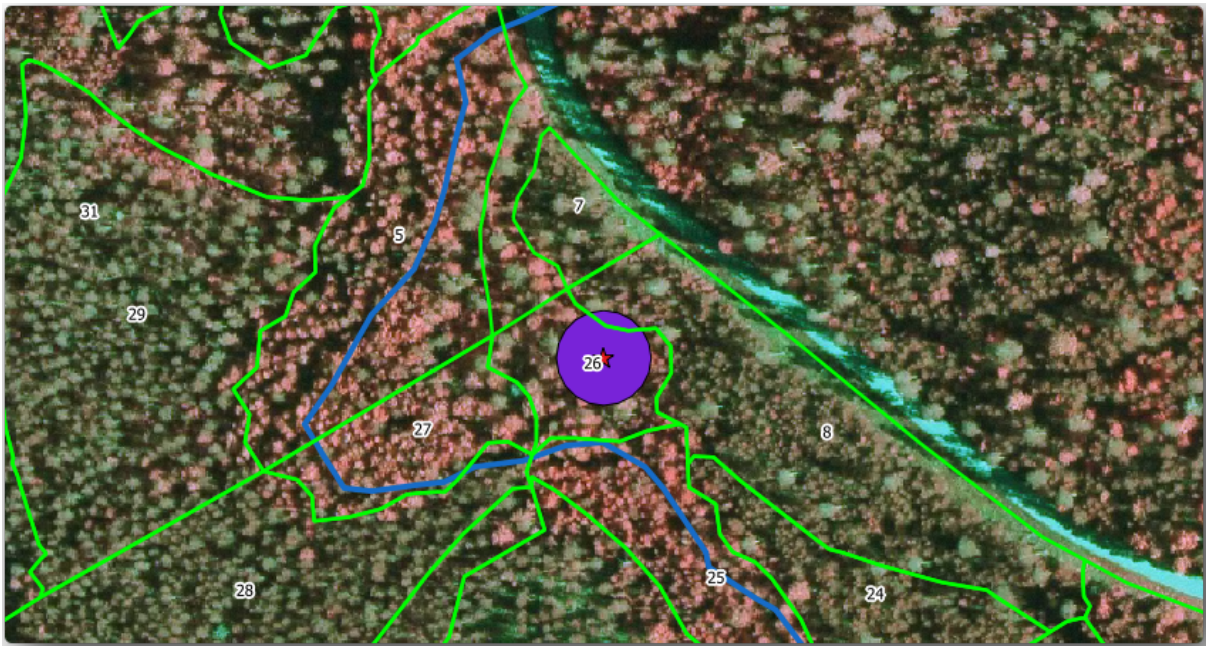
   You can see that there are two locations that are defined as Siberian flying squirrel, and that the area to be protected is indicated by a distance of 15 meters from the locations.

Let's more accurately delimitate that area to protect. We will create a buffer around the point locations, using the protection distance.

1. Open *Vector* ► *Geoprocessing Tools* ► *Buffer*.
2. Set *Input layer* to *squirrel*
3. Set *Distance* to `15 meters`
4. Set *Buffered* to `exercise_data\forestry\squirrel_15m.shp`
5. Check *Open output file afer running algorithm*
6. Click *Run*
7. Once the process is completed, click *Close*



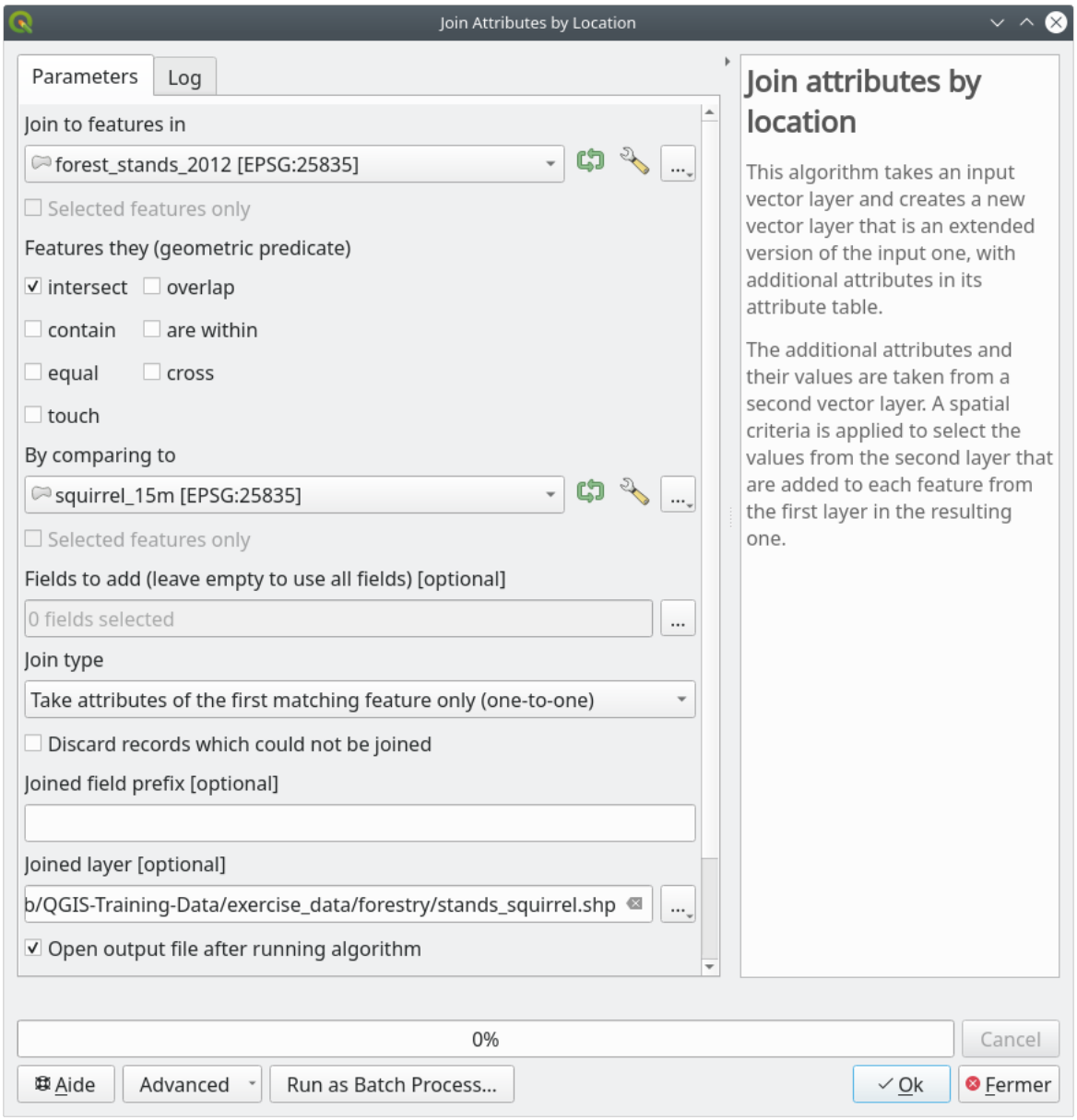
If you zoom in to the location in the northern part of the area, you will notice that the buffer area extends over two neighbouring stands. This means that whenever a forest operation takes place in that stand, the protected location should also be taken into account.



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 locations that have to be protected. This information will then 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.

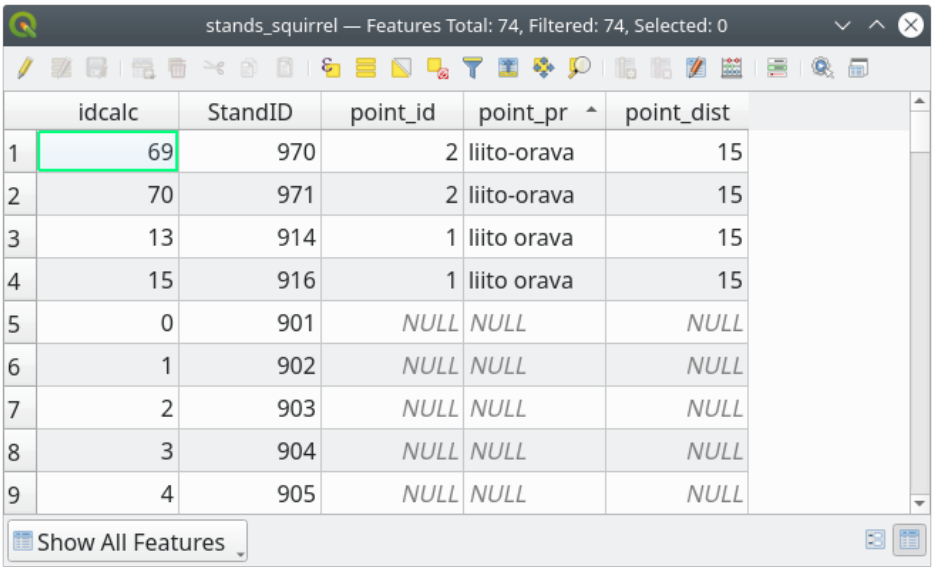
To join the information about the squirrels to your forest stands, you can use the *Join attributes by location* algorithm:

1. Open *Vector* ► *Data Management Tools* ► *Join attributes by location*.
2. Set *Join to features in* to *forest_stands_2012*
3. In *Geometric predicate*, check *intersect*
4. Set *By comparing to* to *squirrel_15m*
5. Set *Join type* as *Take attributes of the first matching feature only (one-to-one)*
6. Leave unchecked *Discard records which could not be joined*
7. Set *Joined layer* to `exercise_data\forestry\stands_squirrel.shp`
8. Check *Open output file afer running algorithm*
9. Click *Run*
10. Once the process is completed, you can *Close* the dialog.



Now you have a new forest stands layer, `stands_squirrel.shp` showing the protection information for the Siberian flying squirrel.

1. Open the attribute table of the `stands_squirrel` layer
2. Sort the table by clicking on *point_pr* field in the table header.



	idcalc	StandID	point_id	point_pr	point_dist
1	69	970	2	liito-orava	15
2	70	971	2	liito-orava	15
3	13	914	1	liito orava	15
4	15	916	1	liito orava	15
5	0	901	NULL	NULL	NULL
6	1	902	NULL	NULL	NULL
7	2	903	NULL	NULL	NULL
8	3	904	NULL	NULL	NULL
9	4	905	NULL	NULL	NULL

You can see that there are some forest stands that have the information about the protection locations. The information in the forest stands data will indicate to the forest manager that there are protection considerations to be taken into account. Then he or she can get the location from the *squirrel* dataset, and visit the area to mark the corresponding buffer around the location so that the operators in the field can avoid disturbing the squirrels environment.

14.4.5 Try Yourself Updating Forest Stands with Distance to the Stream

Following the same approach as for the protected squirrel locations you can now update your forest stands with protection information related to the stream. A few points:

- Remember the buffer is `20` meters around the stream
- You want to have all the protection information in the same vector file, so use `stands_squirrel.shp` as the base layer
- Name your output as `forest_stands_2012_protect.shp`

Once the process is completed, open the attribute table of the output layer and confirm that you have all the protection information for the riparian forest stands associated with the stream.

When you are happy with the results, 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`

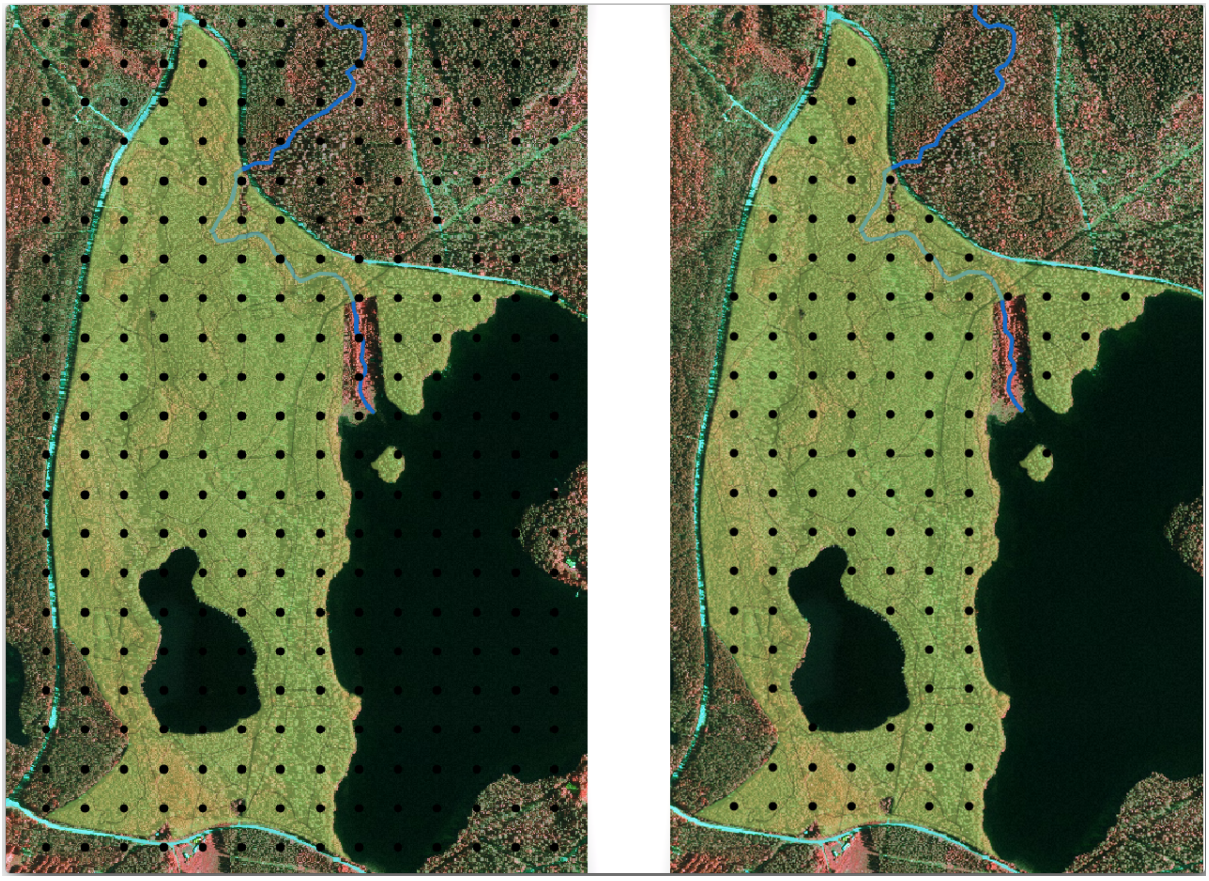
Now you need to create a rectangular grid of points separated 80 meters from each other:

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*.

Bemerkung: 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:

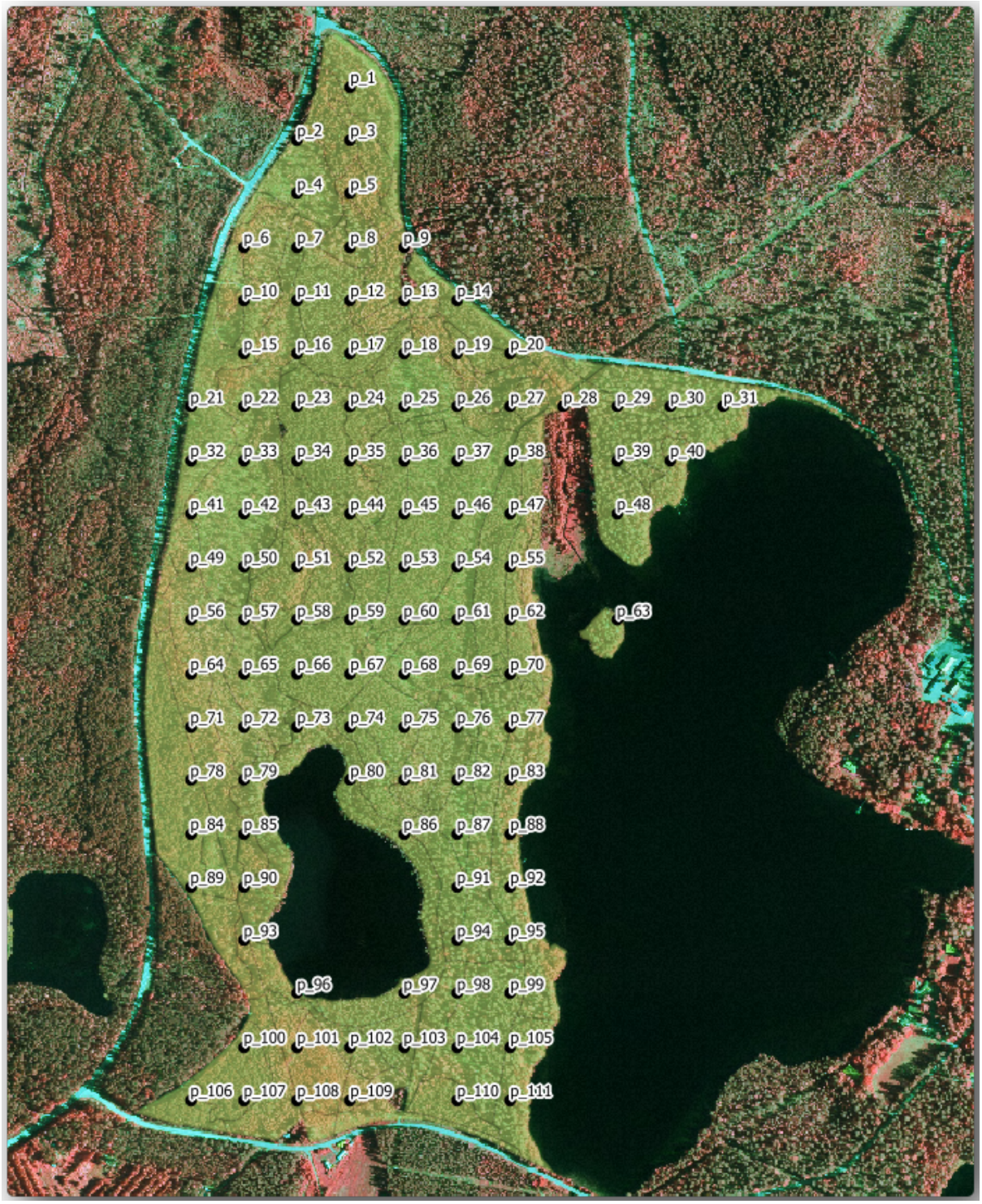
1. Open the *Layer Properties* ► *Labels* for the `systematic_plots_clip` layer.
2. Turn the top menu into *Single Labels*.
3. For the *Value* entry, select the field `id`.
4. Go to the *Buffer* tab, check the *Draw text buffer* and set the buffer *Size* to `1`.
5. Click *OK*.

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.

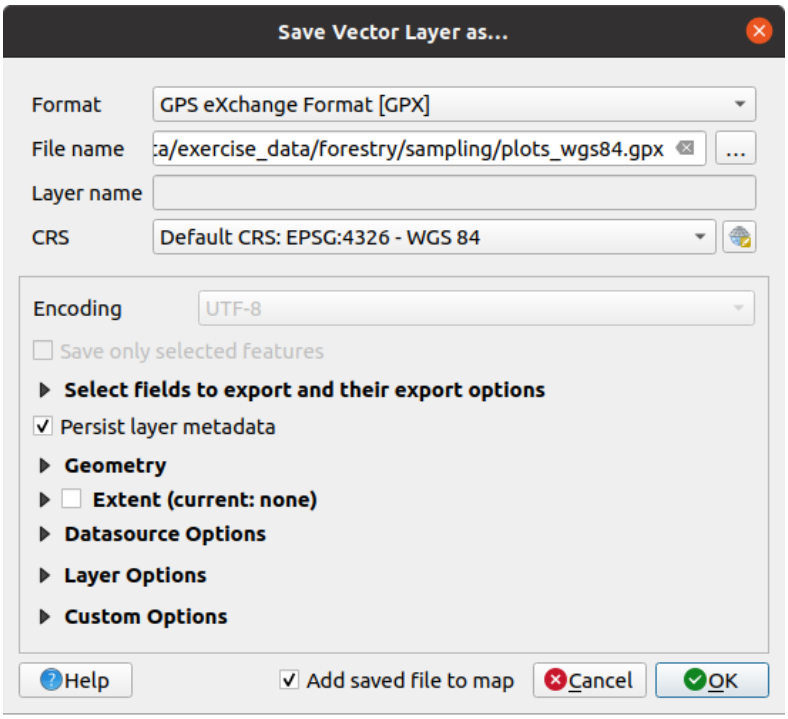
Now you have a new column with plot names that are meaningful to you. For the `systematic_plots_clip` layer, change the field used for labeling to your new `Plot_id` field.


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*.

> **Bemerkung:** 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.

1. 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*
2. Save the project with a new name, `map_creation.qgs`.

To create a printable map, remember that you use the *Layout Manager*:

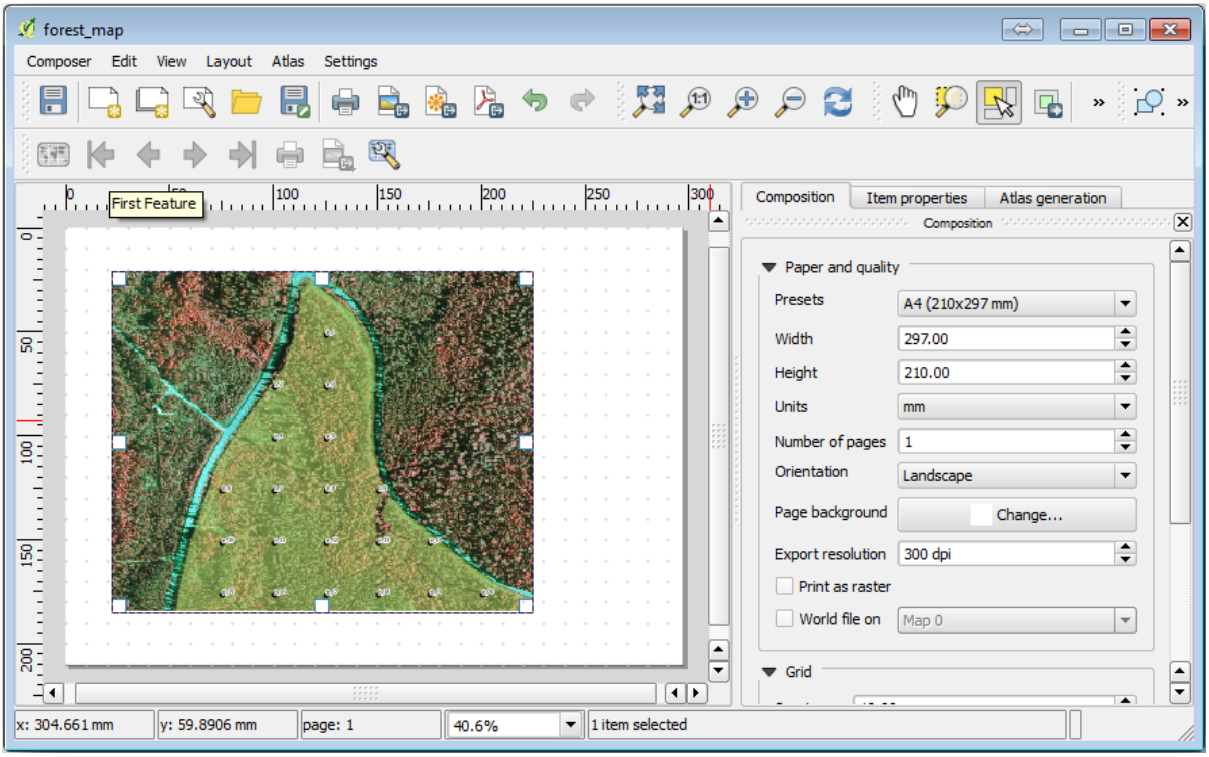
1. Open *Project* ► *Layout Manager…*.
2. In the *Layout manager* dialog:
   1. Under *New from template*, press the *Create…* button next to the *Empty layout* entry
   2. Name your print layout `forest_map`.
   3. Press *OK*. A new print layout is created and opened, showing a blank paper.
3. In the print Layout window, ensure properties are set for an A4 paper:
   1. Right-click over the paper and select *Page properties*. The *Page properties* panel is open on the right side of the layout.
   2. Check *Size* is *A4*.
   3. Check *Orientation* is *Landscape*.
4. Enable the *Layout* tab next to the *Page properties* panel and set *Export resolution* to `300 dpi`.

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:

1. In the *Layout* tab expand the *Guides and Grid* region.
2. Check that *Grid spacing* is set to `10 mm` and that *Snap tolerance* is set to `5 px`.

You need to activate the use of the grid:

1. Open the *View* menu.
2. Check *Show grid*.
3. Check *Snap to grid*.
4. Notice that options for using *Guides* are checked by default, which allows you to see guiding lines when you are moving elements in the layout.
5. 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:
    1. Click on the Add Map button.
    2. 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.

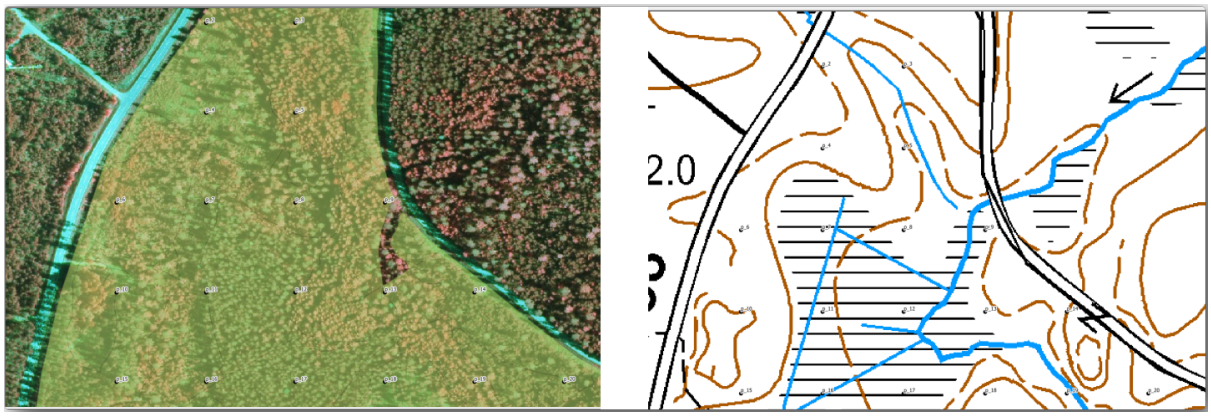
14.6.2 Follow Along: Adding Background Map

Leave the layout open but go back to the map. Let's add some background data and create some styling so that the map content is as clear as possible.

1. Add the background raster `basic_map.tif` that you can find in the `exercise_data\forestry\` folder.
2. If 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.
3. 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:



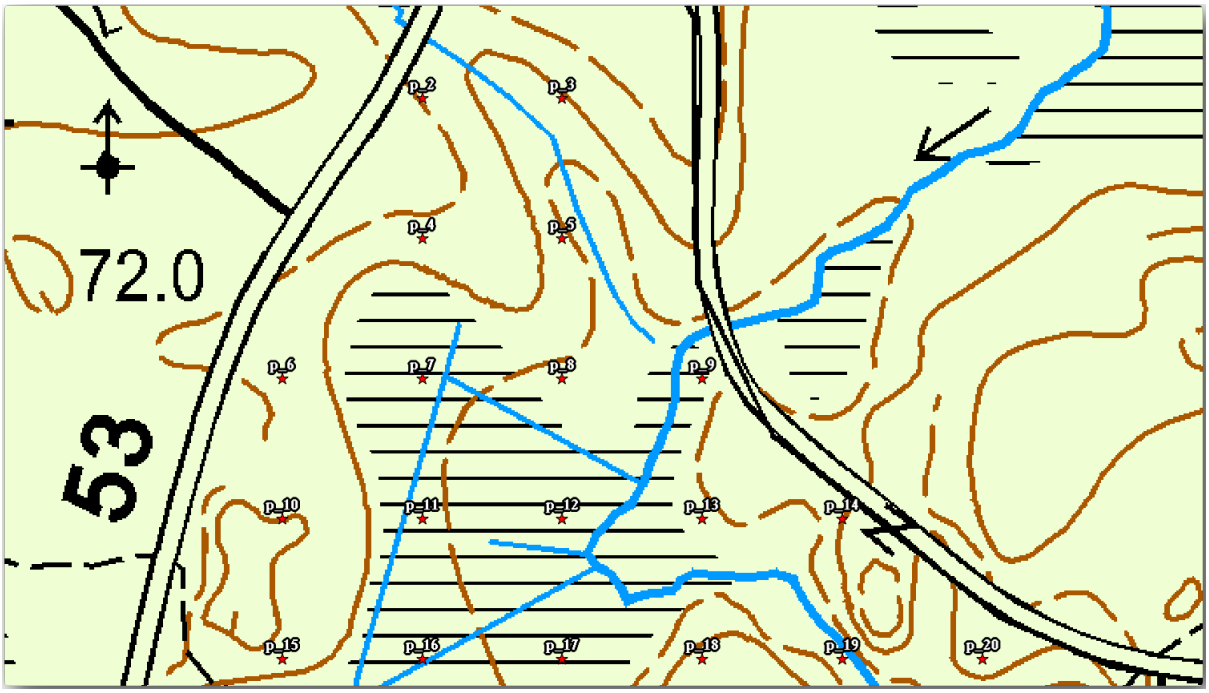
While during the last exercises, the white buffer was OK on top of the aerial image, now that the background image is mostly white you barely can see the labels. But how does it look in the print layout? Let's check it:

1. Go to the print layout window.
2. Use the Select/Move item button to select the map element in the layout.
3. Go to the *Item properties* tab.
4. Click on *Set map extent to match main canvas extent*.
5. If you need to refresh the element, click on *Update map 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

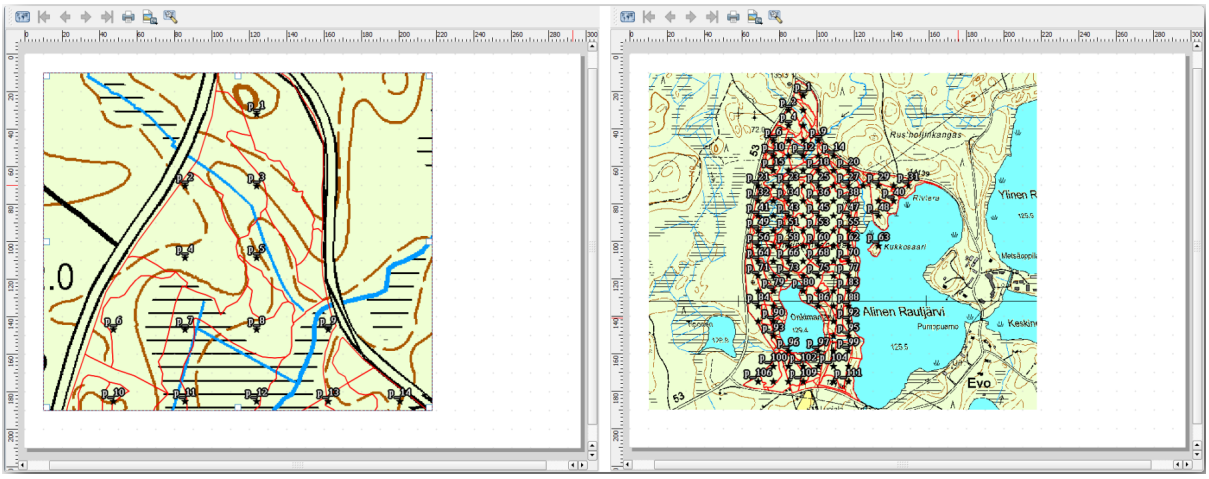
You have been working in *Module: Erstellen und Erkunden einer einfachen Karte* with symbology and in *Module: Vektordaten klassifizieren* with labeling. Go back to those modules if you need to refresh about some of the available options and tools. Your goal is to get the plots locations and their name 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:

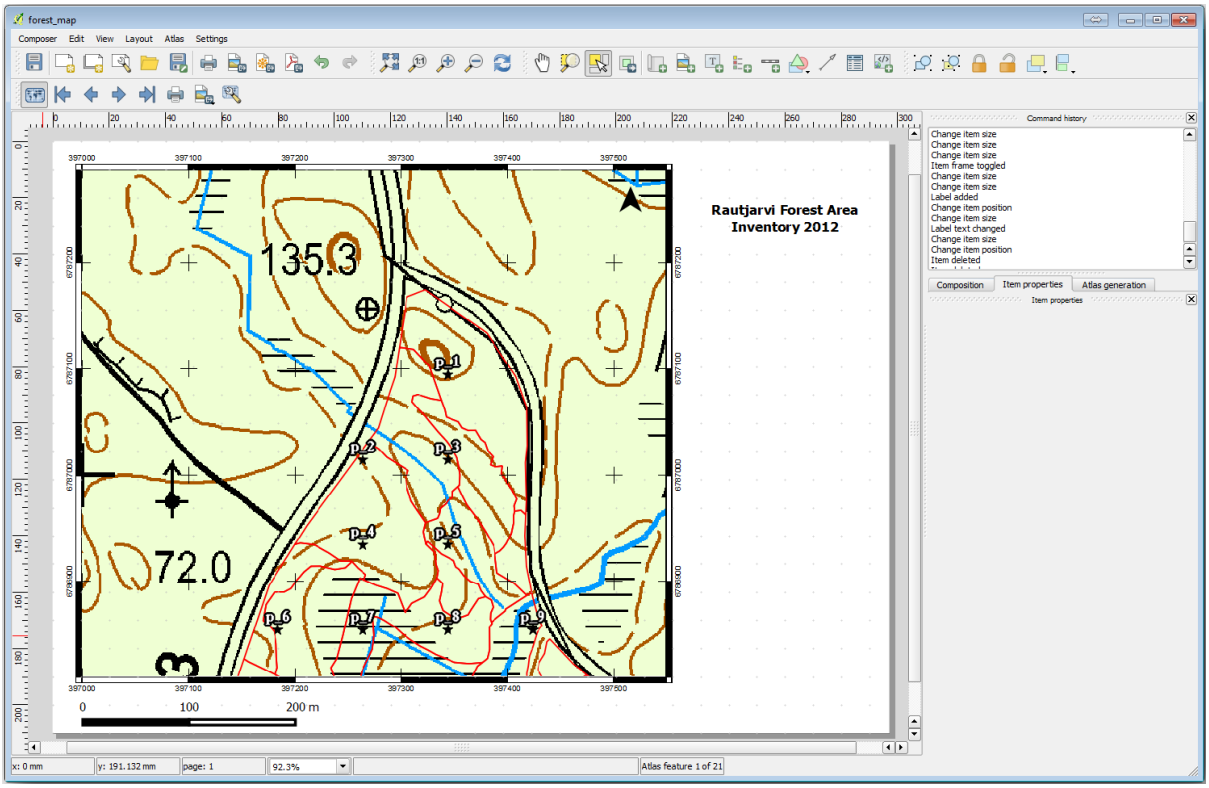
1. Right click on *forest_stands_2012* and select *Duplicate*
2. You get a new layer named `forest_stands_2012 copy` that you can use to define a different style, for example with no filling and red borders.

   Now you have two different visualizations of the forest stands and you can decide which one to display for your detail map.
3. 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 map preview and *Set map extent to match main canvas extent* whenever you change the zoom in your map or the layout.



14.6.4 Try Yourself Create a Basic Map Template

1. Once you have a symbology you are 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.
2. You have created a similar layout already in *Module: Layout der Karten*. Go back to that module as you need. You can look at this example image for reference:



3. Export your map as an image and look at it.
   1. *Layout* ► *Export as Image…*.
   2. Use for example the *JPG format*.

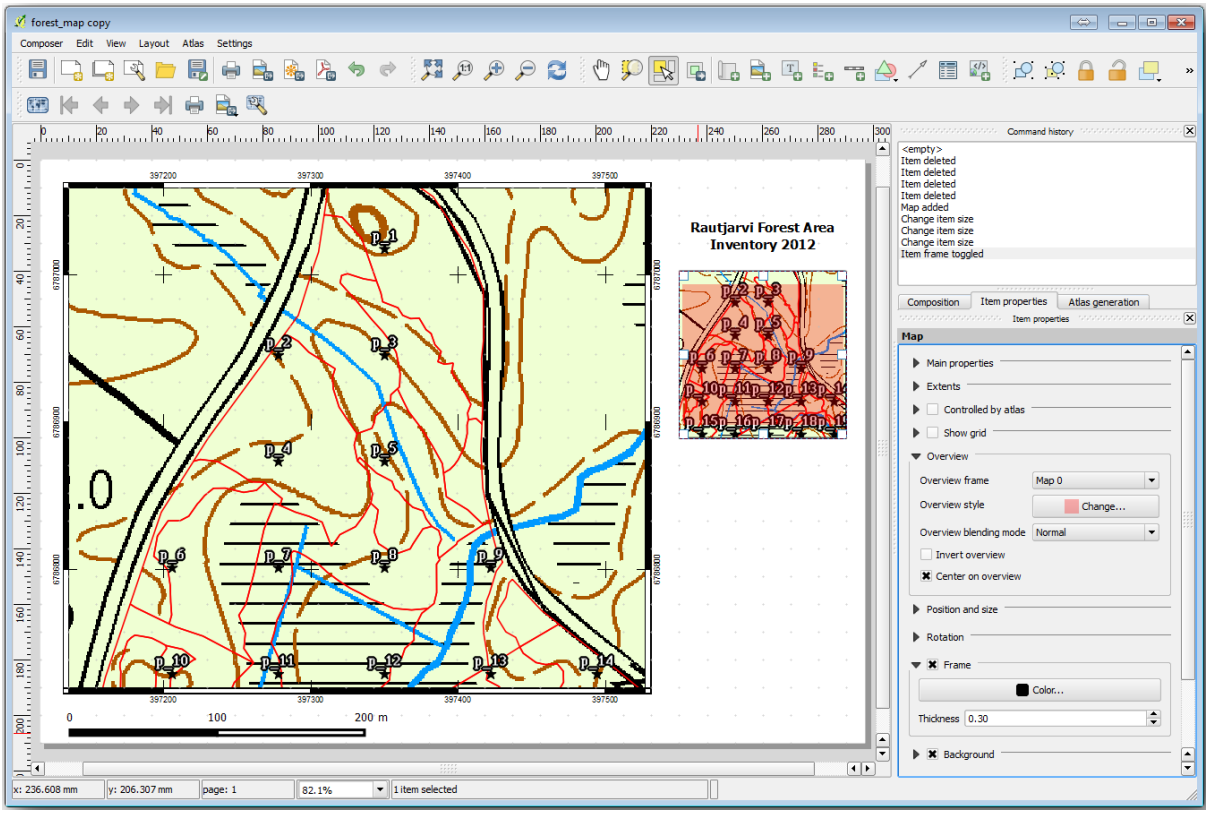
That is what it will look like when printed.

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:

1. Add another map element to the canvas, right under the title text.
2. In the *Item properties* tab, open the *Overview* dropdown.
3. Set the *Overview frame* to *Map 0*. This creates a shadowed rectangle over the smaller map representing the extent visible in the bigger map.
4. 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.

1. Go back to the map, but don't close the *Print Layout*.
2. Right click the *forest_stands_2012* layer and click on *Zoom to Layer Extent*.
3. Deactivate all layers except for *basic_map* and *forest_stands_2012*.
4. Expand the Manage map themes tool on the *Layers* panel and select *Add theme*.
5. Name it `basic_overview`.
6. Go back to the print layout.
7. With the small map selected, click the *Set map extent to match main canvas extent* to set its extents to what you can see in the map window.

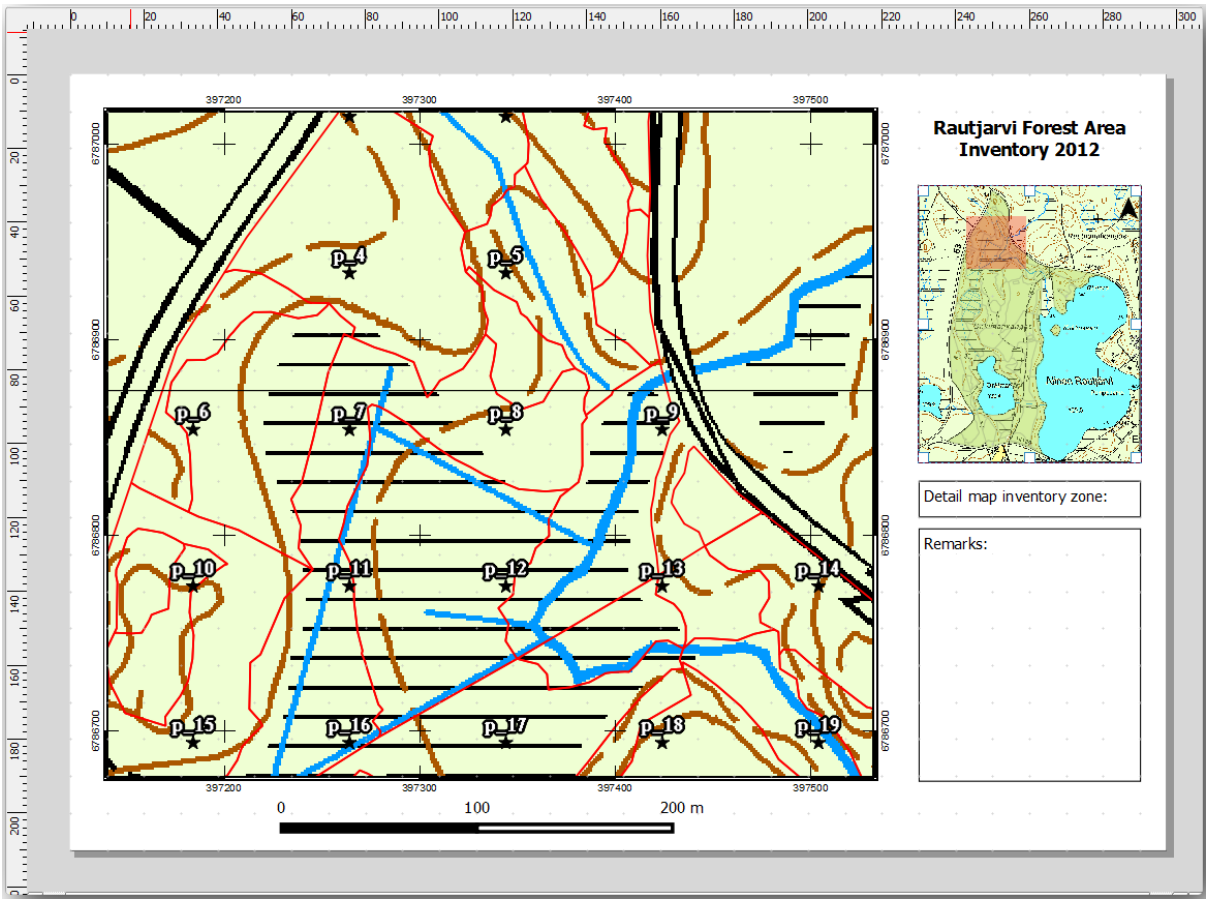
8. Lock the view for the overview map by checking *Follow map theme* and selecting `basic_overview` under *Main properties*.

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. Let's fix that:

1. 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`).
2. Zoom again to have only a few lines of sample plots visible.
3. Go back to the *Print Layout* window.
4. Select the bigger map in your layout.
5. In *Item properties* click on *Update preview* and *Set map extent to match main 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:

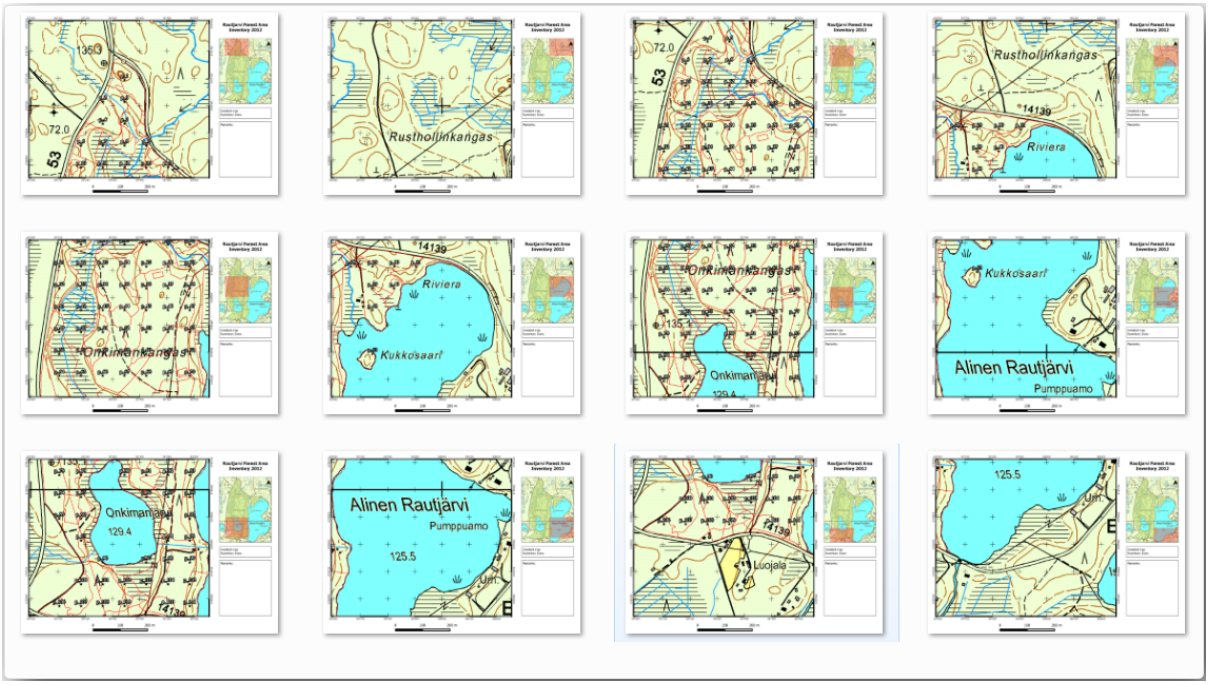
1. Select the Add North Arrow tool.
2. Click and drag a rectangle at the upper right corner of the overview map.
3. In *Item properties* Check *SVG image*
4. Browse *SVG Browser* ► *SVG Groups* for an arrow image.

5. Under *Image rotation*, check the *Sync with map* and select *Map 1* (the overview map).
6. 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.

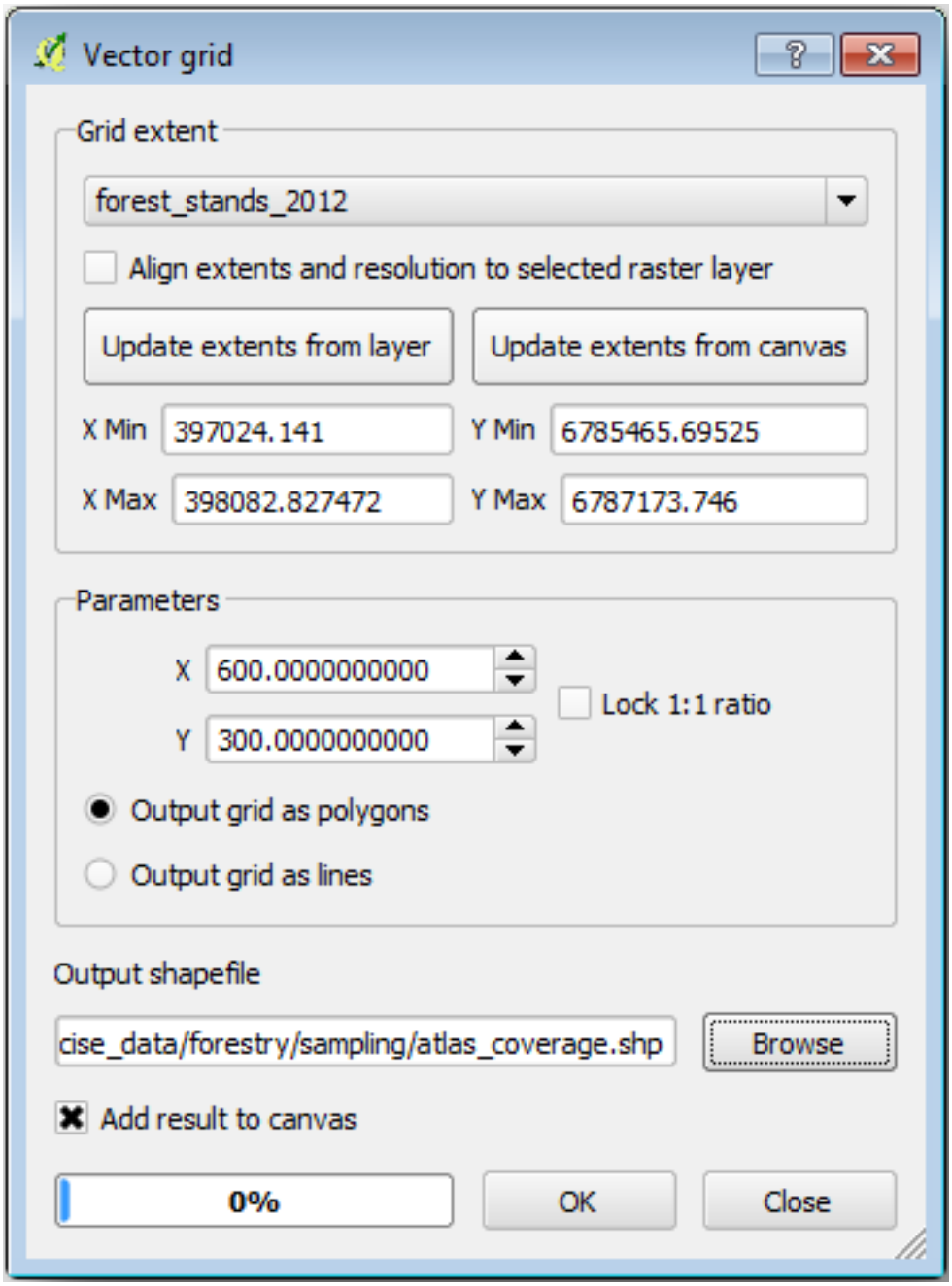
14.6.6 Follow Along: Creating an Atlas Coverage

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:



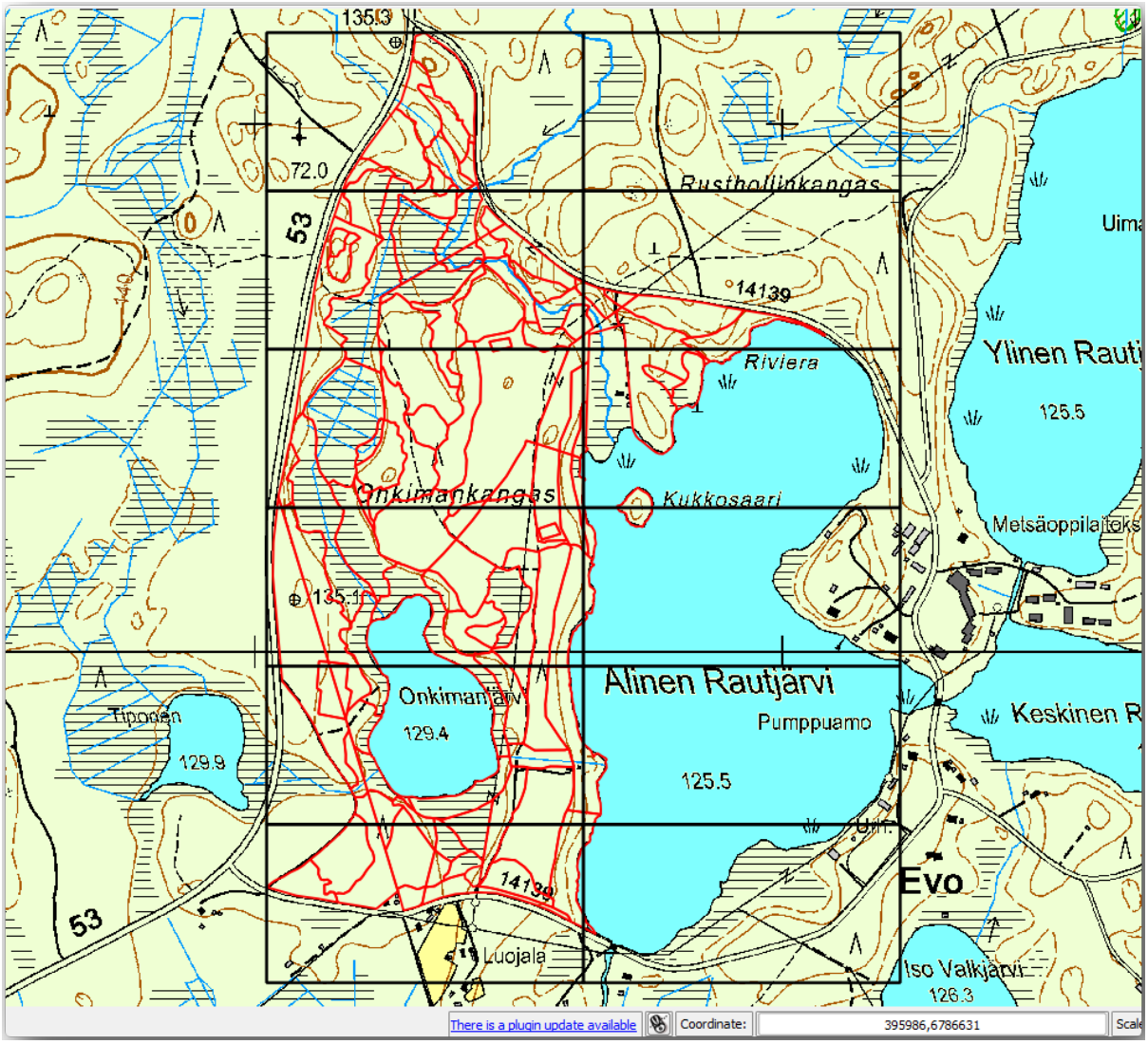
The coverage could be any existing layer, but usually it makes more sense to create one for the specific purpose. Let's create a grid of polygons covering the forest area:

1. In the QGIS map view, open *Vector* ► *Research Tools* ► *Create grid*.
2. Set the tool as shown in this image:



3. Save the output as `atlas_coverage.shp`.
4. Style the new `atlas_coverage` layer so that the polygons have no filling.

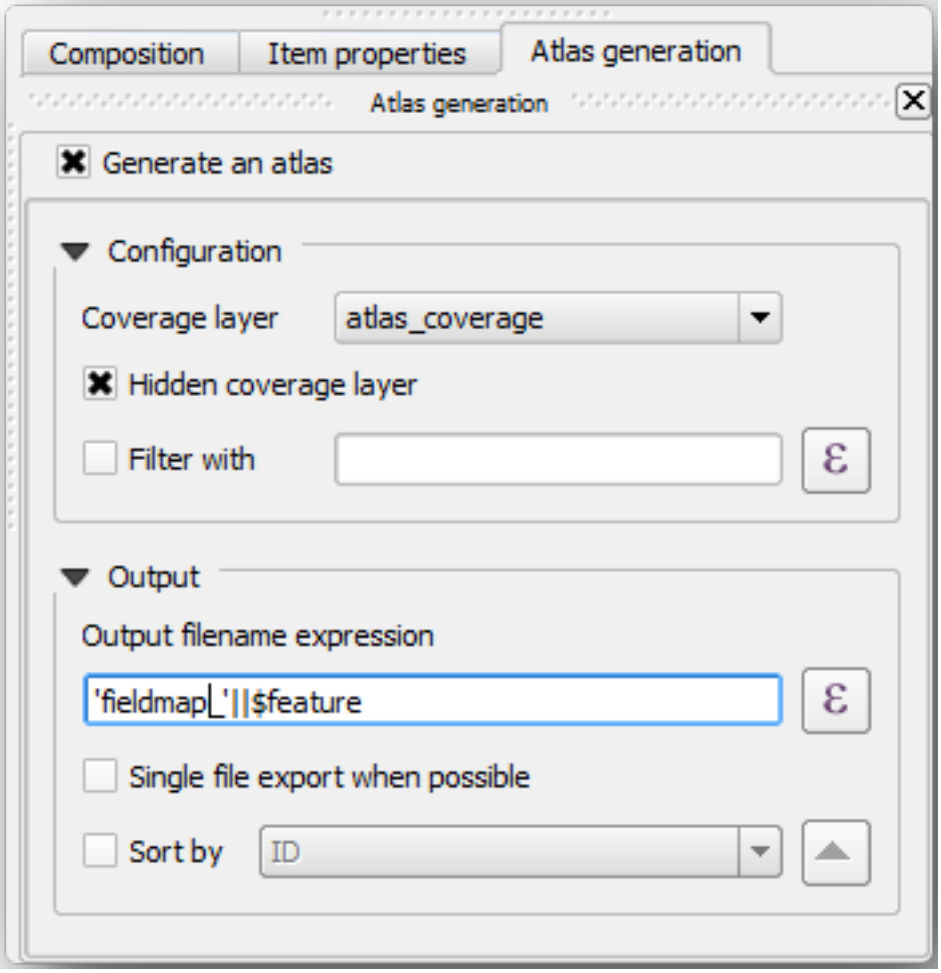
The new polygons are covering the whole forest area and they give you an idea of what each map (created from each polygon) will contain.



14.6.7 Follow Along: Setting Up the Atlas Tool

The last step is to set up the Atlas tool:

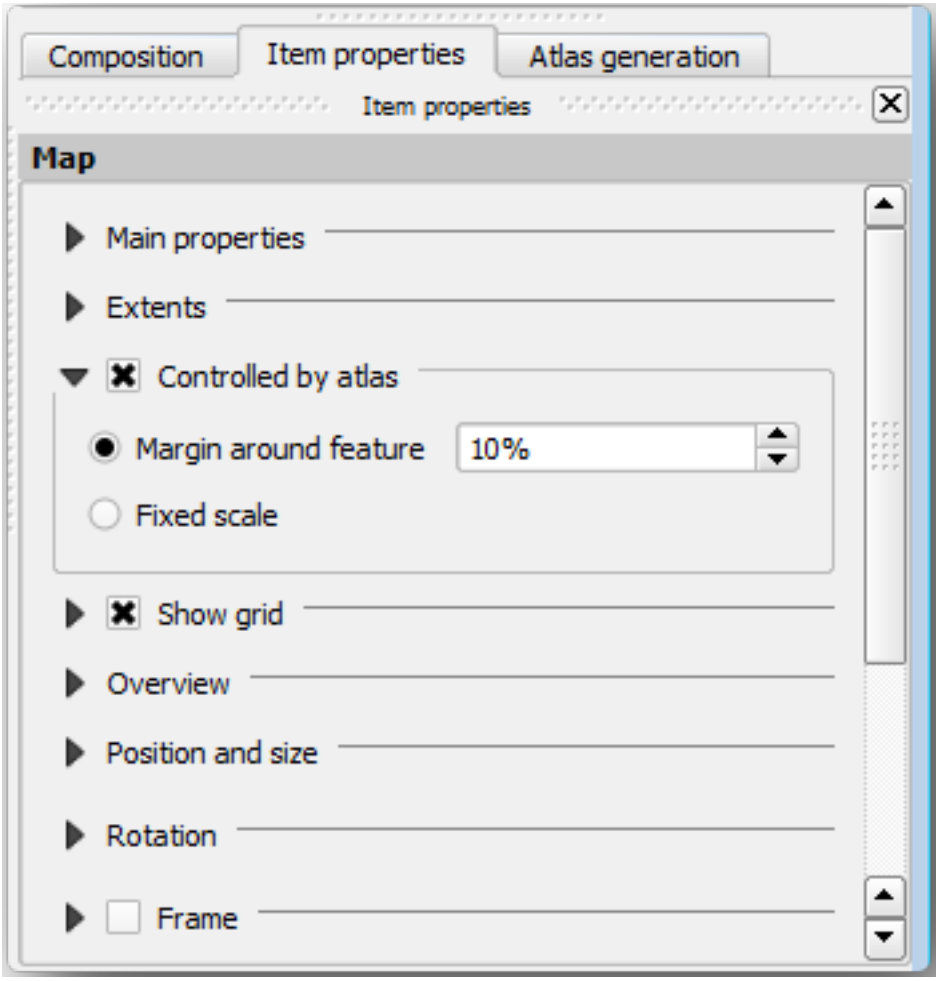
1. Go back to the print layout.
2. In the panel on the right, go to the *Atlas generation* tab.
3. Set the options as follows:



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:

1. Select the bigger map element (`Map 0`).
2. Go to the *Item properties* tab.
3. In the list, check *Controlled by atlas*.
4. 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:

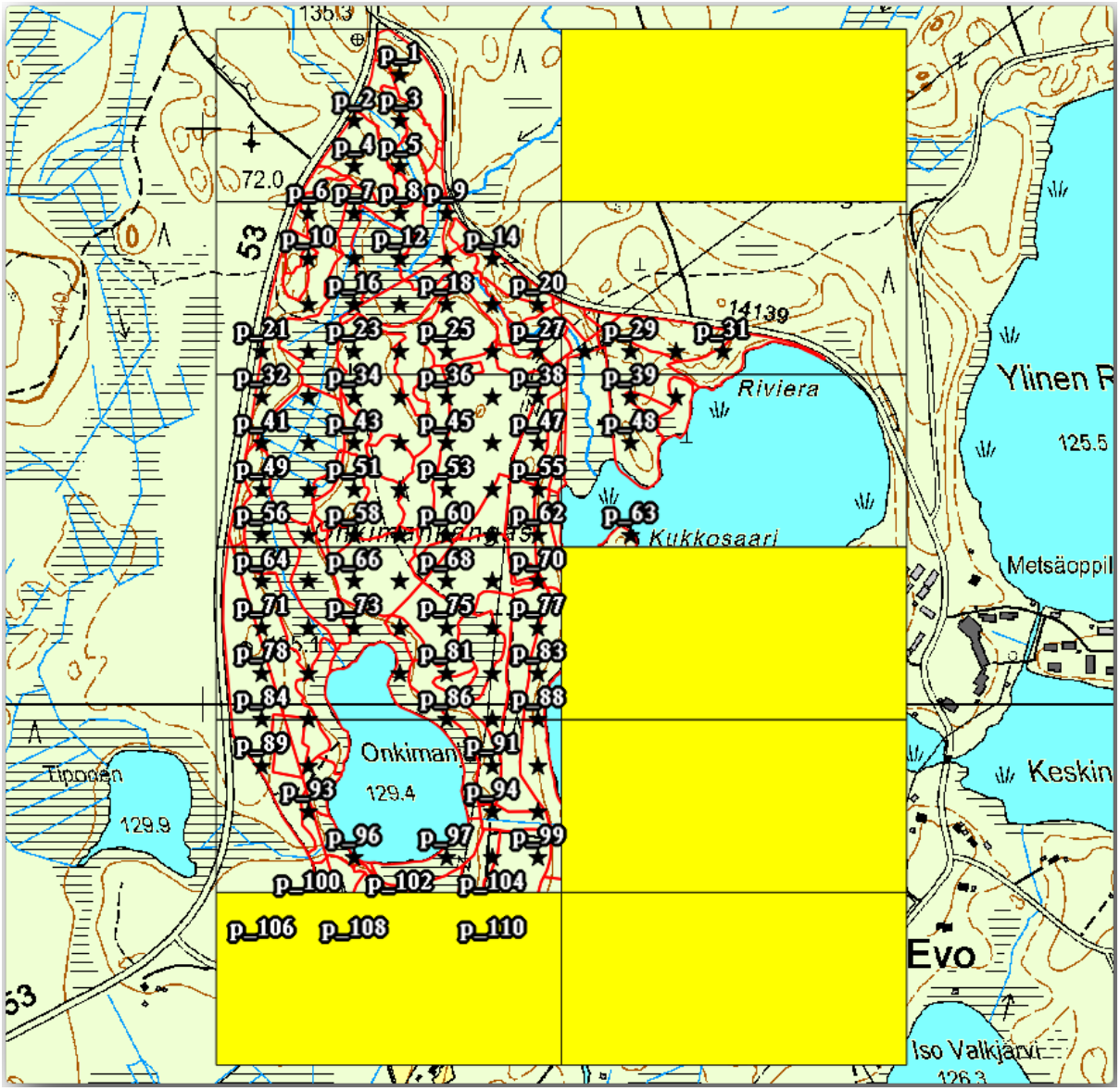
1. Activate the Atlas previews using the button or if your Atlas toolbar is not visible, via *Atlas* ► *Preview Atlas*.
2. You can use the arrows in the Atlas tool bar or in the *Atlas* menu to move through maps that will be created.

Note that some of them cover areas that are not interesting. Let's 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:

1. Go back to the map view.
2. Enable editing for the *atlas_coverage* layer.
3. Select the polygons that are selected (in yellow) in the image below.
4. Remove the selected polygons.
5. 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:

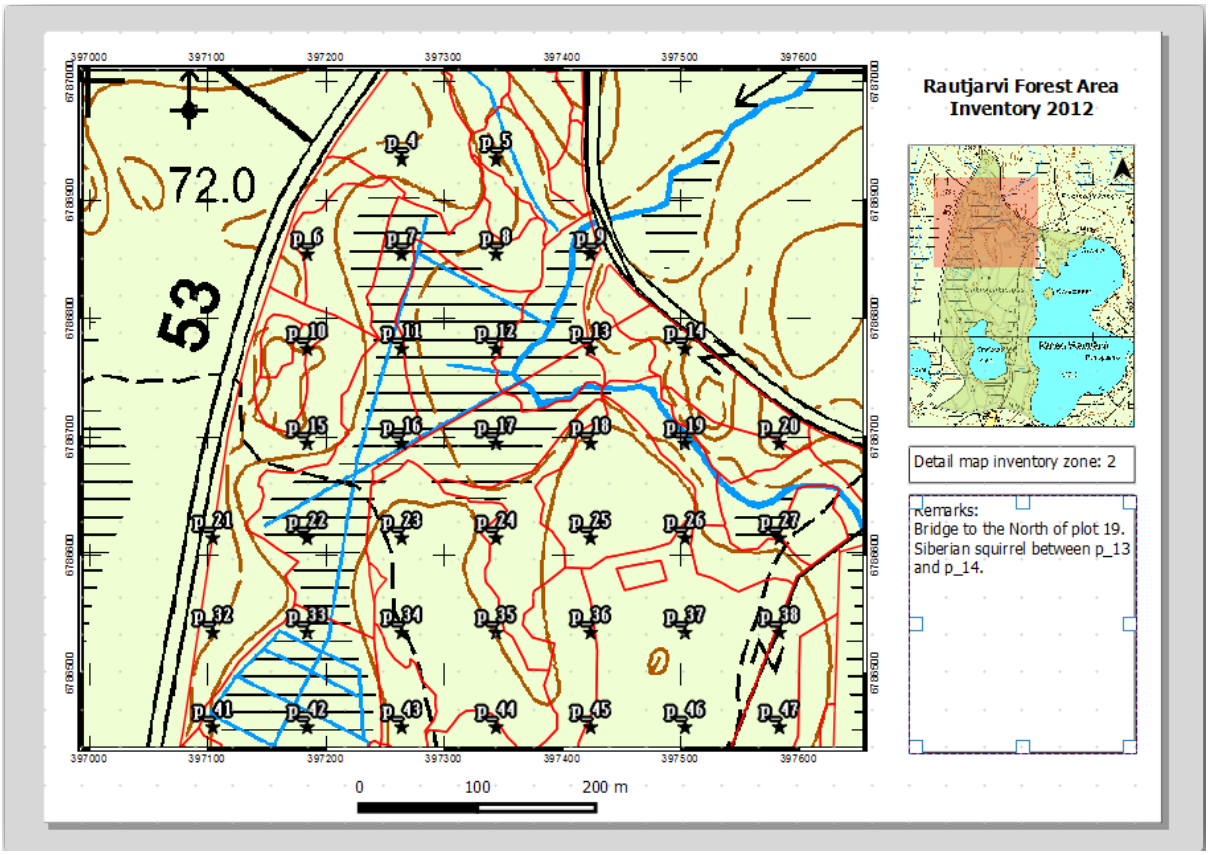
1. Open the *Attribute table* for the *atlas_coverage* layer.
2. Enable editing.
3. Use the calculator to create and populate the following two fields.
4. Create a field named `Zone` of type *Whole number (integer)*.
5. In the *Expression* box write/copy/construct `$rownum`.
6. Create another field named `Remarks`, of type *Text (string)* and a width of `255`.
7. 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.

1. Go back to the *Print Layout*.
2. Select the text label containing `Detailed map....`
3. Set the *Font* size to `12`.
4. Set the cursor at the end of the text in the label.
5. In the *Item properties* tab, inside the *Main properties* click on *Insert or Edit an Expression...*.
6. In the *Function list* double-click on the field *Zone* under *Field and Values*.
7. Click *OK*.
8. The text inside the box in the *Item properties* should show `Detail map inventory zone: [% "Zone" %]`. Note that the `[% "Zone" %]` will be substituted by the value of the field *Zone* for the corresponding feature from the layer *atlas_coverage*.
9. Test the contents of the label by looking at the different Atlas preview maps.
10. Do the same for the labels with the text *Remarks:* using the field with the zone information. You can leave a break line before you enter the expression. You can see the result for the preview of zone 2 in the image below:



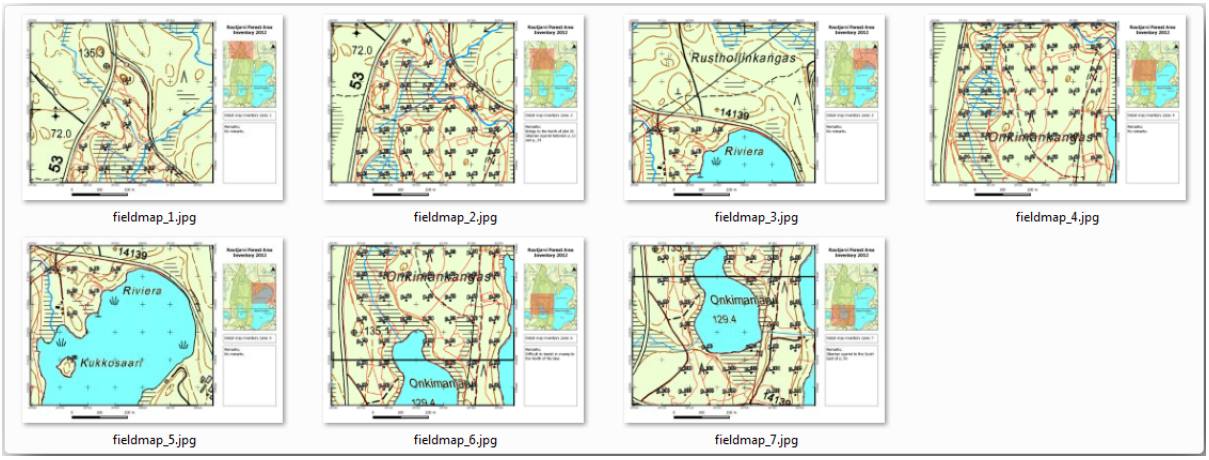
11. Use the Atlas preview to browse through all the maps you will be creating soon and enjoy!

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:

1. Go to the *Atlas generation* tab on the right panel.
2. 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.
3. Open *Layout* ► *Export as PDF…*.
4. Save the PDF file as `inventory_2012_maps.pdf` in your `exercise_data\forestry\samplig\map_creation\` folder.
5. Open the PDF file to check that everything went as expected.
6. 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:



7. In the *Print Layout*, press Save in order to save your print layout changes in the project. This also saves the project file. You can reopen the project at any time and run or edit the atlas.

   It is also possible to 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 in other projects.

8. Close the *Print Layout* and the 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 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 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

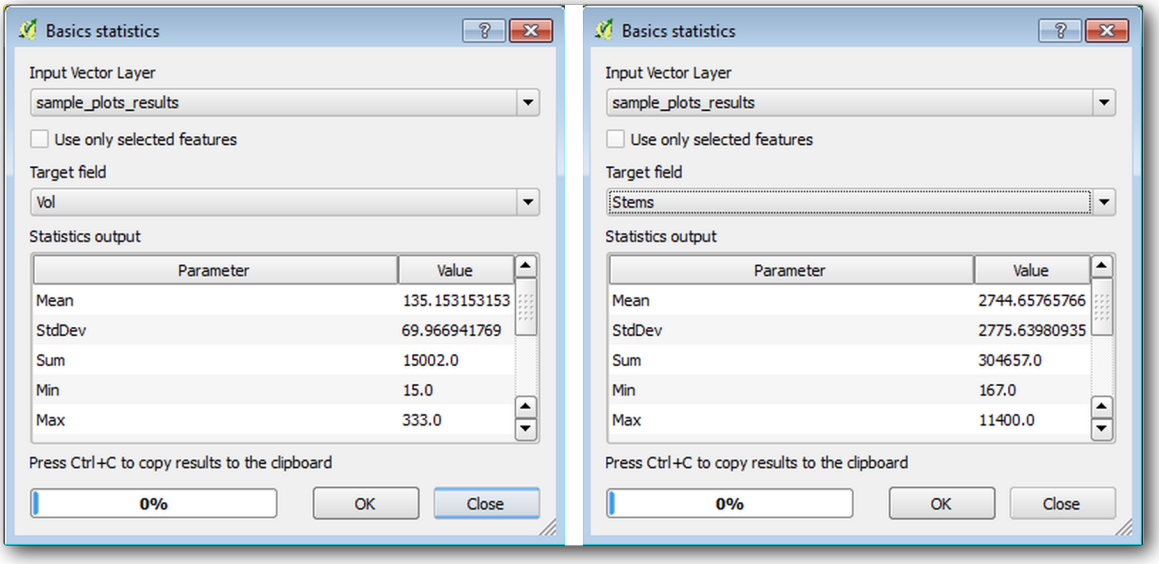
You can calculate the averages for this whole forest area from the inventory results for the some interesting parameters, like the volume and the number of stems per hectare. Since the systematic sample plots represent equal areas, you can directly calculate the averages of the volumes and number of 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

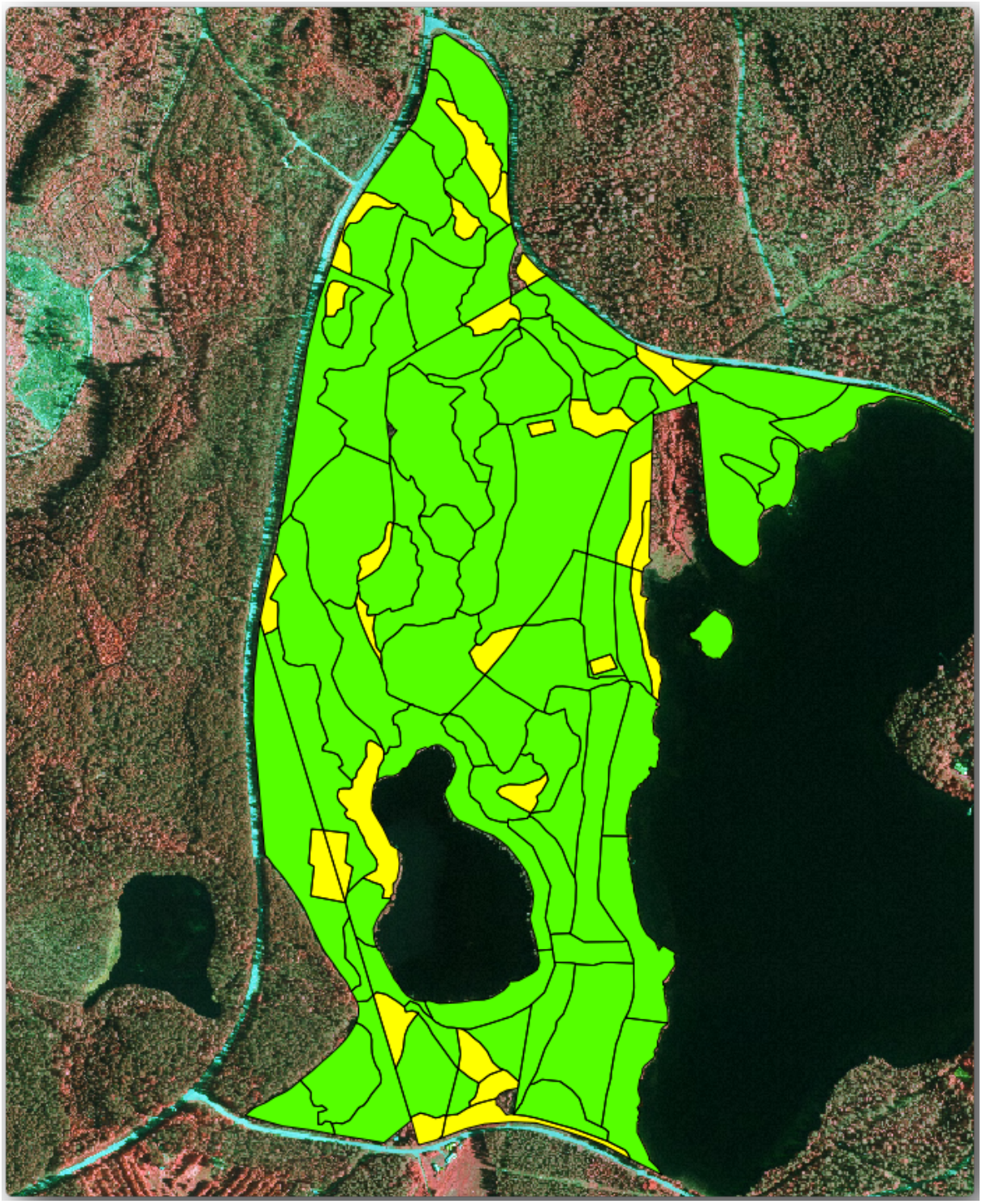
You can make use of those same systematic sample plots to calculate estimates for the different forest stands you digitized previously. Some of the forest stands did not get any sample plot and for those you will not get information. You could have planned some extra sample plots when you planned the 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:



Let's 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*.

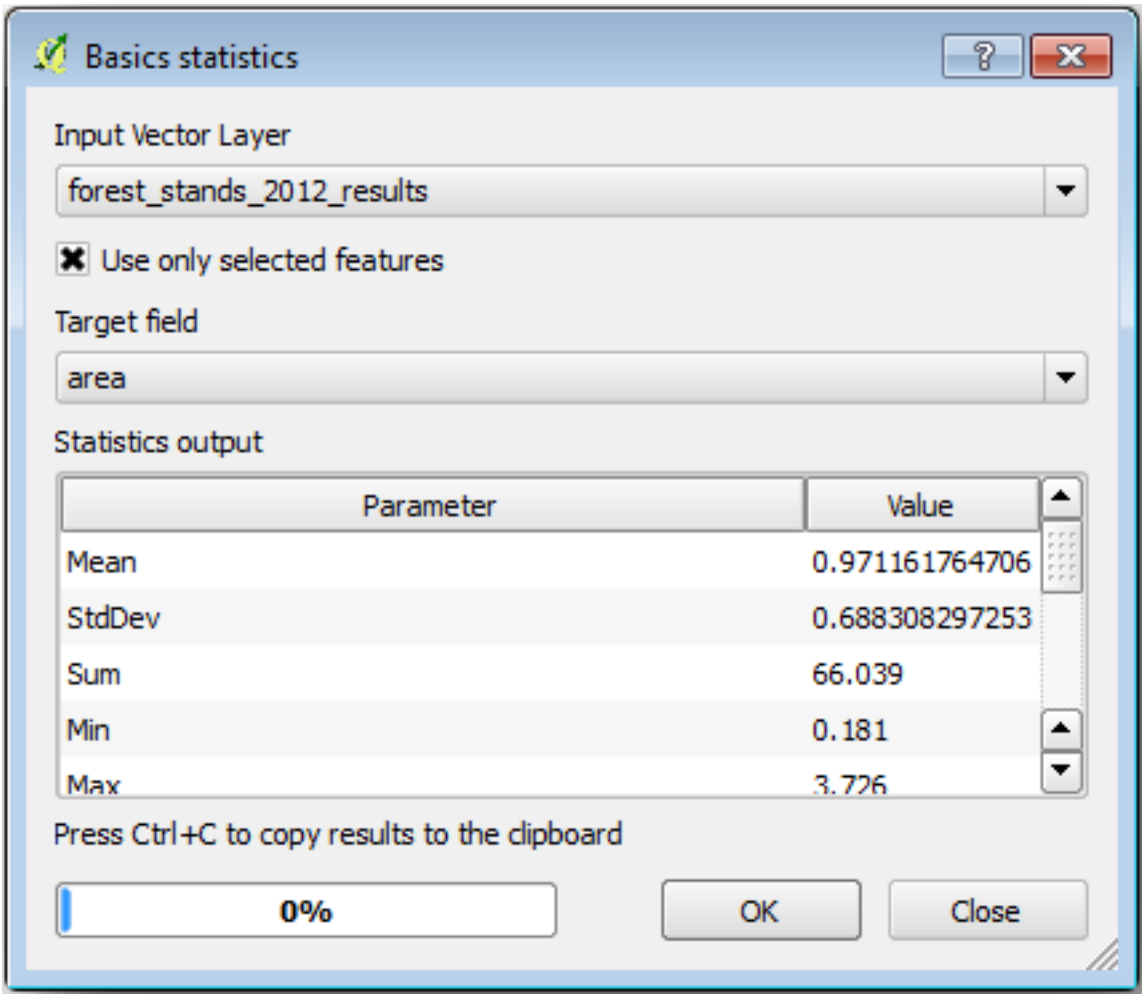
3. Create a new field called `area`.
4. Set the *Output field type* to `Decimal number (real)`.
5. Set the *Precision* to `2`.
6. In the *Expression* box, write `$area / 10000`. This will calculate the area of the forest stands in ha.
7. Click *OK*.

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 zu LiDAR-Daten

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.

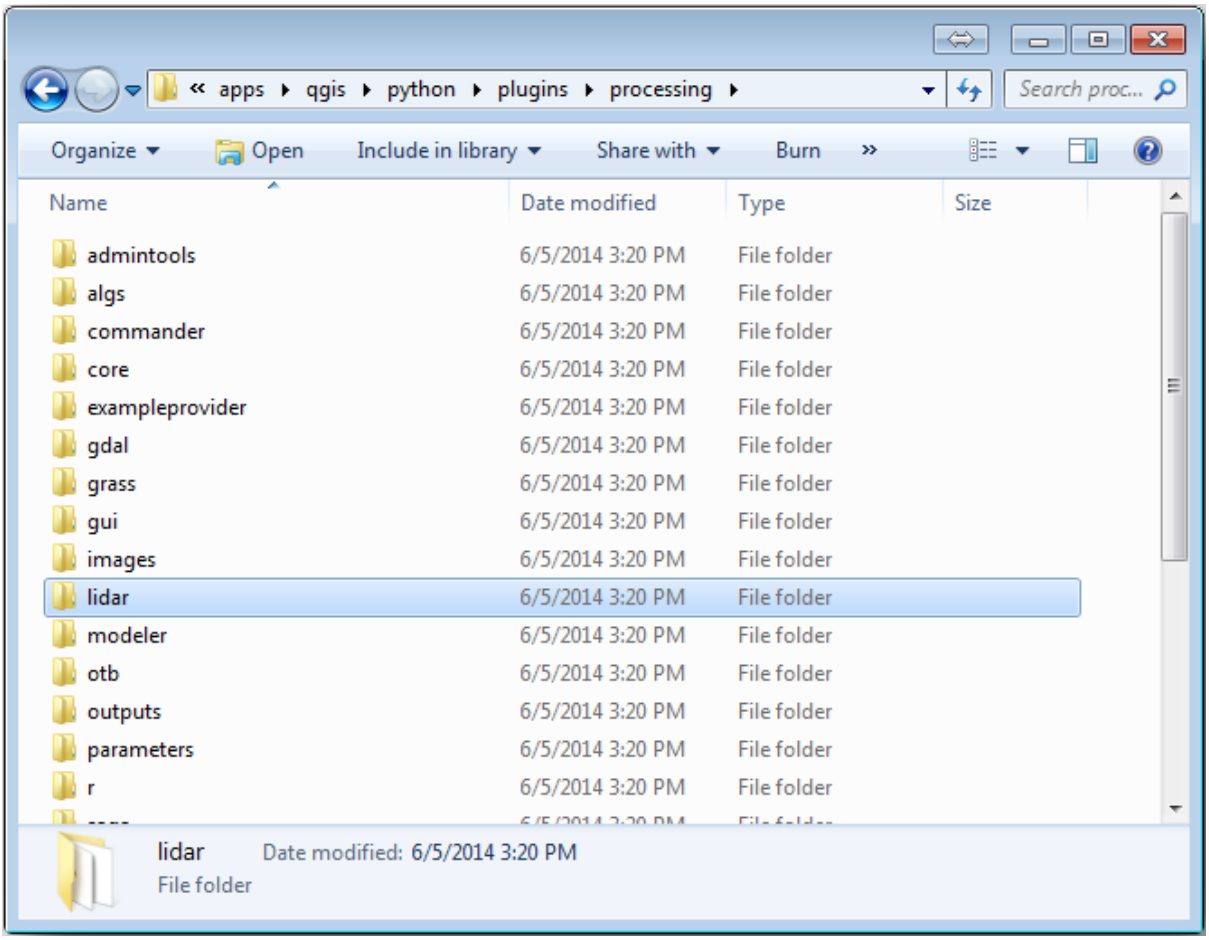
The goal for this lesson: Install LAStools and calculate a DEM from LiDAR data and a hillshade raster.

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.

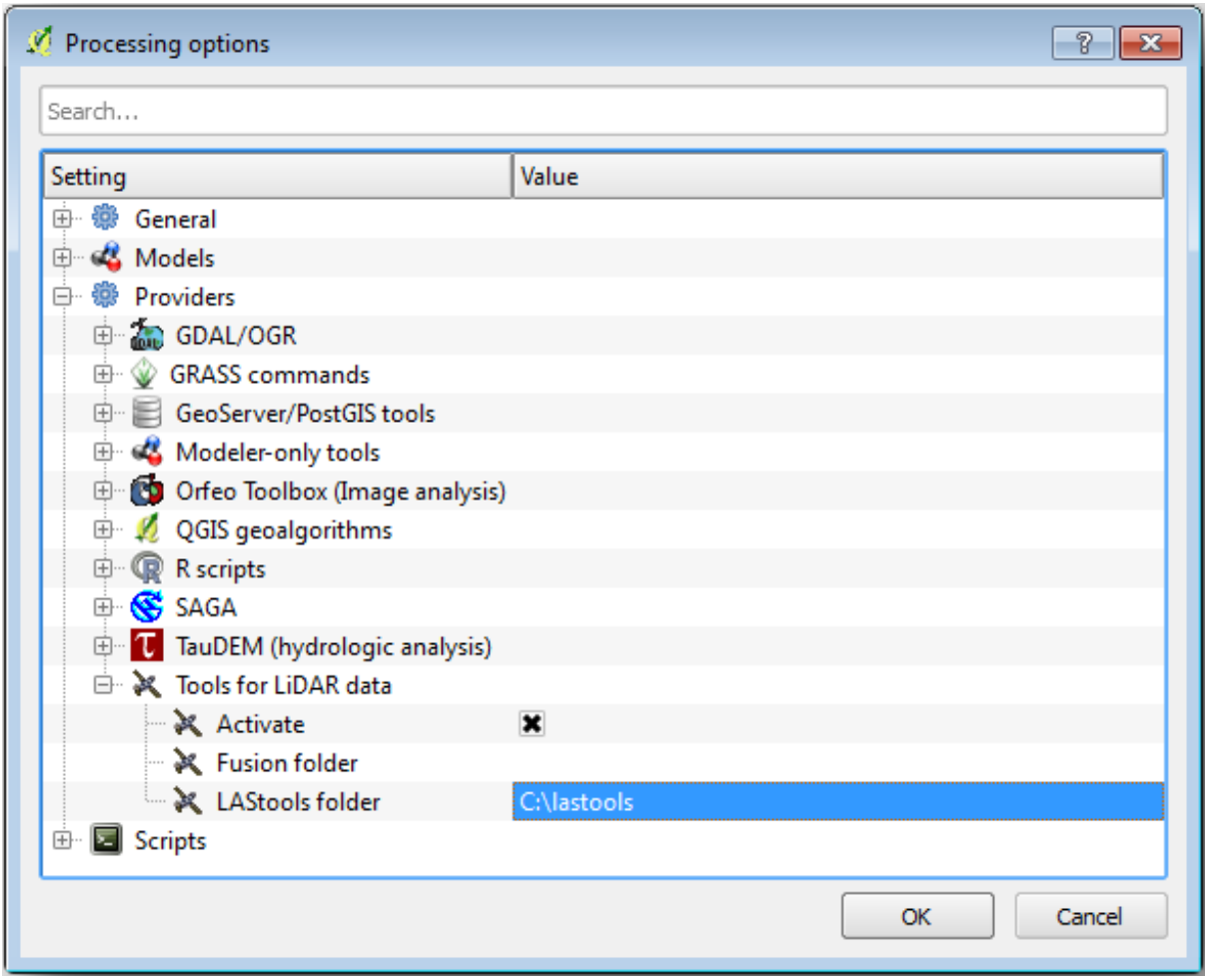
Bemerkung: Read the `LICENSE.txt` file inside the `lastools` folder. Some of the LAStools are open source and other are closed source and require licensing for most commercial and governmental use. For education and evaluation purposes you can use and test LAStools as much as you need to.

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:

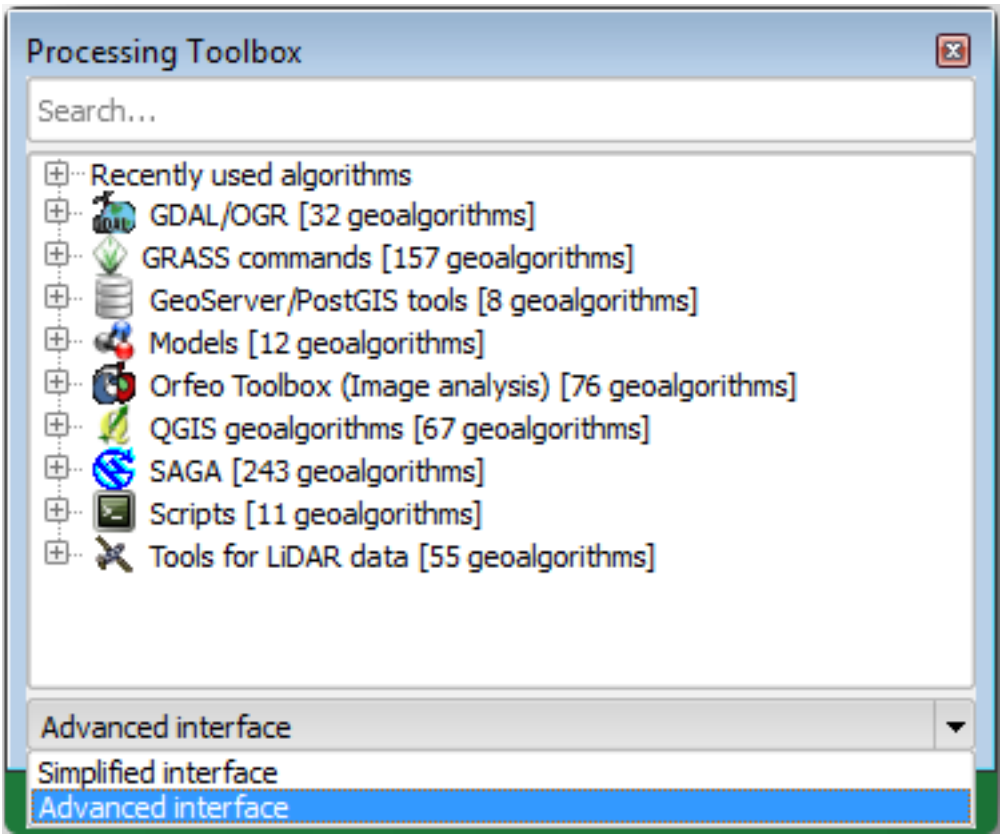
- Go to *Processing* ► *Options and configuration*.
- In the *Processing options* dialog, go to *Providers* and then to *Tools for LiDAR data*.
- Check *Activate*.
- For *LAStools folder* set `c:\lastools\` (or the folder you extracted LAStools to).



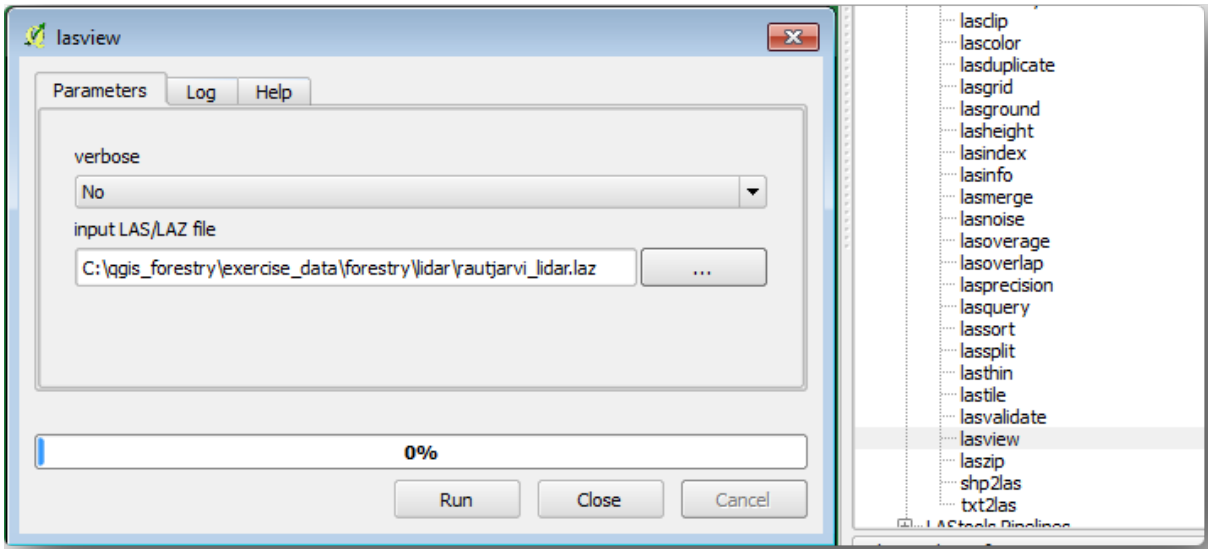
14.8.2 Follow Along: Calculating a DEM with LAStools

You have already used the *Processing* toolbox in *Lesson: Spatial Statistics* 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.

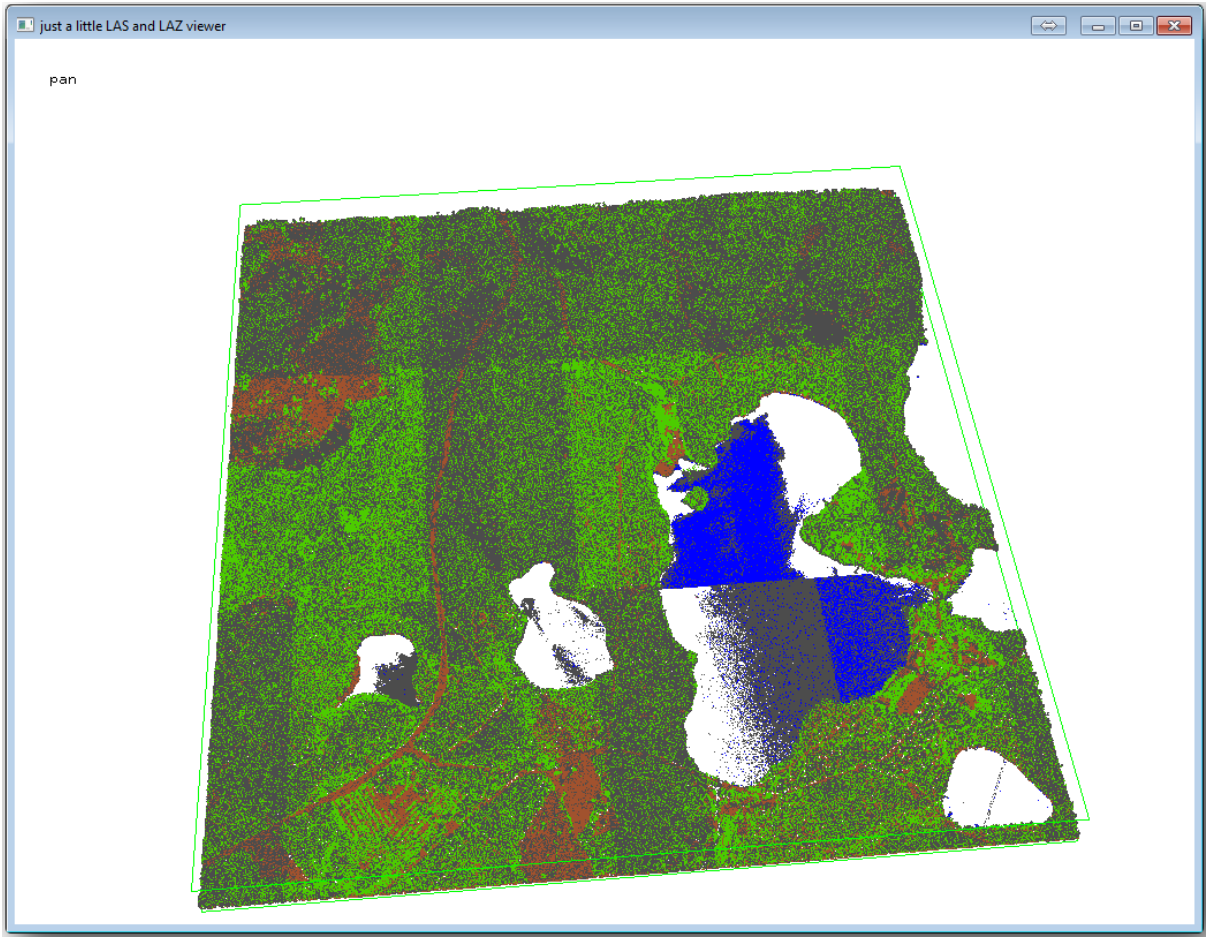


- Expand it to see the tools available, and expand also the *LAStools* category (the number of algorithms may vary).
- Scroll down until you find the *lasview* algorithm, double click it to open.
- At *Input LAS/LAZ file*, browse to `exercise_data\forestry\lidar\` and select the `rautjarvi_lidar.laz` file.



- Klicken Sie auf *Starten*.

Now you can see the LiDAR data in the *just a little LAS and LAZ viewer* dialog window:



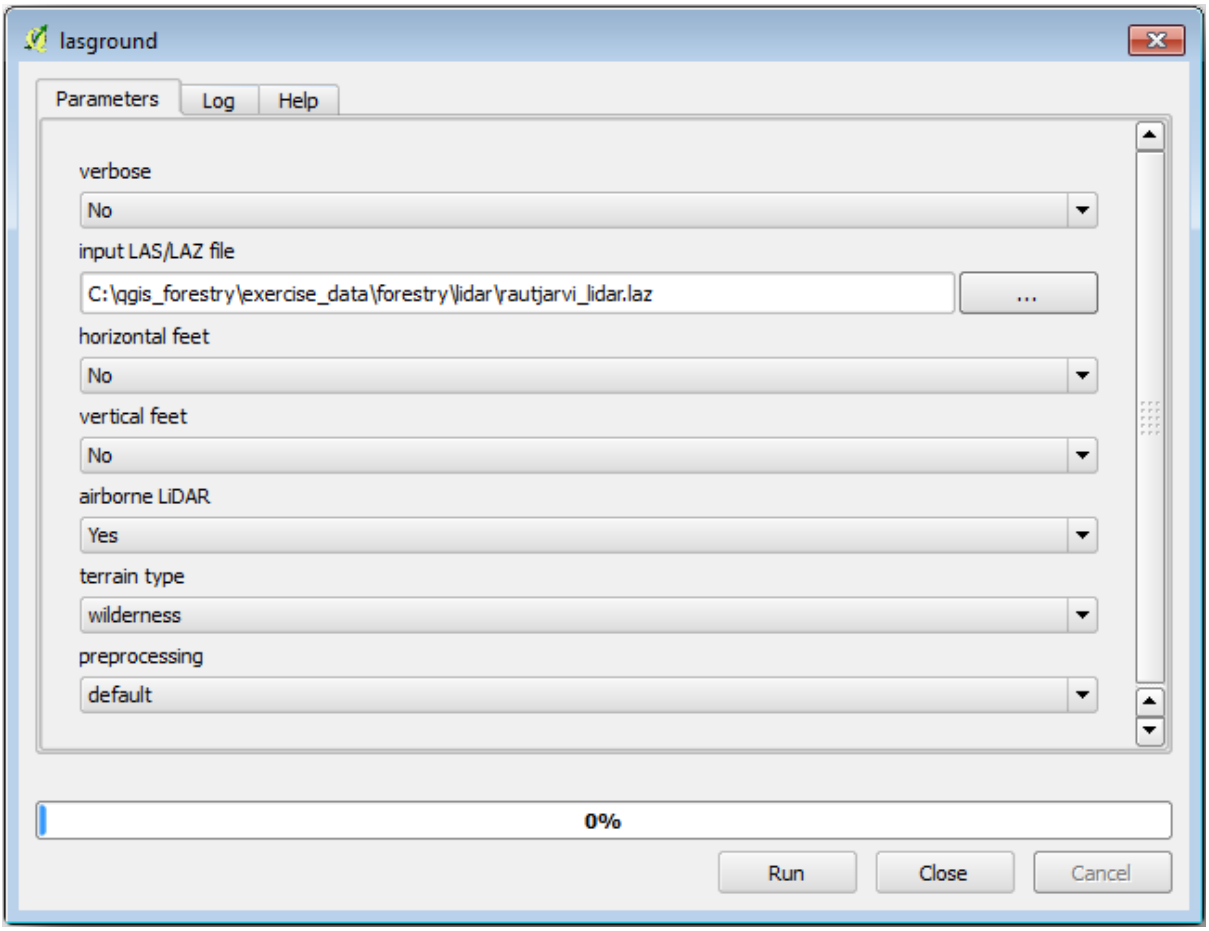
There are many things you can do within this viewer, but for now you can just click and drag on the viewer to pan the LiDAR point cloud to see what it looks like.

Bemerkung: 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.

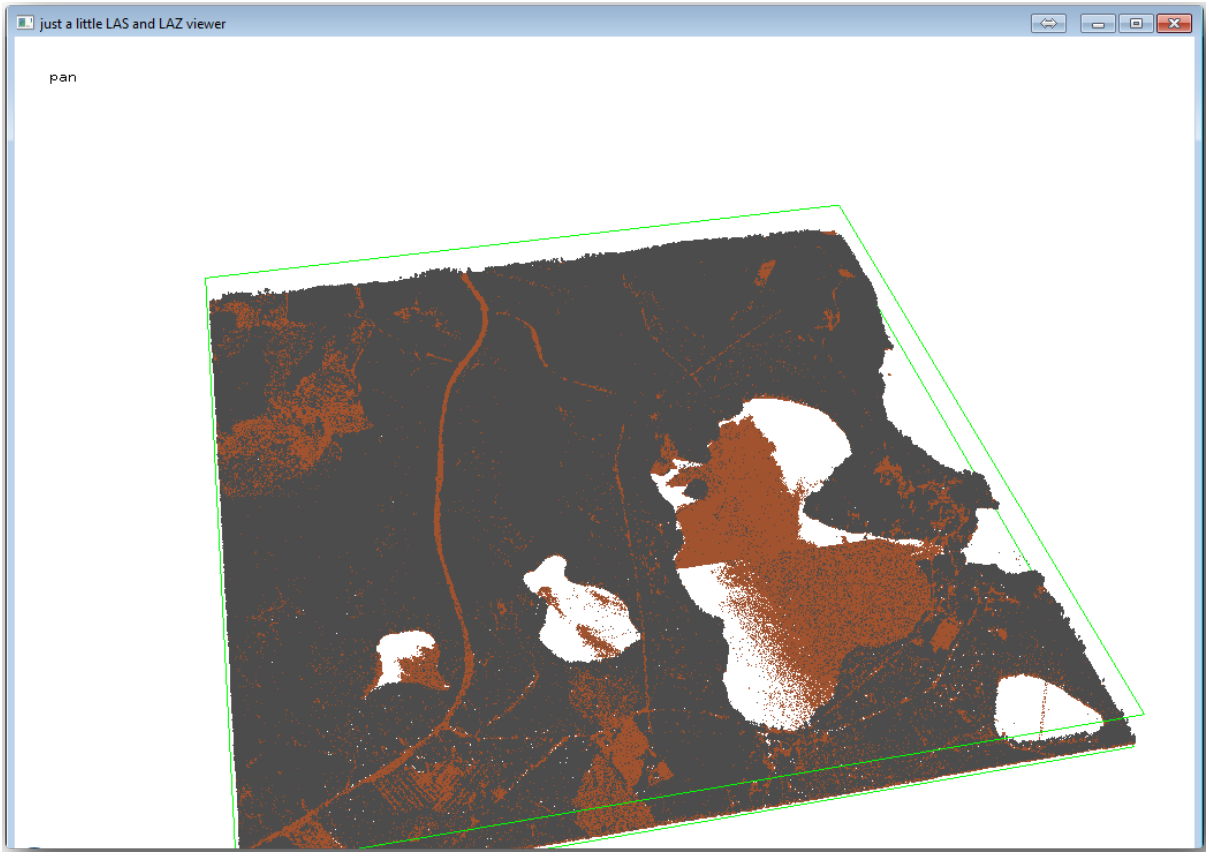
Creating a DEM with LAStools can be done in two steps, first one to classify the point cloud into `ground` and `no ground` points and then calculating a DEM using only the `ground` points.

- 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:



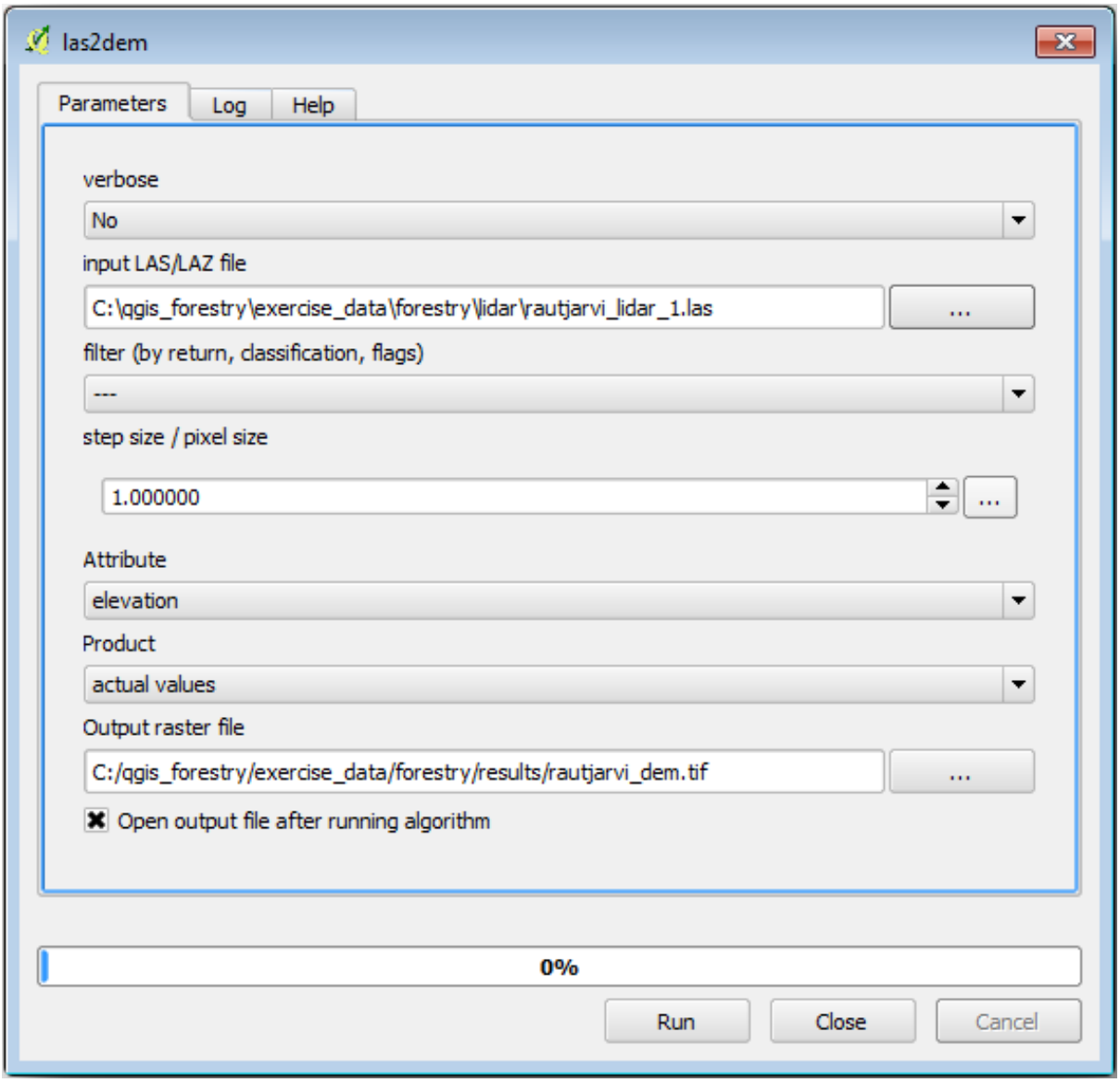
- The output file is saved to the same folder where the `rautjarvi_lidar.laz` is located and it is named `rautjarvi_lidar_1.las`.

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:



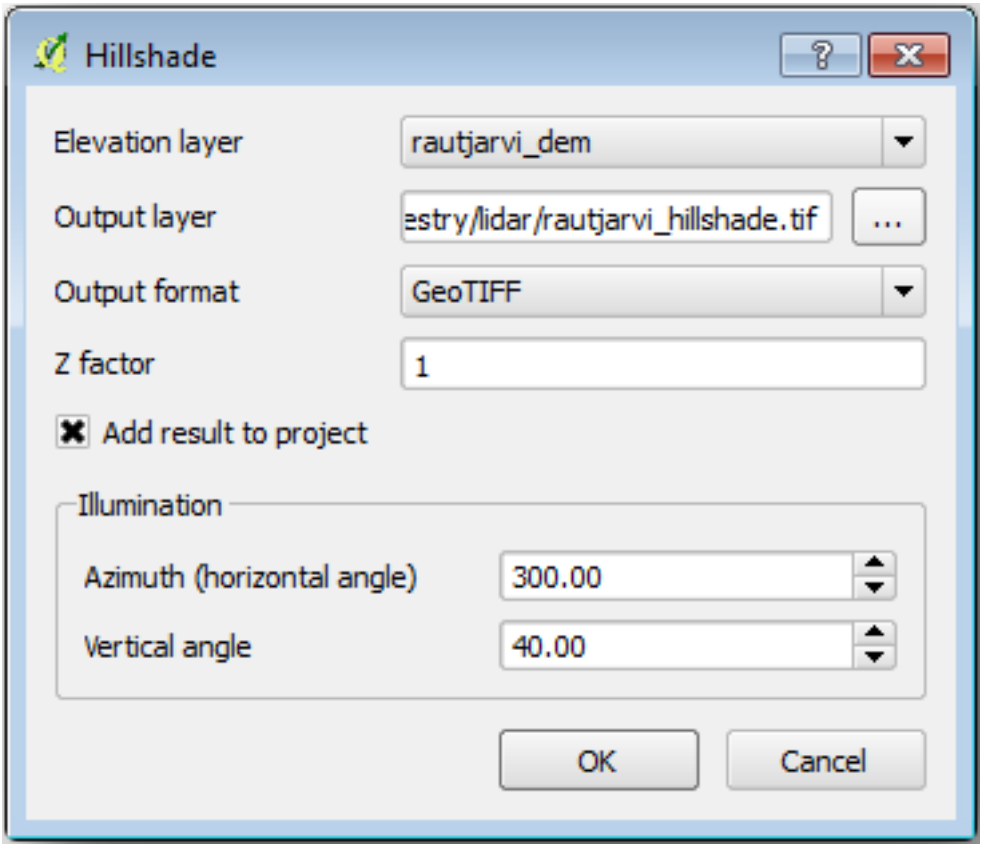
The result DEM is added to your map with the generic name `Output raster file`.

Bemerkung: 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

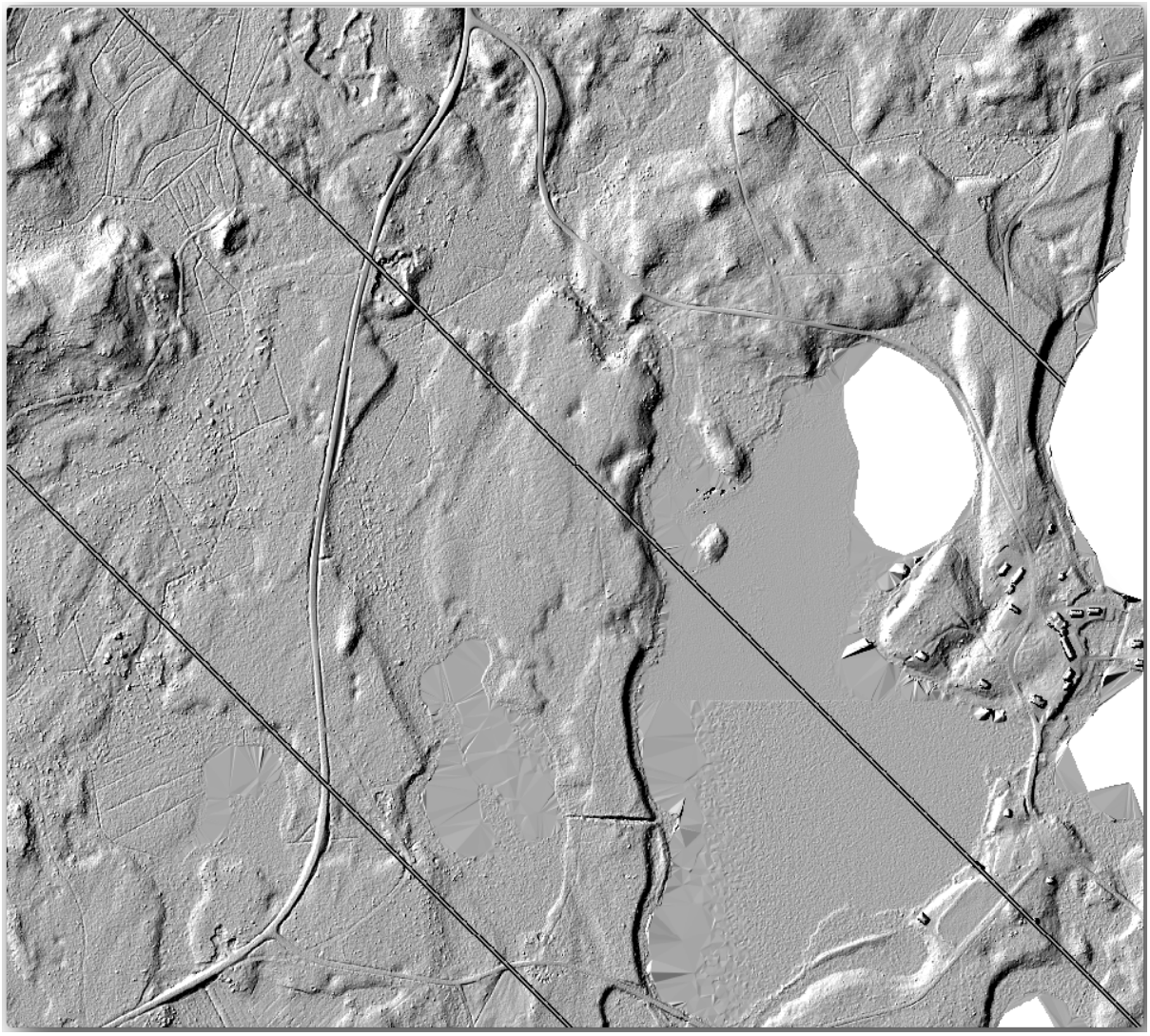
For visualization purposes, a hillshade generated from a DEM gives a better visualization of the terrain:

- 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: Kartendarstellung

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).

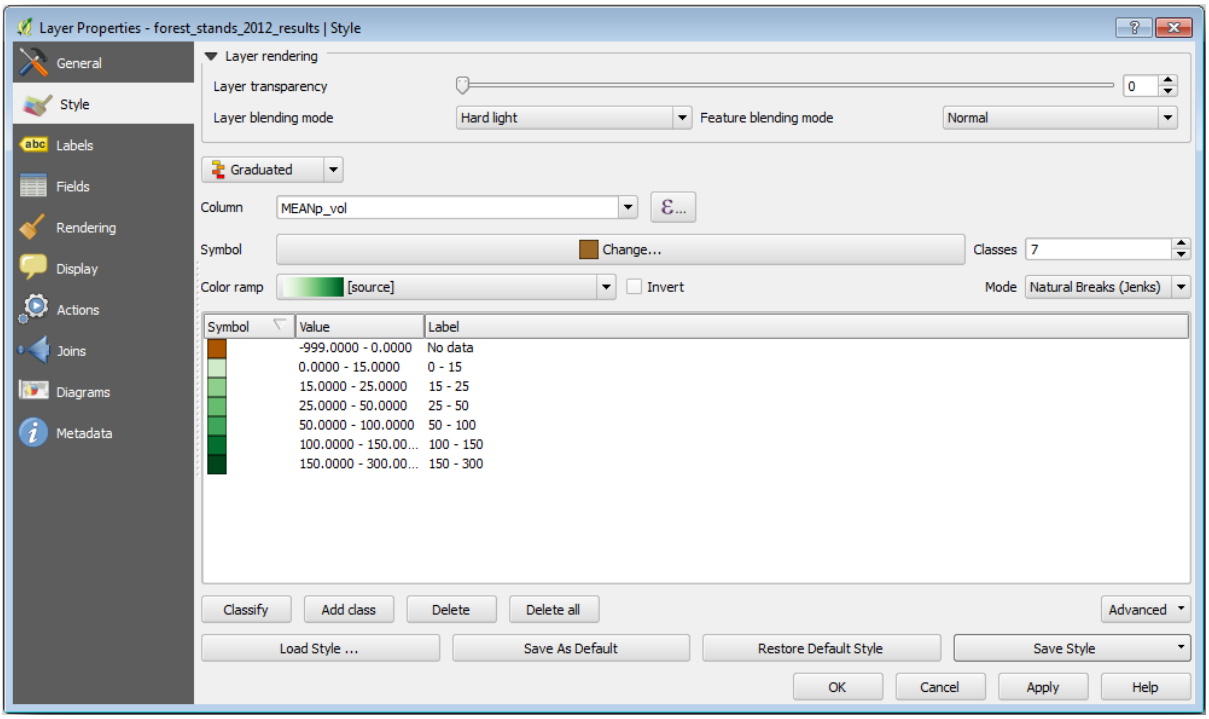
You are going to present the average volumes of your forest stands in a map. If you open the *Attribute table* for the `forest_stands_2012_results` layer, you can see the `NULL` values for the stands without information. To be able to get also those stands into your symbology you should change the `NULL` 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:

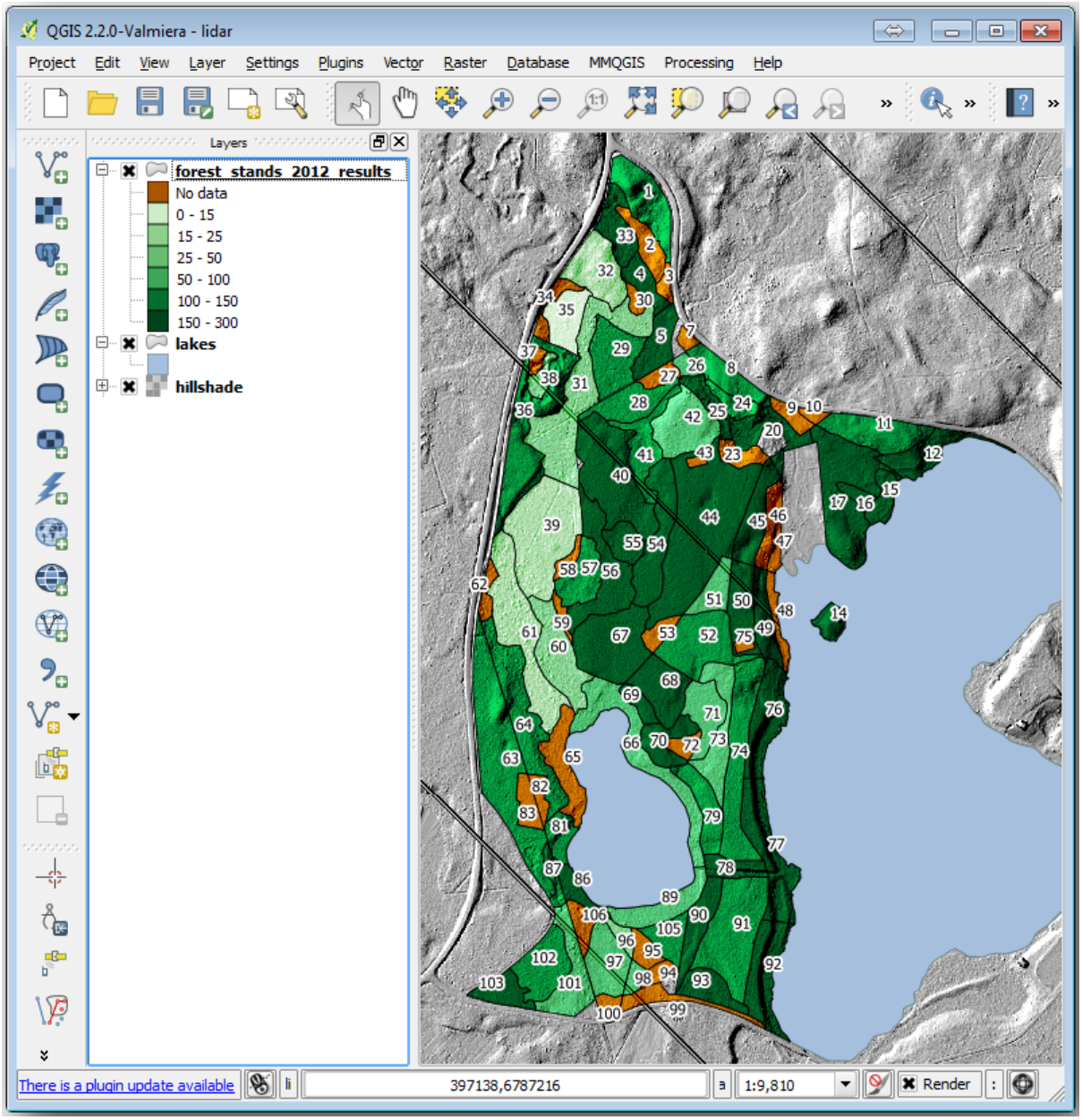
- Open the *Attribute table* and enable editing.
- Select the polygons with `NULL` values.
- Use the calculator to update the values of the `MEANVol` field to `-999` only for the selected features.
- Disable editing and save the changes.

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*.

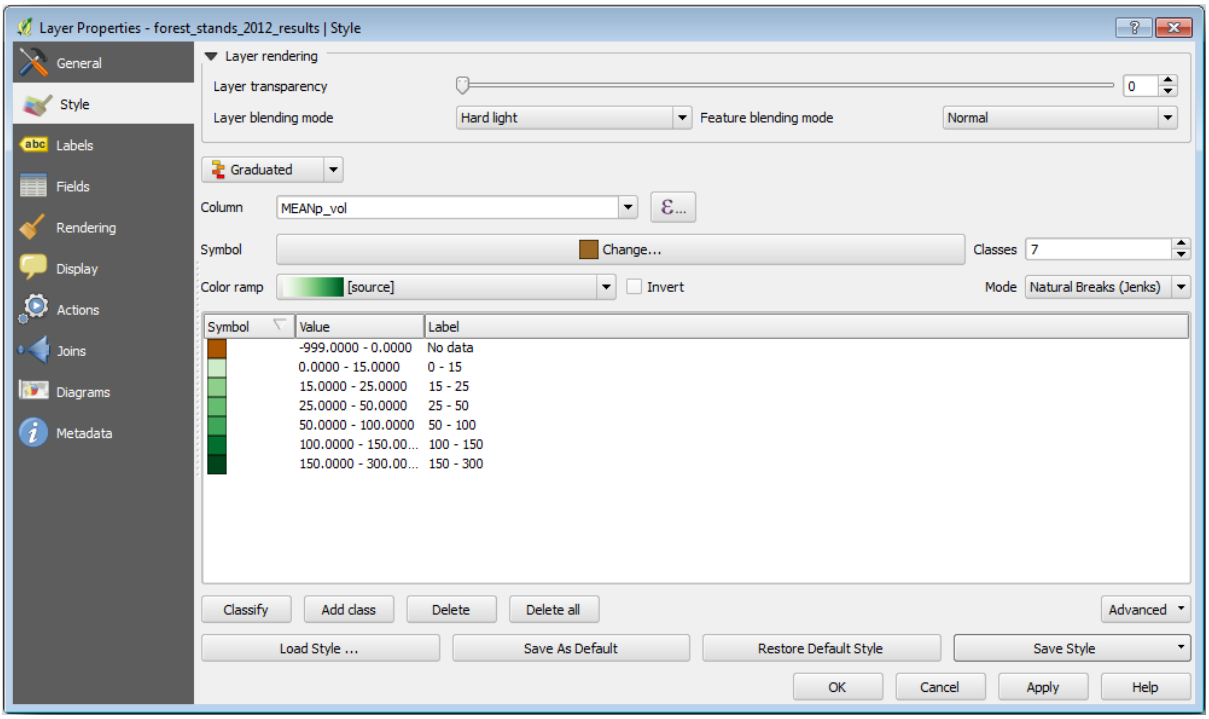


Your map will look something like this:



14.9.2 Try Yourself Try Different Blending Modes

The style you loaded:

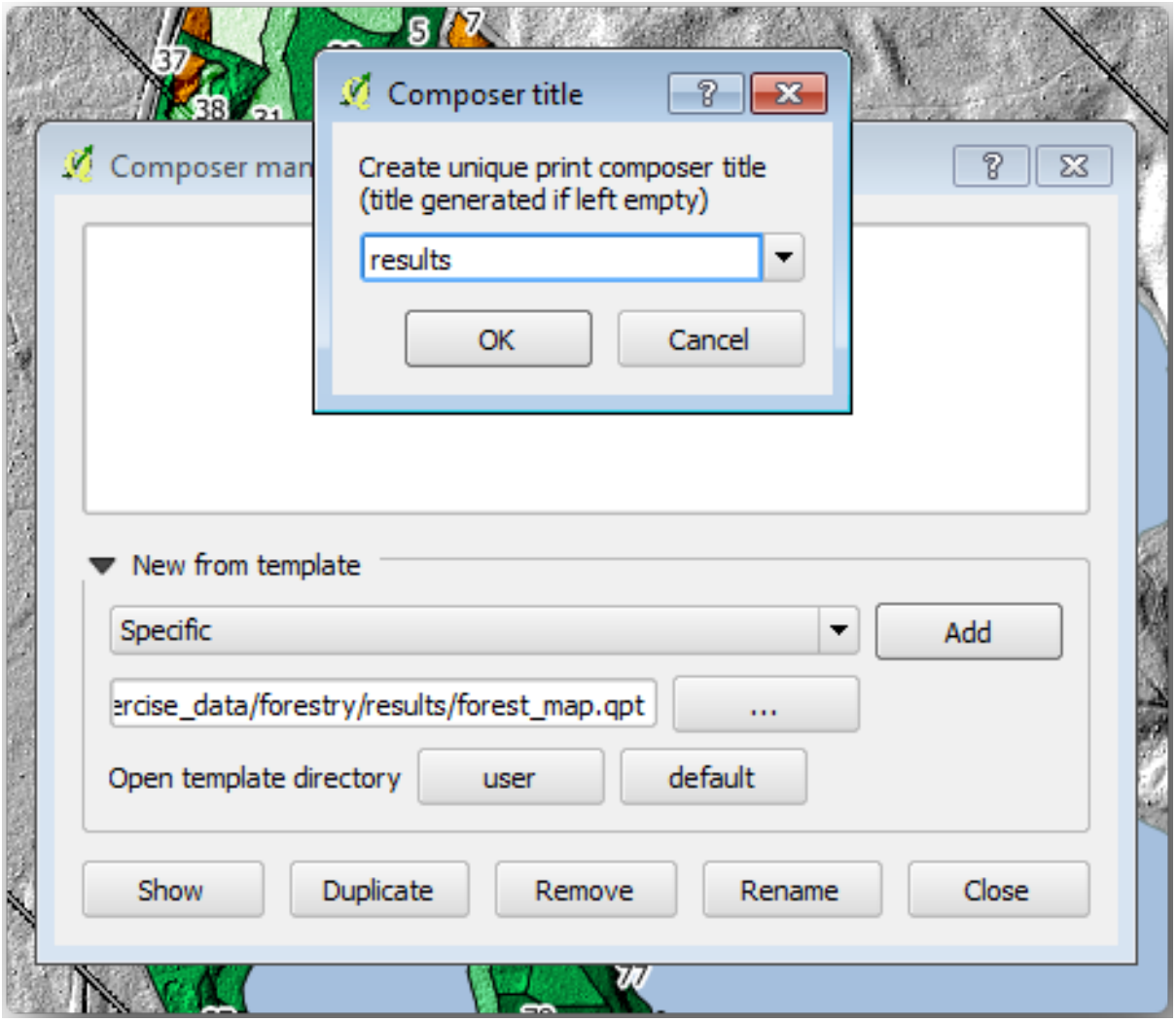


is using the `Hard light` mode for the *Layer blending mode*. Note that the different modes apply different filters combining the underlying and overlying layers, in this case the hillshade raster and your forest stands are used. You can read about these modes in the User Guide.

Try with different modes and see the differences in your map. Then choose the one you like better for your final map.

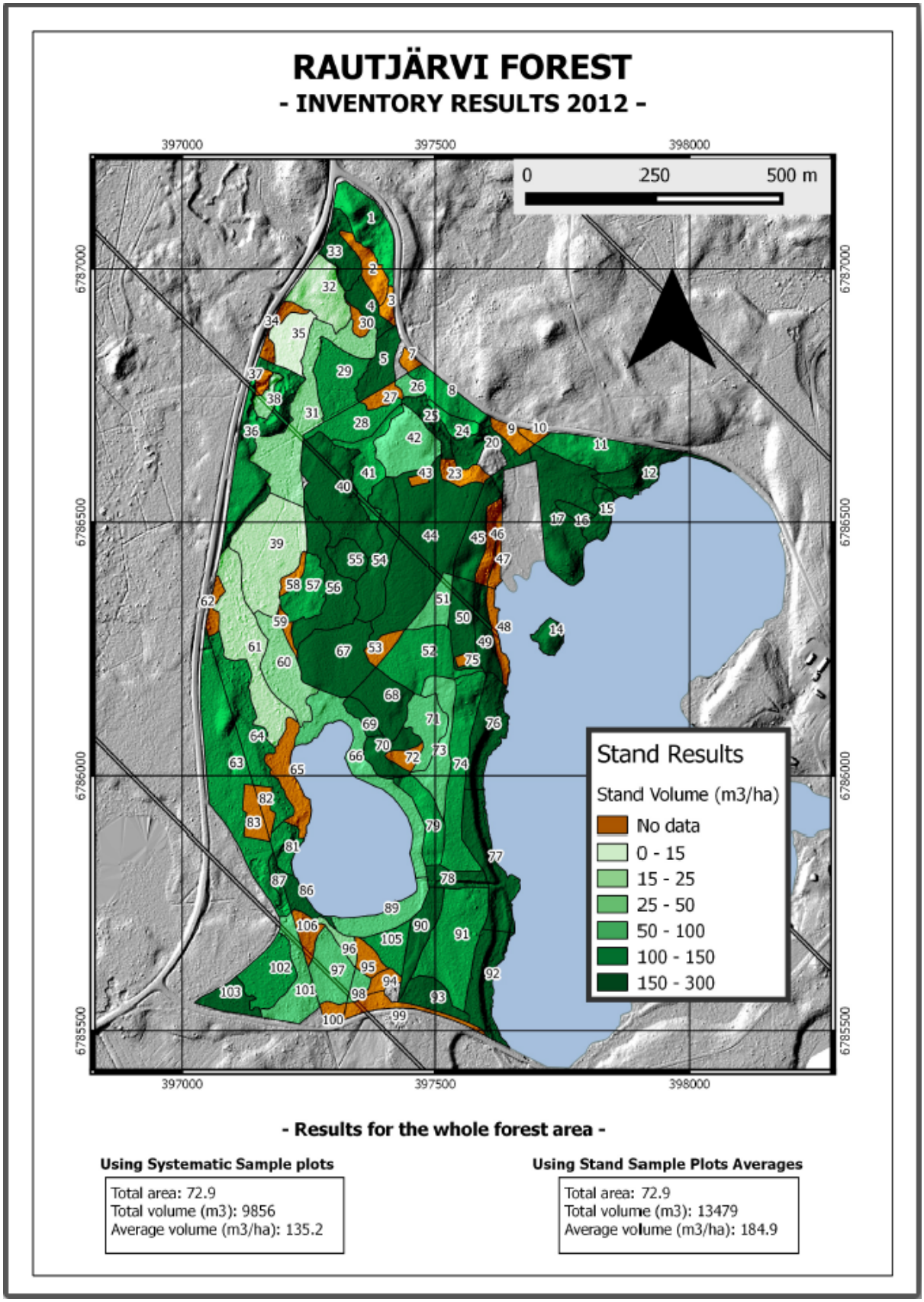
14.9.3 Try Yourself Using a Layout Template to Create the Map result

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.

KAPITEL 15

Module: Datenbank-Konzepte mit PostgreSQL

Relationale Datenbanken sind ein wichtiger Bestandteil eines jeden GIS-Systems. In diesem Modul lernen Sie die Konzepte eines relationalen Datenbankmanagementsystems (RDBMS) kennen und Sie werden PostgreSQL verwenden, um eine neue Datenbank zum Speichern von Daten zu erstellen, sowie andere typische RDBMS-Funktionen kennen lernen.

15.1 Lesson: Einführung in Datenbanken

Bevor wir PostgreSQL verwenden, gehen wir einige Grundlagen der allgemeinen Datenbanktheorie durch. Sie müssen die hier verwendeten Beispielcodes nicht abtippen. Sie dienen nur zur Illustration.

Ziel dieser Übung: Verständnis der grundlegenden Datenbankkonzepte.

15.1.1 Was ist eine Datenbank?

Eine Datenbank besteht aus einer organisierten Sammlung von Daten für einen bestimmten oder mehrere Zwecke in der Regel in digitaler Form. - *Wikipedia*

Ein Datenbankmanagementsystem (DBMS) ist ein Programm, das Datenbanken verwaltet, für die Speicherung von Daten, den Zugriff auf Daten, die Sicherheit, Rettungskopien und weitere Dinge sorgt. - *Wikipedia*

15.1.2 Tabellen

In relationalen Datenbanken und Flat-File-Datenbanken versteht man unter einer Tabelle eine Menge von Datenelementen (Werte), die in vertikalen Spalten (die durch einen Spaltennamen identifiziert sind) und horizontalen Zeilen organisiert sind. Eine Tabelle hat eine bestimmte Anzahl an Spalten und eine unbegrenzte Anzahl an Zeilen. Jede Zeile wird durch die Werte einer Teilmenge der Spalten, die als Schlüsselkandidat gekennzeichnet sind, identifiziert. -*Wikipedia*

```
 id | name  | age
----+-------+-----
  1 | Tim   |  20
  2 | Horst |  88
(2 rows)
```

In SQL-Datenbanken wird eine Tabelle auch als **Relation** bezeichnet.

15.1.3 Spalten / Felder

Eine Spalte ist eine Menge von Datenwerten eines besonders einfachen Typs, ein Wert für jede Zeile der Tabelle. Die Spalten geben die Struktur vor, in der die Zeilen angeordnet sind. Die Bezeichnung Feld wird oftmals abwechselnd mit der Bezeichnung Spalte verwendet. In der Mehrheit wird es jedoch als korrekter angesehen, den Wert der sich an der Überschneidung einer Zeile mit einer Spalte befindet, als Feld (oder Feldwert) zu bezeichnen. -*Wikipedia*

Eine Spalte:

```
| name  |
+-------+
| Tim   |
| Horst |
```

Ein Feld:

```
| Horst |
```

15.1.4 Datensätze

Ein Datensatz ist die in einer Tabellenzeile gespeicherte Information. Jeder Datensatz enthält für jede Spalte der Tabelle ein Feld.

```
2 | Horst |  88  <-- one record
```

15.1.5 Datentypen

Datentypen schränken die Art der Information ein, die in einer Spalte gespeichert werden kann. - *Tim und Horst*

Es viele verschiedene Datentypen. Hier sind die am weitesten verbreiteten:

- `String` - zur Speicherung von freiem Text
- `Integer` - um ganzzahlige Zahlen zu speichern
- `Real` - zur Speicherung reeller Zahlen
- `Date` - um den Geburtstag von Horst zu speichern, so dass niemand ihn vergisst
- `Boolean` - um einfache wahr/falsch Werte zu speichern

Man kann der Datenbank auch erlauben gar nichts in einem Feld zu speichern. Wenn ein Feld keinen Wert enthält, wird der Inhalt des Feldes als **Null Wert** bezeichnet:

```
insert into person (age) values (40);

select * from person;
```

Ergebnis:

```
id | name  | age
---+-------+-----
 1 | Tim   |  20
 2 | Horst |  88
 4 |       |  40  <-- null for name
(3 rows)
```

Es gibt noch viele weitere Datentypen, die man nutzen kann - check the PostgreSQL manual!

15.1.6 Modellierung einer Adressendatenbank

Wir werden ein einfaches Fallbeispiel betrachten, um zu sehen wie eine Datenbank konsturiert ist. Wir werden eine Adressendatenbank erstellen.

Try Yourself

Schreiben Sie die Eigenschaften einer einfachen Adresse auf, die wir in der Datenbank speichern wollen.

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
```

Struktur einer Adresse

Die Bestandteile, die zu einer Adresse gehören entsprechen den Spalten. Die Art der Information, die in jeder Spalte gespeichert ist, ist der Datentyp. Im nächsten Abschnitt werden wir unseren Entwurf der Adressentabelle analysieren und verbessern.

15.1.7 Datenbanktheorie

Der Prozess der Erstellung einer Datenbank beinhaltet die Erstellung eines Modells der Wirklichkeit; man repräsentiert Konzepte aus der Wirklichkeit durch Objekte in der Datenbank.

15.1.8 Normalisierung

Eine der grundsätzlichen Ideen einer Datenbank besteht in der Vermeidung von doppelten Daten / Redundanz. Der Prozess des Entfernens von Redundanz aus einer Datenbank wird Normalisierung genannt.

Normalisierung ist ein systematischer Weg, um sicher zu stellen, dass die Datenbankstruktur für allgemeine Abfragen passend und frei von bestimmten ungewollten Eigenschaften - Unregelmäßigkeiten beim einfügen, aktualisieren und löschen von Daten - die zum Verlust der Datenintegrität führen können. - *Wikipedia*

Es gibt verschiedene Arten der Normalisierung.

Schauen wir uns ein einfaches Beispiel an:

```
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)
```

Stellen Sie sich vor, Sie haben viele Freunde die in einer Straße mit demselben Namen oder derselben Stadt leben. Bei jeder Duplizierung der Daten wird Speicherplatz verbraucht. Noch schlimmer ist, dass wenn sich der Name einer Stadt ändert, viele Einträge in der Datenbank korrigiert werden müssen.

15.1.9 Try Yourself

Verändern sie die obige *people* Tabelle, um Dopplungen zu verringern und die Datenstruktur zu normalisieren.

Unter dem folgendem Link erfährt man mehr über die Datenbanknormalisierung: 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 Indexe

Ein Datenbankindex ist eine Datenstruktur, die die Geschwindigkeit von Abfrageoperationen auf einer Datenbanktabelle erhöht. - *Wikipedia*

Stellen Sie sich vor, Sie lesen ein Fachbuch und suchen nach der Erklärung eines Begriffes, aber das Buch hat keinen Index! Sie müssten beginnend mit dem ersten Kapitel das ganze Buch durcharbeiten, bis Sie die gesuchte Information finden. Der Index im Buch hilft, schnell zu der Seite im Buch mit der relevanten Information zu springen:

```
create index person_name_idx on people (name);
```

Jetzt ist die Suche nach Namen schneller:

```
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 Sequenzen

Eine Sequenz ist eine Folge eindeutiger Zahlen. Sie wird normalerweise als eindeutiger Identifikator in einer Spalte einer Tabelle verwendet.

In diesem Beispiel ist id eine Sequenz - die Zahl wird mit jeder Aufnahme eines neuen Datensatzes in die Tabelle erhöht:

```
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 Diagramme

In einer normalisierten Datenbank hat man normalerweise viele Relationen (Tabellen). Das Entity-Relationship-Diagramm (ER-Diagramm) wird verwendet, um die logischen Beziehungen zwischen den Relationen zu entwerfen. Denken Sie an unsere nicht normalisierte Tabelle *people* am Anfang der Lektion:

```
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)
```

Mit etwas Arbeit können wir sie in zwei Tabellen teilen. Der Straßenname von Personen die in derselben Straße wohnen, muss dadurch nicht wiederholt werden:

```
select * from streets;

 id |     name
----+--------------
 1  | Plein Street
(1 row)
```

und:

```
select * from people;

 id |     name      | house_no | street_id |  phone_no
----+---------------+----------+-----------+-------------
  1 | Horst Duster |        4 |         1 | 072 121 122
(1 row)
```

Wir können nun die beiden Tabellen mit Hilfe der ‚Schlüssel' `streets.id` und `people.streets_id` verbinden.

Wenn wir ein ER-Diagramm für diese beiden Tabellen entwerfen, sieht es in etwa so aus:

Das ER-Diagramm hilft uns ‚eins zu viele' Beziehungen auszudrücken. In diesem Fall zeigt das Pfeilsymbol, dass in einer Straße viele Personen leben können.

Try Yourself

Unser Modell *people* hat immer noch einige Probleme bezüglich der Normalisierung - versuchen Sie es weiter mit Hilfe eines ER-Diagramms zu normalisieren.

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.

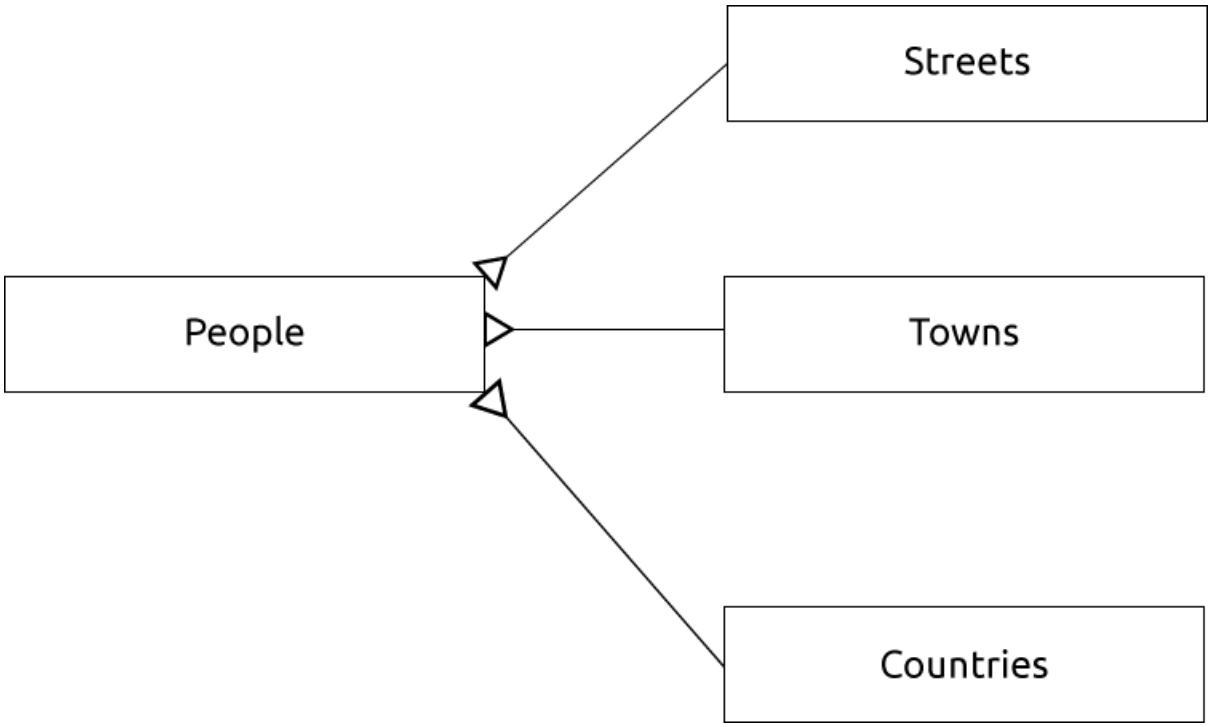
One way to further normalise the table is to split the name field into *first_name* and *last_name*:

```
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 Einschränkungen, Primärschlüssel und Fremdschlüssel

Eine Datenbankeinschränkung wird verwendet, um sicherzustellen, dass die Daten in einer Relation so gespeichert werden, wie es der Modellierer vorgesehen hat. Zum Beispiel kann eine Einschränkung der Postleitzahl darin bestehen, dass die Zahl zwischen `1000` und `9999` liegt.

Ein Primärschlüssel besteht aus einem oder mehreren Feldwerten, die einen Datensatz eindeutig machen. Oftmals wird der Primärschlüssel id genannt und ist eindeutige Zahlenfolge.

Ein Fremdschlüssel ist eine Referenz zu einem eindeutigen Datensatz in einer anderen Tabelle (Nutzung des Primärschlüssels dieser Tabelle).

In ER-Diagrammen basiert die Verbindung zwischen Tabellen normalerweise auf Fremschlüsseln die auf Primärschlüssel verweisen.

In unserem Beispiel, zeigt die Definition der Tabelle, dass die Spalte Straße ein Fremdschlüssel ist. Er verweist auf den Primärschlüssel in der Tabelle Straßen:

```
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 Transaktionen

Wenn beim hinzufügen, ändern oder löschen von Daten in einer Datenbank etwas schief geht, sollte die Datenbank immer in einem konsisten Zustand bleiben. Die meisten Datenbanken enthalten eine sogenannte Transaktionsunterstützung. Transaktionen erlauben es einen Rollback zu erstellen, d.h. einen Zustand vor Ausführung der Transaktion in den bei Problemen zurück gekehrt werden kann.

Ein Beispiel wäre ein Abrechnungssystem in dem man Mittel von einem Konto auf ein anderes Konto transferieren möchte. Es ergibt sich folgende Schrittfolge:

- entferne R20 von Joe
- Füge R20 bei Anne hinzu

Wenn etwas während des Prozesses schief geht (z.B. Stromausfall), wird ein Rollback der Transaktion ausgeführt.

15.1.15 In Conclusion

Datenbanken erlauben es, Daten in einer strukturierten Weise mit Hilfe von einfachem Programierkode zu managen.

15.1.16 What's Next?

Nachdem wir gesehen haben, wie Datenbanken in der Theorie arbeiten, wollen wir nun eine neue Datenbank erstellen und die behandelte Theorie umsetzen.

15.2 Lesson: Implementierung des Datenmodells

Nachdem wir die ganze Theorie behandelt haben, wollen wir nun eine neue Datenbank erstellen. Diese Datenbank wird für unsere Übungen in den folgenden Lektionen verwendet.

Das Ziel dieser Lektion: Die Installation der benötigten Software und deren Verwendung, um unsere Beispieldatenbank zu erstellen.

15.2.1 Installation von PostgreSQL

Bemerkung: 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.

Unter Ubuntu:

```
sudo apt install postgresql-9.1
```

Sie sollten folgende Nachricht erhalten:

```
[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
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
0 upgraded, 4 newly installed, 0 to remove and 5 not upgraded.
Need to get 5,012kB of archives.
After this operation, 19.0MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Drücken Sie `Y` und `Enter` und warten bis der Download und die Installation abgeschlossen sind.

15.2.2 Hilfe

PostgreSQL hat eine sehr gute online Dokumentation.

15.2.3 Erstellen eines Datenbankbenutzers

Unter Ubuntu:

Starten Sie nach dem Abschluss der Installation das folgende Kommando, um Postgres Nutzer zu werden. Erstellen Sie dann einen neuen Datenbanknutzer:

```
sudo su - postgres
```

Geben Sie bei der Abfrage Ihr normales login Passwort ein (Sie benötigen dazu sudo Rechte).

Erstellen Sie nun in der Postgres bash Eingabeaufforderung den Datenbanknutzer. Stellen Sie sicher, dass der Benutzername Ihrem Unix login Namen entspricht. Die Vereinfachung besteht darin, dass Postgres Sie automatisch authentifiziert, wenn Sie als dieser Nutzer eingeloggt sind:

```
createuser -d -E -i -l -P -r -s qgis
```

Geben Sie ein Passwort ein, wenn Sie danach gefragt werden. Sie sollten ein anderes Passwort als Ihr login Passwort verwenden.

Was bedeuten diese Optionen?

```
-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
```

Jetzt können Sie die Postgres Nutzer bash shell Umgebung durch folgende Eingabe verlassen:

```
exit
```

15.2.4 Überprüfen Sie das neue Konto:

```
psql -l
```

Die Ausgabe sollte ähnlich dem folgendem aussehen:

```
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)
```

Drücken sie zum Verlassen Q.

15.2.5 Eine Datenbank erstellen

Das Kommando createdb wird zum Erstellen einer neuen Datenbank verwendet. Es sollte von der Eingabeaufforderung der bash shell aus gestartet werden:

```
createdb address -O qgis
```

Sie können sich durch folgendes Kommando vergewissern, dass Ihre neue Datenbank erstellt wurde:

```
psql -l
```

Die Ausgabe sollte in etwa so aussehen:

```
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)
```

Drücken sie zum Verlassen Q.

15.2.6 Start einer Datenbank shell Sitzung

Sie können sich ganz einfach mit Ihrer Datenbank verbinden:

```
psql address
```

Geben Sie zum Verlassen der psql Datenbank shell folgendes ein:

```
\q
```

Hilfe zur shell erhalten Sie durch Eingabe von:

```
\?
```

Hilfe zur Benutzung von SQL Kommandos erhalten Sie durch:

```
\help
```

Um Hilfe für ein bestimmtes Kommando zu erhalten, geben Sie (beispielsweise) das Folgende ein:

```
\help create table
```

Siehe auch Psql cheat sheet.

15.2.7 Erstellen von Tabellen in SQL

Wir erstellen nun einige Tabellen! Dazu verwenden wir unser ER-Diagramm als Vorlage. Stellen Sie zuerst die Verbindung zur Adressdatenbank her:

```
psql address
```

Erstellen Sie dann eine Tabelle **:Datei:`streets`**:

```
create table streets (id serial not null primary key, name varchar(50));
```

`serial` und `varchar` sind **Datentypen**. `serial` teilt PostgreSQL mit, dass eine ganzzahlige Reihe (automatisch vergebene Zahlen) zur Füllung des Feldes `id` für jeden neuen Datensatz erstellt werden soll. `varchar(50)` teilt PostgreSQL mit, dass ein Zeichenkettenfeld mit einer Länge von 50 Zeichen erstellt werden soll.

Wir bemerken, dass das Kommando mit einem `;` endet - alle SQL Kommandos sollten so enden. Wenn Sie `Enter` drücken, wird psql eine ähnliche Meldung wie folgt ausgeben:

```
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
```

Das bedeutet, dass Ihre Tabelle mit dem Primärschlüssel `streets_pkey` für `streets.id` erfolgreich erstellt wurde.

Achtung: Wenn Sie die Eingabe ohne `;` beenden, erhalten Sie eine Meldung wie: `address-#`. Dies geschieht, weil PG noch eine weitere Eingabe erwartet. Geben Sie zum Abschließen des Kommandos `;` ein.

Um die Tabelleneigenschaften anzuzeigen, können wir das Folgende eingeben:

```
\d streets
```

Als Ausgabe erhalten wir in etwa das Folgende:

```
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)
```

So zeigen Sie den Inhalt Ihrer Tabellen an:

```
select * from streets;
```

Als Ausgabe erhalten wir in etwa das Folgende:

```
id | name
---+------
(0 rows)
```

Wir sehen, dass unsere Tabelle momentan leer ist.

Try Yourself

Erstellen Sie analog zum obigen Vorgehen eine Tabelle personen:

Fügen Sie Felder wie Telefonnummer, Adresse, Name usw. hinzu (achten Sie darauf, dass die Feldnamen gültig sind). Stellen Sie sicher, dass die Tabelle eine ID Spalte mit demselben Datentyp wie Beispiel oben erhält.

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 Erstellen von Schlüsseln in SQL

Das Problem der obigen Lösung ist, dass die Datenbank nicht weiß, das Personen und Straßen in einer logischen Beziehung stehen. Um diese Beziehung auszudrücken, müssen wir einen Fremdschlüssel erstellen, der auf den Primärschlüssel der Tabelle streets zeigt.

Es gibt zwei Wege, dies zu erreichen:

- Hinzufügen des Schlüssels nachdem die Datenbank erstellt wurde
- Definition des Schlüssels bei Erstellung der Tabelle

Unsere Tabelle wurde schon erstellt, wir verwenden also den ersten Weg:

```
alter table people
  add constraint people_streets_fk foreign key (street_id) references streets(id);
```

Damit teilen wir der Tabelle `people` mit, dass ihr Feld `street_id` einen gültigen Wert aus der Spalte `id` der Tabelle `streets` enthalten muss.

Es ist der gebräuchlichere Weg, eine Einschränkung bei der Erstellung einer Tabelle vorzugeben:

```
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
```

Nach Hinzufügen der Einschränkung sieht unser Tabellenschema wie folgt aus:

```
Table "public.people"

  Column   |         Type          |            Modifiers
-----------+-----------------------+---------------------------------
 id        | integer               | not null default
           |                       | nextval('people_id_seq'::regclass)
 name      | character varying(50) |
 house_no  | integer               | not null
 street_id | integer               | not null
 phone_no  | character varying     |
Indexes:
  "people_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
  "people_streets_fk" FOREIGN KEY (id) REFERENCES streets(id)
```

15.2.9 Indexe mit SQL erstellen

Wir möchten blitzschnelle Suchen nach Namen von Personen. Um dies zu erreichen, können wir einen Index über die Namensspalte unserer Tabelle der Personen erstellen:

```
create index people_name_idx on people(name);

\d people
```

Das Ergebnis ist:

```
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 Löschen von Tabellen mit SQL

Wenn man Tabellen los werden möchte, kann man das Kommando `drop` verwenden:

```
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!

Wenn man dasselbe Kommando `drop table` mit der Tabelle *people* verwendenen würde, wäre es erfolgreich:

```
drop table people;
```

Bemerkung: Wenn Sie das Kommando eingegeben haben und die Tabelle `people` gelöscht wurde, ist es ein guter Zeitpunkt, um die Tabelle neu zu erstellen. Wir werden sie für die nächsten Übungen benötigen.

15.2.11 Ein Wort zu pgAdmin III

Wir zeigen die SQL Kommandos in der *psql* Kommandozeile, da dies ein sehr nützlicher Weg ist, um etwas über Datenbanken zu lernen. Es gibt aber auch schnellere und einfachere Wege, um eine Menge der hier gezeigten Dinge zu tun. Installieren Sie pgAdmin III und Sie können Tabellen mit Hilfe der Maus in einer grafischen Oberfläche erstellen, löschen oder ändern.

Unter Ubuntu installieren Sie es in etwa so:

```
sudo apt install pgadmin3
```

pgAdmin III wird in einem anderen Modul noch ausführlicher behandelt.

15.2.12 In Conclusion

Sie haben jetzt gesehen, wie man eine brandneue Datenbank komplett von null ausgehend erstellen kann.

15.2.13 What's Next?

Als nächstes lernen Sie, wie man das DBMS nutzt, um neue Daten hinzuzufügen.

15.3 Lesson: Daten in das Modell einfügen

Das Modell, das wir erstellt haben, soll jetzt mit den vorgesehenen Daten gefüllt werden.

Ziel dieser Lektion: Das Erlernen des Einfügens neuer Daten in ein Datenbankmodell.

15.3.1 Die Einfügeanweisung

Wie fügen Sie Daten zu einer Tabelle hinzu? Das SQL Kommando `INSERT` stellt die erforderliche Funktionalität dazu bereit:

```
insert into streets (name) values ('High street');
```

Einige Sachen müssen beachtet werden:

- Nach dem Tabellennamen (`streets`) werden die zu füllenden Spaltennamen aufgeführt (in diesem Fall nur die Spalte `name`).
- Fügen Sie nach dem Schlüsselwort `values` die Liste der Feldwerte ein.
- Zeichenketten müssen von einfachen Anführungszeichen umgeben werden.
- Beachten Sie, dass wir keinen Wert für die Spalte `id` einfügen, weil die Spalte automatisch mit einer Sequenz gefüllt wird.
- Wenn man die `id` manuell verändert, riskiert man schwerwiegende Probleme bezüglich der Integrität der Datenbank.

Wir sollten nach erfolgreichem Abschluss die Meldung `INSERT 0 1` sehen.

Wir können das Ergebnis unserer Einfügeaktion durch Auswahl aller in der Tabelle enthaltenden Daten sehen:

```
select * from streets;
```

Ergebnis:

```
select * from streets;
 id |     name
----+--------------
  1 | High street
(1 row)
```

Try Yourself

Nutzen Sie das Kommando `INSERT`, um neue Straßen zur Tabelle `streets` hinzuzufügen.

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 Aneinanderreihen von Einfügeaktionen aufgrund von Beschränkungen

15.3.3 Try Yourself

Versuchen ein Objekt Person zu der Tabelle `people` mit den folgenden Details hinzuzufügen:

```
Name: Joe Smith
House Number: 55
Street: Main Street
Phone: 072 882 33 21
```

Bemerkung: Denken Sie daran, dass wir die Telefonnummern als Zeichenketten, nicht als Ganzzahlen definiert haben.

In diesem Fall erhalten wir eine Fehlermeldung, wenn wir vorher noch keinen Datensatz für Main Street in der Tabelle `streets` erstellt haben.

Wir bemerken auch:

- Man kann keine Straße unter Nutzung ihres Namens hinzufügen
- Man kann vor dem Anlegen des Datensatzes für die Straße in der Tabelle streets keine Straße mit Hilfe der `id` hinzufügen

Denken Sie daran, dass unsere zwei Tabellen mit Hilfe eines Erst-/Fremdschlüsselpaars verbunden sind. Das bedeutet, dass keine zulässige Person ohne zulässigen dazugehörenden Datensatz für eine Straße erstellt werden kann.

Fügen Sie unter Nutzung dieses Wissens eine neue Person zur Datenbank hinzu.

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 Auswahl von Daten

Wir haben die Syntax zur Selektion von Datensätzen schon demonstriert. Lassen Sie uns einige weitere Beispiele ansehen:

```
select name from streets;
```

```
select * from streets;
```

```
select * from streets where name='Main Road';
```

In späteren Abschnitten gehen wir detaillierter darauf ein, wie man Daten auswählt und filtert.

15.3.5 Aktualisierung von Daten

Was ist, wenn man bereits existierende Daten verändern möchte? Z.B. einen Straßennamen der sich geändert hat:

```
update streets set name='New Main Road' where name='Main Road';
```

Seien Sie mit solchen Aktualisierungsanweisungen sehr vorsichtig - wenn mehr als ein Datensatz Ihrer `WHERE` Bedingung entspricht, werden alle aktualisiert!

Eine bessere Lösung ist die Verwendung des Primärschlüssels der Tabelle um den zu ändernden Datensatz anzusprechen:

```
update streets set name='New Main Road' where id=2;
```

Die Ausgabe sollte lauten `UPDATE 1`.

Bemerkung: Die Kriterien der `WHERE` Anweisung unterscheidet zwischen Groß- und Kleinschreibung `Main Road` ist nicht dasselbe wie `Main road`

15.3.6 Löschen von Daten

Nutzen Sie das `DELETE` Kommando, um ein Objekt aus einer Tabelle zu löschen:

```
delete from people where name = 'Joe Smith';
```

Sehen wir uns nun die Tabelle der Personen an:

```
address=# select * from people;

 id | name | house_no | street_id | phone_no
----+------+----------+-----------+----------
(0 rows)
```

15.3.7 Try Yourself

Nutzen Sie das Gelernte, um einige neue Freunde in Ihre Datenbank einzutragen:

```
      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

Sie wissen nun, wie man neue Daten zu vorhandenen schon erstellten Modellen hinzufügt. Denken Sie daran, dass wenn Sie neue Datenarten hinzufügen, sie unter Umständen auch neue Datenmodelle erstellen müssen.

15.3.9 What's Next?

Nachdem wir nun einige Daten hinzugefügt haben, werden wir die Nutzung von Abfragen zum Zugriff auf die Daten lernen.

15.4 Lesson: Abfragen

Wenn Sie ein Kommando wie `SELECT ...` schreiben, wird es in der Regel als Abfrage bezeichnet - Sie befragen die Datenbank nach Informationen.

Ziel dieser Lektion: Zu lernen, wie man Abfragen erstellt, die nützliche Informationen zurückgeben.

Bemerkung: Wenn Sie es noch nicht in der letzten Lektion gemacht haben, fügen Sie die folgenden Personen zur Tabelle `people` hinzu. Wenn Sie Fehlermeldungen zu Einschränkungen des Fremdschlüssels erhalten, müssen Sie zuerst das Objekt ‚Main Road' in die streets Tabelle einfügen

---

```
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 Ordnen der Ergebnisse

Lassen Sie uns eine Personenliste geordnet nach ihrer Hausnummer abrufen:

```
select name, house_no from people order by house_no;
```

Ergebnis:

```
     name      | house_no
---------------+----------
 Joe Bloggs    |        3
 Roger Jones   |       33
 Jane Smith    |       55
 Sally Norman  |       83
(4 rows)
```

Sie können die Ergebnisse nach den Werten von mehreren Spalten sortieren lassen:

```
select name, house_no from people order by name, house_no;
```

Ergebnis:

```
     name      | house_no
---------------+----------
 Jane Smith    |       55
 Joe Bloggs    |        3
 Roger Jones   |       33
 Sally Norman  |       83
(4 rows)
```

15.4.2 Filtern

Oftmals will man nicht jeden einzelnen Datensatz der Datenbank sehen, besonders wenn tausende von Einträgen enthalten sind und man nur einen oder zwei sehen möchte.

Es folgt ein Beispiel eines numerischen Filters, der nur Objekte zurück gibt, deren `house_no` kleiner ist als 50:

```
select name, house_no from people where house_no < 50;

     name     | house_no
  ------------+----------
   Joe Bloggs |        3
   Roger Jones |       33
  (2 rows)
```

Man kann Filter (Benutzung der `WHERE` Klausel) und Sortierung kombinieren (Benutzung der `ORDER BY` Klausel):

```
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)
```

Man kann auch auf Basis von Textdaten filtern:

```
select name, house_no from people where name like '%s%';

     name     | house_no
  ------------+----------
   Joe Bloggs |        3
   Roger Jones |       33
  (2 rows)
```

Hier wurde die Klausel `LIKE` verwendet, um alle Namen, die `s` enthalten zu finden. Man sieht, dass die Abfrage zwischen Groß- und Kleinschreibung unterscheidet, so dass der Eintrag `Sally Norman` nicht zurückgegeben wurde.

Wenn man eine Zeichenkette ohne Unterscheidung zwischen Groß- und Kleinschriebung suchen möchte, kann man die Klausel `ILIKE` verwenden:

```
select name, house_no from people where name ilike '%r%';

      name     | house_no
  -------------+----------
   Roger Jones |       33
   Sally Norman |       83
  (2 rows)
```

Diese Abfrage gab alle Objekte **people** zurück, die ein `r` oder `R` in ihrem Namen enthalten.

15.4.3 Verbindungen

Und wenn man die Details zu einer Person und ihre Adresse anstatt der ID sehen möchte? Um das zu erreichen, muss man die zwei Tabellen mit einer einzelnen Abfrage verbinden. Lassen Sie uns ein Beispiel ansehen:

```
select people.name, house_no, streets.name
from people,streets
where people.street_id=streets.id;
```

Bemerkung: Bei Verbindungen müssen immer die zwei Tabellen angegeben werden, aus denen die Informationen stammen, in diesem Fall people und streets. Man muss außerdem angeben, welche Schlüssel miteinander verbunden werden (Fremd- und Primärschlüssel). Ohne diese Angabe erhält man eine Liste aller möglichen Kombinationen an Personen und Straßen, ohne zu wissen, welche Person in welcher Straße wohnt!

Die richtige Ausgabe sieht wie folgt aus:

```
     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)
```

Wir werden uns Verbindungen später im Zusammenhang mit komplexeren Abfragen noch einmal ansehen. Vorerst müssen wir uns nur merken, dass sie einen einfachen Weg zur Kombination von Informationen aus zwei oder mehr Tabellen darstellen.

15.4.4 Unterauswahl

Unterauswahlen erlauben es, Objekte aus einer Tabelle basierend auf den Daten einer anderen Tabelle auszuwählen, die über eine Fremdschlüsselverbindung angebunden ist. In unserem Fall wollen wir Personen finden, die in einer bestimmten Straße wohnen.

Lassen Sie uns zuerst unsere Daten etwas optimieren:

```
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;
```

Schauen wir uns die Daten nach den Veränderungen an: wir können unsere Abfrage aus dem vorherigen Kapitel wieder verwenden:

```
select people.name, house_no, streets.name
from people,streets
where people.street_id=streets.id;
```

Ergebnis:

```
     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)
```

Nun erstellen wir eine Unterauswahl basierend auf diesen Daten. Wir wollen nur die Personen ausgeben, die in der Straße mit `street_id 1` leben:

```
select people.name
from people, (
    select *
    from streets
    where id=1
  ) as streets_subset
where people.street_id = streets_subset.id;
```

Ergebnis:

```
     name
--------------
 Roger Jones
 Sally Norman
(2 rows)
```

Auch wenn es ein sehr einfaches Beispiel und unnötig für unseren kleinen Datensatz ist, so zeigt es doch wie nützlich und wichtig Unterauswahlen beim Abfragen großer und komplexer Datensätze sind.

15.4.5 Abfragen zur Aggregierung

Eine der mächtigsten Merkmale einer Datenbank ist die Fähigkeit, die Daten in den Tabellen der Datenbank zusammenzufassen. Diese Zusammenfassungen werden Aggregierungs-Abfragen genannt. Es folgt ein typisches Beispiel solch einer Abfrage, die ermittelt, wieviele Personen-Objekte sich in unserer Personentabelle befinden:

```
select count(*) from people;
```

Ergebnis:

```
 count
-------
     4
(1 row)
```

Wir können die Anzahl auch über den Straßennamen zusammenfassen:

```
select count(name), street_id
from people
group by street_id;
```

Ergebnis:

```
 count | street_id
-------+-----------
     2 |         1
     1 |         3
     1 |         2
(3 rows)
```

Bemerkung: Da wir keine `ORDER BY` Klausel verwendet haben, weicht die Reihenfolge der Ergebnisse unter Umständen von der hier dargestellten ab.

Try Yourself

Fassen Sie die Personen nach Straßen zusammen und geben Sie die richtigen Straßen anstatt der street_ids aus.

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

Wir haben gesehen, wie man Abfragen zur Ausgabe von Daten nutzt, um sinnvolle Information aus unserer Datenbank zu erhalten.

15.4.7 What's Next?

Als Nächstes werden wir sehen, wie man Sichten aus den erstellten Abfragen erstellt.

15.5 Lesson: Ansichten

Das Schreiben einer Abfrage erfordert viel Zeit und Aufwand. Mit Hilfe von Sichten kann man die Formulierung einer SQL Abfrage in Form einer wiederverwendbaren ‚virtuellen Tabelle' speichern.

Ziel dieser Lektion: Das Speichern einer Abfrage als Sicht.

15.5.1 Erstellen einer Sicht

Man kann eine Sicht wie eine Tabelle behandeln, die Daten stammen aber aus einer Abfrage. Sehen wir uns eine einfache Sicht an:

```
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;
```

Wie man sieht, ist der einzige Unterschied der Teil `create view roads_count_v as` am Anfang. Wir können nun Daten aus der Sicht auswählen:

```
select * from roads_count_v;
```

Ergebnis:

```
 count |    name
-------+-------------
     1 | Main Road
     2 | High street
     1 | Low Street
(3 rows)
```

15.5.2 Veränderung einer Sicht

Eine Sicht ist nicht unveränderlich und enthält auch keine ‚echten Daten'. Das bedeutet, dass sie leicht verändert werden kann, ohne die eigentlichen Daten in der Datenbank zu beeinflussen:

```
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;
```

(Dieses Beispiel zeigt die empfohlene Praxis zur Nutzung von GROSSSCHREIBUNG für alle SQL Schlüsselwörter.)

Wir haben eine `ORDER BY` Klausel hinzugefügt, so dass die Zeilen der Sicht nun schön geordnet sind:

```
select * from roads_count_v;

 count |    name
-------+-------------
     2 | High street
     1 | Low Street
     1 | Main Road
(3 rows)
```

15.5.3 Verwerfen einer Sicht

Wenn man eine Sicht nicht länger benötigt, kann sie wie folgt gelöscht werden:

```
drop view roads_count_v;
```

15.5.4 In Conclusion

Mit Hilfe von Sichten kann man Abfragen speichern. Auf die Ausgabe der Abfragen kann man dann wie auf eine Tabelle zugreifen.

15.5.5 What's Next?

Manchmal möchte man, dass die Ädnerung von bestimmten Daten auch Auswirkungen auf andere Teile der Datenbank hat. In der nächsten Lektion sehen wir, wie man das macht.

15.6 Lesson: Regeln

Regeln erlauben es den „Abfragebaum" einer eingebenen Abfrage umzuschreiben. Eine verbreitete Nutzung ist die Erstellung von Sichten, einschließlich aktualisierbarer Sichten. - *Wikipedia*

Ziel dieser Lektion: Zu lernen, wie man neue Regeln für eine Datenbank erstellt.

15.6.1 Erstellen einer Regel zum protokollieren

Angenommen man möchte alle Änderungen des Feldes phone_no in der Tabelle people in einer anderen Tabelle people_log aufzeichnen. Dann müsste man eine neue Tabelle erstellen:

```
create table people_log (name text, time timestamp default NOW());
```

Im nächsten Schritt erstellt man eine Regel, die alle Änderungen des Feldes phone_no in der Tabelle people in die Tabelle people_log protokolliert:

```
create rule people_log as on update to people
  where NEW.phone_no <> OLD.phone_no
  do insert into people_log values (OLD.name);
```

Um die Regel zu testen, ändern wir eine Telefonnummer:

```
update people set phone_no = '082 555 1234' where id = 2;
```

Überprüfen Sie, ob die Tabelle `people` richtig aktualisiert wurde:

```
select * from people where id=2;

 id |    name     | house_no | street_id |   phone_no
----+-------------+----------+-----------+--------------
  2 | Joe Bloggs |        3 |         2 | 082 555 1234
(1 row)
```

Mit Hilfe der Regel die wir erstellt haben, sieht die Tabelle `people_log` nun wie folgt aus:

```
select * from people_log;

    name     |            time
-------------+----------------------------
 Joe Bloggs | 2014-01-11 14:15:11.953141
(1 row)
```

Bemerkung: Der Wert des Feldes `time` wird entsprechend des aktuellen Datums und der Zeit vergeben.

15.6.2 In Conclusion

Regeln erlauben es, Daten automatisch hinzuzufügen oder zu ändern basierend auf Änderungen in anderen Teilen der Datenbank.

15.6.3 What's Next?

Das nächste Modul enthält eine Einführung in räumliche Datenbanken mit PostGIS. Die behandelten Datenbankkonzepte werden mit GIS Daten angewendet.

KAPITEL 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 created by Boundless that are now hosted on the PostGIS's website:

- Introduction to PostGIS
- PostGIS Database Tips and Tricks

See also PostGIS In Action.

16.1 Lesson: PostGIS-Einrichtung

Die Einrichtung von PostGIS Funktionen erlaubt den Zugriff auf räumliche Funktionen innerhalb von PostgreSQL.

Ziel dieser Lektion: Die Installation von räumlichen Funktionen und eine kurze Demonstration der Auswirkungen.

Bemerkung: 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 Installation unter Ubuntu

Postgis kann leicht mit Hilfe von apt installiert werden.

```
$ sudo apt install postgresql
$ sudo apt install postgis
```

Wirklich, es ist nicht schwer…

Bemerkung: 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 Installation unter 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.

Weitere Informationen zur Installation unter Windows finden Sie hier: PostGIS website.

16.1.3 Installation auf anderen Plattformen

The PostGIS website download has information about installing on other platforms including macOS and on other Linux distributions

16.1.4 Konfiguration von Datenbanken zur Nutzung von PostGIS

Nachdem PostGIS installiert wurde, muss Ihre Datenbank zur Nutzung der Erweiterungen konfiguriert werden. Wenn Sie PostGIS in der Version > 2.0 installiert haben, ist das mit dem folgenden Kommando, hier für die Adressendatenbank aus der vorherigen Übung, ganz einfach.

```
$ psql -d address -c "CREATE EXTENSION postgis;"
```

Bemerkung: 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 Die installierten PostGIS Funktionen

PostGIS kann man als eine Sammlung von in der Datenbank integrierten Funktionen sehen, die die Kernfunktionalität von PostgreSQL für den Umgang mit räumlichen Daten erweitern. Unter ‚Umgang mit räumlichen Daten' verstehen wir das speichern, abrufen, abfragen und verändern von Daten. Um das zu erreichen, werden eine Reihe von Funktionen in der Datenbank installiert.

Unsere PostgreSQL `address` Datenbank ist jetzt dank PostGIS für räumliche Daten vorbereitet. Wir werden das in den kommenden Kapiteln noch vertiefen, aber hier erst einmal eine kleine Kostprobe. Sagen wir, wir wollen einen Punkt aus einem Text erstellen. Wir verwenden zuerst das psql Kommando, um Funktionen mit Bezug zu Punkten zu finden. Wenn Sie noch nicht mit der Datenbank `address` verbunden sind, stellen Sie die Verbindung jetzt her. Starten sie dann:

```
\df *point*
```

Dieses Kommando haben wir gesucht: `st_pointfromtext`. Benutzen Sie die Pfeil-nach-unten-Taste um die Liste durchzublättern. Drücken Sie danach `Q`, um zur psql Kommandozeile zurückzukehren.

Probieren Sie dieses Kommando:

```
select st_pointfromtext('POINT(1 1)');
```

Ergebnis:

```
st_pointfromtext
----------------------------------------------
0101000000000000000000F03F000000000000F03F
(1 row)
```

Drei wichtige Dinge:

- Mit Hilfe von `POINT(1 1)` haben wir einen Punkt an der Position 1,1 (Annahme: EPSG:4326) definiert,
- Wir haben eine SQL Anweisung ausgeführt, aber nicht an einer Tabelle, sondern nur über Daten, die in der Kommandozeile eingegeben wurden,
- Die Ausgabezeile ergibt nicht viel Sinn.

Die Ausgabe erfolgte im OGC Format ‚Well Known Binary' (WKB). Wir werden uns dieses Format im nächsten Kapitel genauer ansehen.

Um die Ergebnisse in Textform zu sehen, durchsuchen wir kurz die Funktionsliste nach etwas, das Text ausgibt:

```
\df *text
```

Die Abfrage nach der wir suchen ist `st_astext`. Wir kombinieren sie mit der vorherigen Abfrage:

```
select st_astext(st_pointfromtext('POINT(1 1)'));
```

Ergebnis:

```
 st_astext
------------
  POINT(1 1)
  (1 row)
```

Wir gaben die Zeichenkette `POINT(1,1)` ein, wandelten sie mit Hilfe von `st_pointfromtext()` in einen Punkt, um sie anschließend mit Hilfe von `st_astext()` zurück in eine vom Menschen lesbare Form zu bringen, d.h. in unsere anfängliche Zeichenkette.

Ein letztes Beispiel bevor wir uns eingehend mit PostGIS beschäftigen:

```
select st_astext(st_buffer(st_pointfromtext('POINT(1 1)'),1.0));
```

Was passiert hier? Es wird ein Puffer von 1 Grad um unseren Punkt erstellt und das Ergebnis als Text ausgegeben.

16.1.6 Räumliche Bezugssysteme

Zusätzlich zu den PostGIS Funktionen enthält die Erweiterung eine Sammlung an Definitionen von räumlichen Bezugssystemen (RBS) entsprechend der European Petroleum Survey Group (EPSG). Sie werden für Operationen wie z.B. Transformation von Koordinatenbezugssystemen (KBS) verwendet.

Wir können die RBS Definitionen untersuchen. Sie sind in normalen Datenbanktabellen gespeichert.

Lassen Sie uns zuerst das Schema der Tabelle mit Hilfe des folgenden Kommandos am psql Prompt ansehen:

```
\d spatial_ref_sys
```

Das Ergebnis sollte wie folgt aussehen:

```
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)
```

Man kann normale SQL Abfragen (die wir in den einleitenden Kapiteln gelernt haben) verwenden, um die Tabelle anzusehen oder zu verändern. Es ist allerdings keine gute Idee Datensätze in der Tabelle zu verändern oder zu löschen, außer man weiß genau was man tut.

Eine interessante SRID ist EPSG:4326 - das geographische Referenzsystem mit Längen- und Breitengraden, das den WGS 84 Ellipsoid verwendet. Sehen wir es uns genauer an:

```
select * from spatial_ref_sys where srid=4326;
```

Ergebnis:

```
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
```

`srtext` enthält die Definition der Projektion im well known text format (ähnlich der .prj Datei bei shape-Dateien).

16.1.7 In Conclusion

Wir haben nun PostGIS Funktionen in unserer PostgreSQL Datenbank installiert. Damit sind wir in der Lage die umfangreichen räumlichen Funktionen von PostGIS zu nutzen.

16.1.8 What's Next?

Als nächstes lernen wir, wie räumliche Objekte in einer Datenbank abgebildet werden.

16.2 Simple-Feature-Modell

Wie können wir geographische Objekte in einer Datenbank repräsentieren und speichern? In dieser Lektion behandeln wir einen Ansatz, nämlich das von der OGC definierte Simple-Feature-Modell.

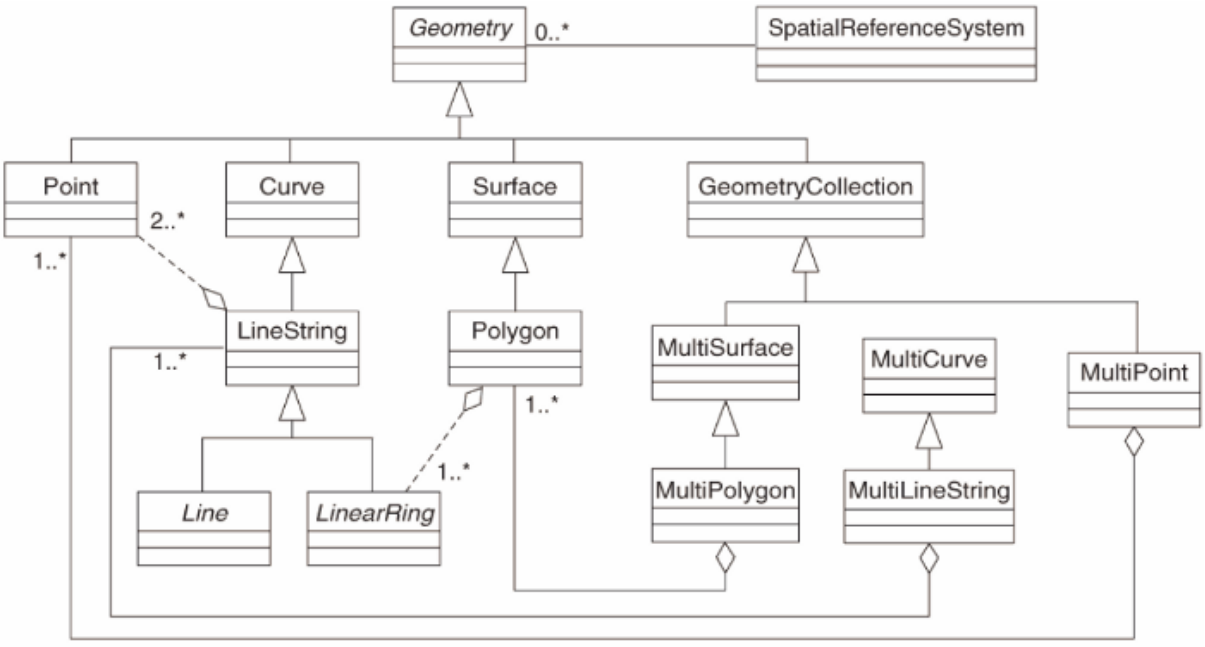
Ziel dieser Lektion: Das SFS Modell kennen und nutzen lernen.

16.2.1 Was ist OGC

Das Open Geospatial Consortium (OGC) ist eine internationale gemeinnützige Organisation, die 1994 gegründet wurde und allgemeingültige Standards konsensual festlegt. In der OGC arbeiten mehr als 370+ kommerzielle, gemeinnützige, wissenschaftliche und staatliche Organisationen weltweit in einem offenen konsensualen Prozess an der Förderung der Entwicklung und Umsetzung von Standards für räumliche Daten und Dienste, GIS Datenverarbeitung und dem Teilen von Daten. - *Wikipedia*

16.2.2 Was ist das SFS Modell

Das Simple-Feature-Modell für SQL (SFS) ist ein *nicht-topologischer* Weg, um räumliche Daten in einer Datenbank zu speichern. Es definiert Funktionen zum Zugriff, zur Verarbeitung und zum Aufbau solcher Daten.



Das Modell definiert räumliche Daten des Typs Punkt, Linie und Polygon (und deren Aggregation zu zusammengesetzten Objekten).

For further information, have a look at the OGC Simple Feature for SQL standard.

16.2.3 Hinzufügen eines Geometriefeldes zu einer Tabelle

Wir fügen ein Punktfeld in unsere people Tabelle ein:

```
alter table people add column the_geom geometry;
```

16.2.4 Hinzufügen einer Einschränkung basierend auf dem Geometrietyp

Wir sehen, dass das Geometriefeld nicht automatisch den Typ der Geometrie bestimmt. Dies wird mit einer Einschränkung festgelegt:

```
alter table people
add constraint people_geom_point_chk
    check(st_geometrytype(the_geom) = 'ST_Point'::text
          OR the_geom IS NULL);
```

Damit wird der Inhalt der Tabelle eingeschränkt, so dass nur noch eine Punktgeometrie oder der NULL Wert akzeptiert werden.

16.2.5 Try Yourself

Erstellen Sie eine neue Tabelle cities und fügen Sie einige passende Felder und ein Geomtriefeld für Polygone (die Stadtgrenzen) hinzu. Stellen Sie sicher, dass die Geometrien auf Polygone eingeschränkt sind.

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 Füllen der geometry_columns Tabelle

An dieser Stelle sollten wir einen Eintrag in `geometry_columns` Tabelle machen:

```
insert into geometry_columns values
  ('','public','people','the_geom',2,4326,'POINT');
```

Warum? `geometry_columns` wird von einigen Anwendungen genutzt, um zu erkennen, welche Tabellen einer Datenbank räumliche Daten enthalten.

Bemerkung: Wenn die obige `INSERT` Anweisung einen Fehler zurück gibt, starten Sie zuerst die folgende Abfrage:

```
select * from geometry_columns;
```

Wenn die Spalte `f_table_name` den Wert `people` enthält, wurde die Tabelle schon registriert und man muss nichts weiter tun.

Der Wert `2` bezieht sich auf die Anzahl der räumlichen Dimensionen, in diesem Fall zwei: **X** und **Y**.

Der Wert `4326` bezieht sich auf die genutzte Projektion. In diesem Fall ist das WGS 84, auf die sich die Nummer 4326 bezieht (wurde vorher unter dem Stichwort EPSG diskutiert).

Try Yourself

Hinzufügen eines passenden Eintrags *geometry_columns* für unseren neuen cities Layer

Answer

```
insert into geometry_columns values
      ('','public','cities','the_geom',2,4326,'POLYGON');
```

16.2.7 Hinzufügen eines Geomtriedatensatzes zur Tabelle mit Hilfe von SQL

Jetzt sind unsere Tabelle für räumliche Daten vorbereitet und wir können Geometrien darin ablegen:

```
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)');
```

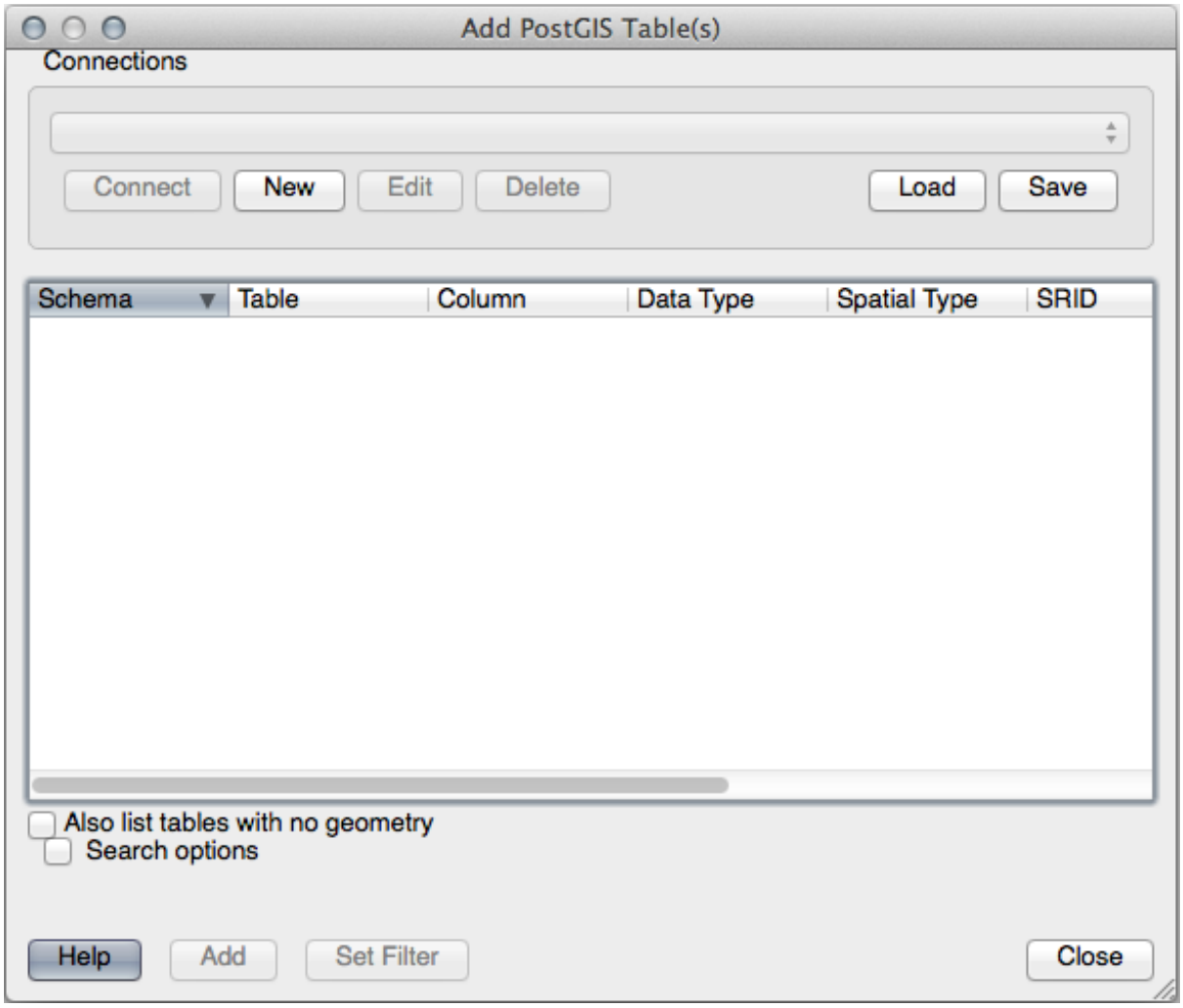
Bemerkung: In dem obigen Eintrag müssen Sie angeben, welche Projektion (SRID) genutzt werden soll. Das liegt daran, dass wir die Geometrie des neuen Punktes als reinen Text eingegeben haben. Dadurch wurde nicht automatisch die richtige Projektion hinzugefügt. Offensichtlich benötigt der neue Punkt dieselbe SRID wie der Datensatz zu dem er hinzugefügt wird. Wir müssen die SRID daher vorgeben.

Wenn man an dieser Stelle eine graphische Benutzerschnittstelle verwendet, würde z.B. die Projektion für jeden Punkt automatisch vergeben werden. Man braucht sich in diesem Fall keine Gedanken über die richtige Projektion für jeden einzelnen hinzuzufügenden Punkt machen, wenn man die Projektion, wie wir es gemacht haben, vorher für den Datensatz festgelegt hat.

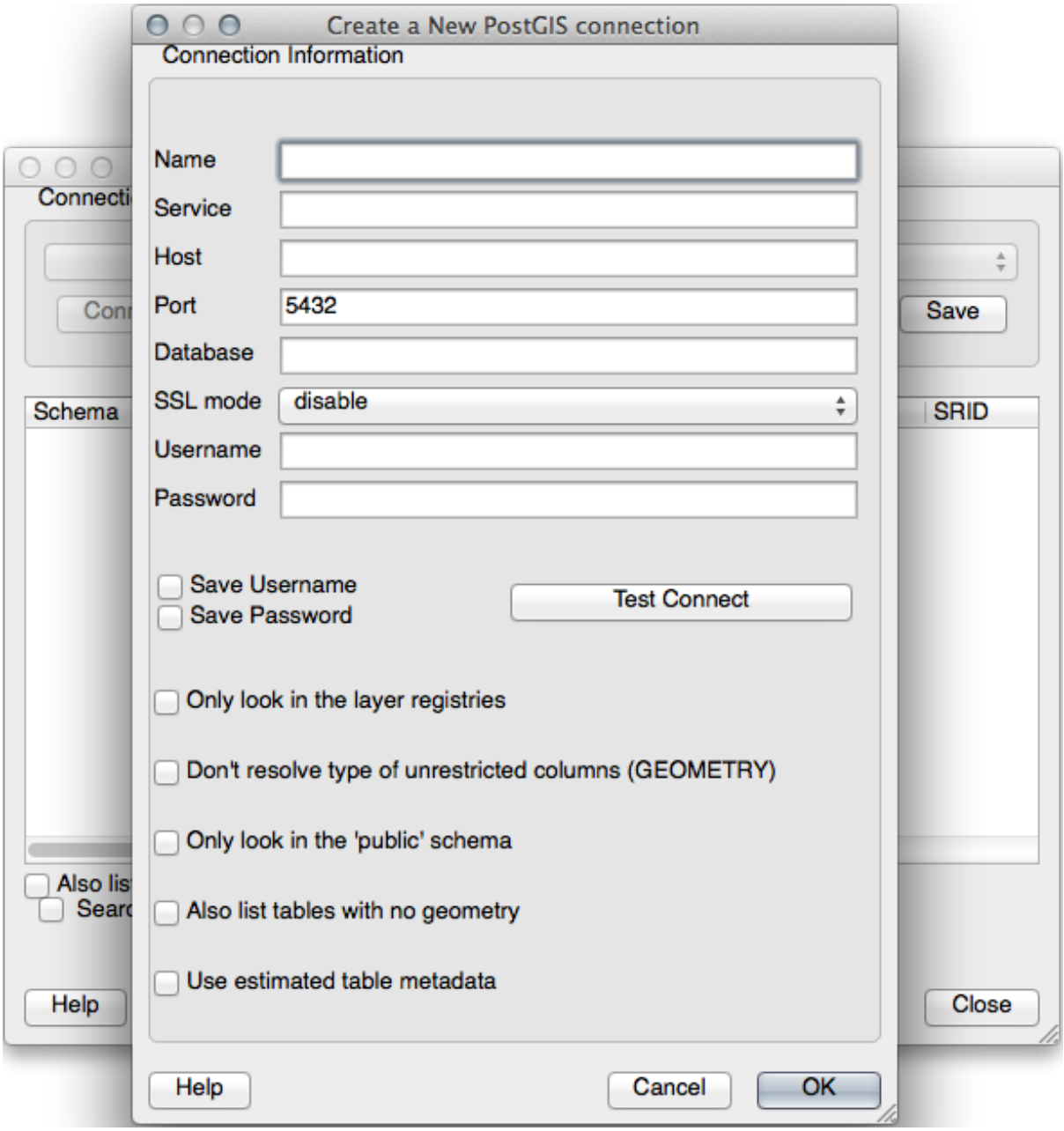
Jetzt ist ein guter Zeitpunkt, um QGIS zu öffnen und zu versuchen unsere `people` anzuzeigen. Wir sollten auch versuchen Datensätze zu editieren, hinzuzufügen und zu löschen und dann Auswahlabfragen erstellen, um zu sehen wie sich die Datensätze ändern.

Um einen PostGIS Layer in QGIS zu laden, wählt man im Menü *Layer* ► *PostGIS-Layer hinzufügen* oder drückt den entsprechenden Knopf auf der Werkzeugleiste:

Es öffnet sich der folgende Dialog:



Klicken Sie auf die Schaltfläche *Neu*, um den folgenden Dialog zu öffnen:



Erstellen Sie dann eine neue Verbindung, z.B.:

```
Name: myPG
Service:
Host: localhost
Port: 5432
Database: address
User:
Password:
```

Klicken Sie auf *Test Connect*, um zu testen, ob QGIS die Datenbank `address` gefunden hat und Benutzername und Passwort richtig sind. Wenn es funktioniert, Setzen einen Haken bei *Benutzername speichern* und *Passwort speichern*. Klicken Sie danach zum Erstellen der Verbindung auf *OK*.

Zurück im Dialog *PostGIS-Layer hinzufügen* klicken Sie auf *Verbinden* und fügen wie gewohnt Layer zu Ihrem Projekt hinzu.

Try Yourself

Formulieren Sie eine Abfrage, die Personennamen, Straßennamen und Lage (aus der the_geom Spalte) als reinen Text anzeigt.

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

Wir haben gesehen, wie man räumliche Objekte zu unserer Datenbank hinzufügt und sie in einem GIS anzeigt.

16.2.9 What's Next?

Als nächstes werden wir sehen, wie man Daten in unsere Datenbank importiert und exportiert.

16.3 Lesson: Importieren und Exportieren

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 library 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: Räumliche Abfragen

Räumliche Abfragen unterscheiden sich nicht von anderen Datenbankabfragen. Man kann die Geometriespalte wie jede andere Spalte in der Datenbank verwenden. Durch die Installation von PostGIS in unserer Datenbank kommen weitere zusätzliche Funktionen zur Abfrage der Datenbank hinzu.

Ziel dieser Lektion: Verständnis über die Implementierung von räumlichen Funktionen im Vergleich mit „normalen“ nicht räumlichen Funktionen.

16.4.1 Räumliche Operationen

Wenn man wissen möchte, welche Punkte innerhalb eines Abstandes von 2 Grad zu einem Punkt (X,Y) sind, kann man folgende Abfrage verwenden:

```
select *
from people
where st_distance(the_geom,'SRID=4326;POINT(33 -34)') < 2;
```

Ergebnis:

```
 id |     name      | house_no | street_id |   phone_no     |   the_geom
----+---------------+----------+-----------+----------------+---------------
  6 | Fault Towers |       34 |         3 | 072 812 31 28 | 01010008040C0
(1 row)
```

Bemerkung: Der obige Wert von the_geom wurde aus Platzgründen abgeschnitten. Wenn man den Punkt in vom Menschen lesbaren Koordinaten sehen möchte, kann man so ähnlich vorgehen wie oben unter „Einen Punkt als WKT anzeigen“.

Woher wissen wir, das die obige Abfrage alle Punkte innhalb von 2 *Grad* ausgibt? Warum nicht 2 *Meter* oder irgendeine andere Einheit?

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 Räumliche Indexe

Wir können auch räumliche Indexe definieren. Ein räumlicher Index beschleunigt Ihre räumlichen Abfragen stark. Um einen räumlichen Index über eine Geometriespalte zu erstellen, geht man wie folgt vor:

```
CREATE INDEX people_geo_idx
  ON people
  USING gist
  (the_geom);

\d people
```

Ergebnis:

```
Table "public.people"
   Column    |          Type          |                 Modifiers
 ------------+------------------------+-------------------------------------------
  id         | integer                | not null default
             |                        | nextval('people_id_seq'::regclass)
  name       | character varying(50)  |
  house_no   | integer                | not null
  street_id  | integer                | not null
  phone_no   | character varying      |
  the_geom   | geometry               |
Indexes:
  "people_pkey" PRIMARY KEY, btree (id)
  "people_geo_idx" gist (the_geom)  <-- new spatial key added
  "people_name_idx" btree (name)
Check constraints:
  "people_geom_point_chk" CHECK (st_geometrytype(the_geom) = 'ST_Point'::text
  OR the_geom IS NULL)
Foreign-key constraints:
  "people_street_id_fkey" FOREIGN KEY (street_id) REFERENCES streets(id)
```

16.4.3 Try Yourself

Verändern Sie die Tabelle cities, so dass die Geometriespalte einen räumlichen Index erhält.

Answer

```
CREATE INDEX cities_geo_idx
  ON cities
  USING gist (the_geom);
```

16.4.4 Demonstration der räumlichen Funktionen von PostGIS

Um die räumlichen Funktionen von PostGIS zu demonstrieren, werden wir eine neue Datenbank mit (fiktionalen) Daten erstellen.

Erstellen Sie zu Beginn eine neue Datenbank (verlassen Sie die psql Konsole zuerst):

```
createdb postgis_demo
```

Denken Sie daran, die PostGIS Erweiterungen zu installieren:

```
psql -d postgis_demo -c "CREATE EXTENSION postgis;"
```

Importieren Sie als Nächstes die Daten aus dem Verzeichnis `exercise_data/postgis/`. Sehen Sie sich dazu die Anleitung in der vorherigen Lektion an, aber denken Sie daran, dass Sie eine neue PostGIS Verbindung zu der neuen Datenbank erstellen müssen. Sie können mit Hilfe des Terminals oder des DB Managers importieren. Importieren Sie die Dateien in die folgenden Datenbanktabellen:

- `points.shp` nach `building`
- `lines.shp` nach `road`
- `polygons.shp` nach `region`

Laden Sie diese drei Datenbanklayer wie gewohnt mit Hilfe des *PostGIS-Layer hinzufügen* Dialogs in QGIS ein. Beim Öffnen der Attributtabellen sehen wir, dass beide ein Feld `id` und ein Feld `gid` haben, die beim PostGIS Import erzeugt wurden.

Nachdem die Tabellen importiert wurden, können wir PostGIS zur Abfrage der Daten verwenden. Gehen Sie zurück zur Konsole (Kommandozeile) und geben folgendes Kommando ein:

```
psql postgis_demo
```

Wir werden einige dieser Auswahlanweisungen durch die Erstellung von Sichten demonstrieren, so dass Sie sie in QGIS öffnen und die Ergebnisse sehen können.

Auswahl nach der Lage

Wir erhalten alle Bauwerke in der Region KwaZulu:

```
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';
```

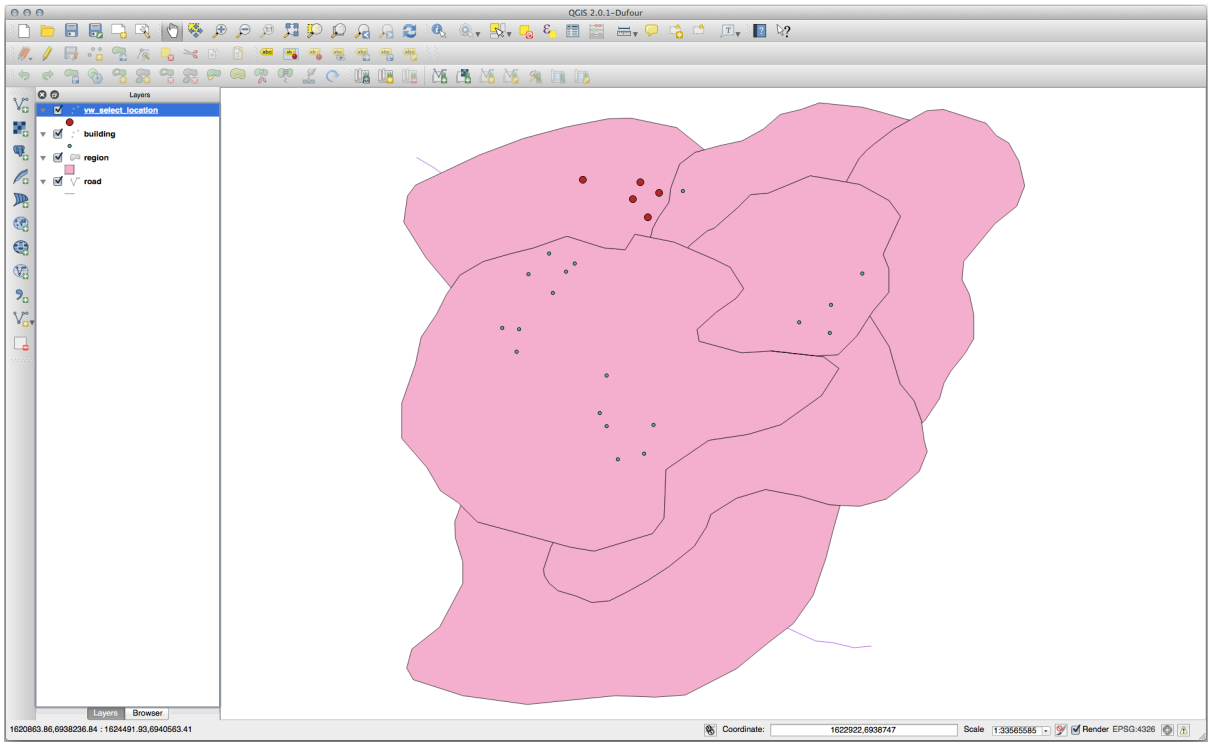
Ergebnis:

```
 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)
```

Oder wir erstellen eine Sicht daraus:

```
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';
```

Fügen Sie die Sicht als Layer in QGIS ein:



Nachbarn auswählen

Geben Sie eine Liste aller Namen von Regionen aus, die an die Hokkaido Region angrenzen:

```
SELECT b.name
  FROM region a, region b
    WHERE st_touches(a.the_geom, b.the_geom)
    AND a.name = 'Hokkaido';
```

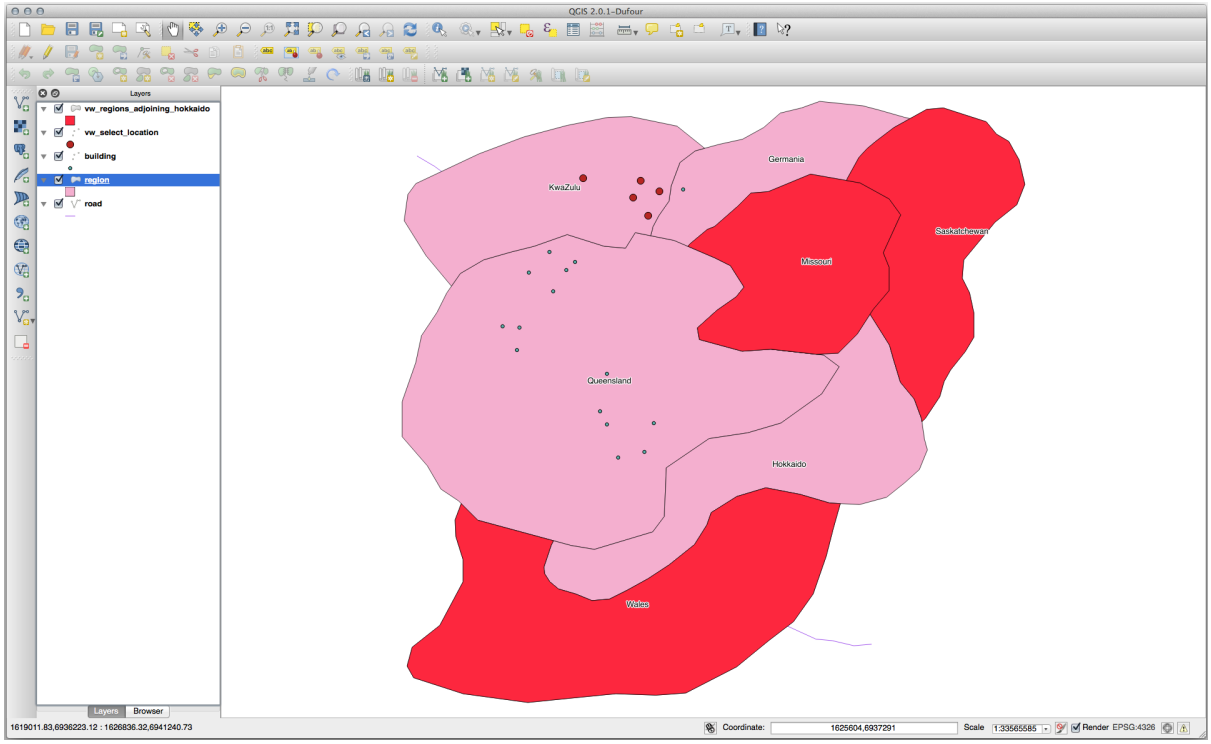
Ergebnis:

```
    name
--------------
 Missouri
 Saskatchewan
 Wales
(3 rows)
```

Als Sicht:

```
CREATE VIEW vw_regions_adjoining_hokkaido AS
  SELECT b.gid, b.name, b.the_geom
    FROM region a, region b
      WHERE st_touches(a.the_geom, b.the_geom)
      AND a.name = 'Hokkaido';
```

In QGIS:

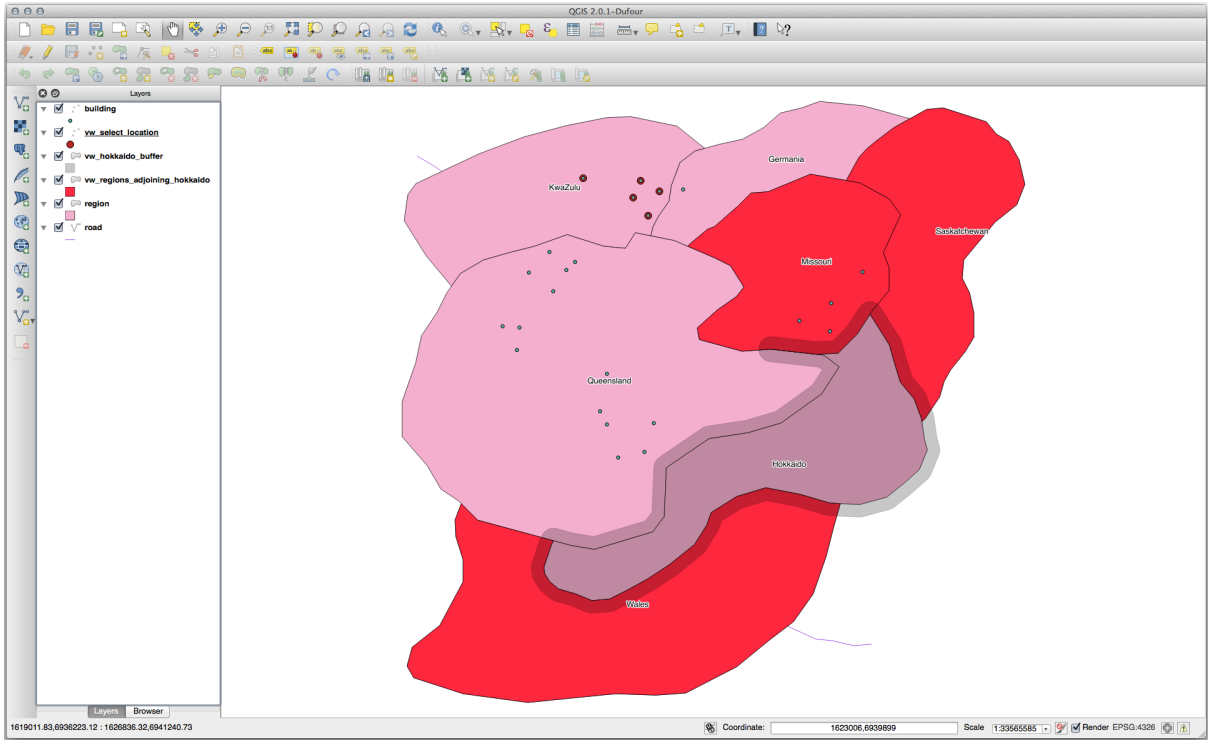


Achten Sie auf die fehlende Region (Queensland). Das kann an einem Topologiefehler liegen. Solche Artefakte können auf potentielle Fehler in den Daten hinweisen. Um das Problem zu lösen ohne sich in möglichen Problemen des Datensatzes zu verzetteln, können wir einen Puffer verwenden:

```
CREATE VIEW vw_hokkaido_buffer AS
  SELECT gid, ST_BUFFER(the_geom, 100) as the_geom
    FROM region
      WHERE name = 'Hokkaido';
```

Damit wird ein Puffer von 100 Metern um die Region Hokkaido erstellt.

Die dunklere Fläche ist der Puffer:

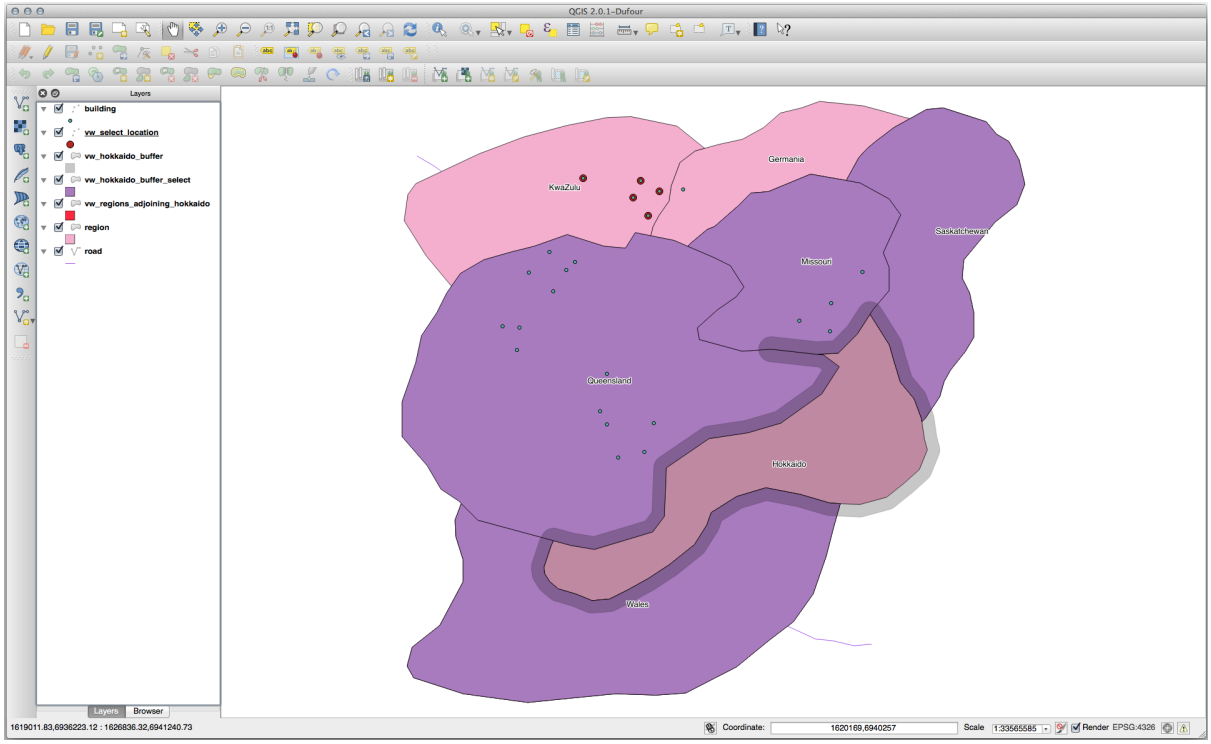


Auswahl mit Hilfe des Puffers:

```
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 dieser Abfrage wird die Sicht mit Puffer genauso wie eine normale Tabelle verwendet. Die Sicht erhält den Alias `a` und ihr Geometriefeld `a.the_geom` wird verwendet, um alle Polygone in der Tabelle `region` (Alias `b`) zu selektieren die sie überschneiden. Hokkaido ist dagegen aus der Auswahl ausgeschlossen, da wir nur die angrenzenden Regionen selektieren wollen.

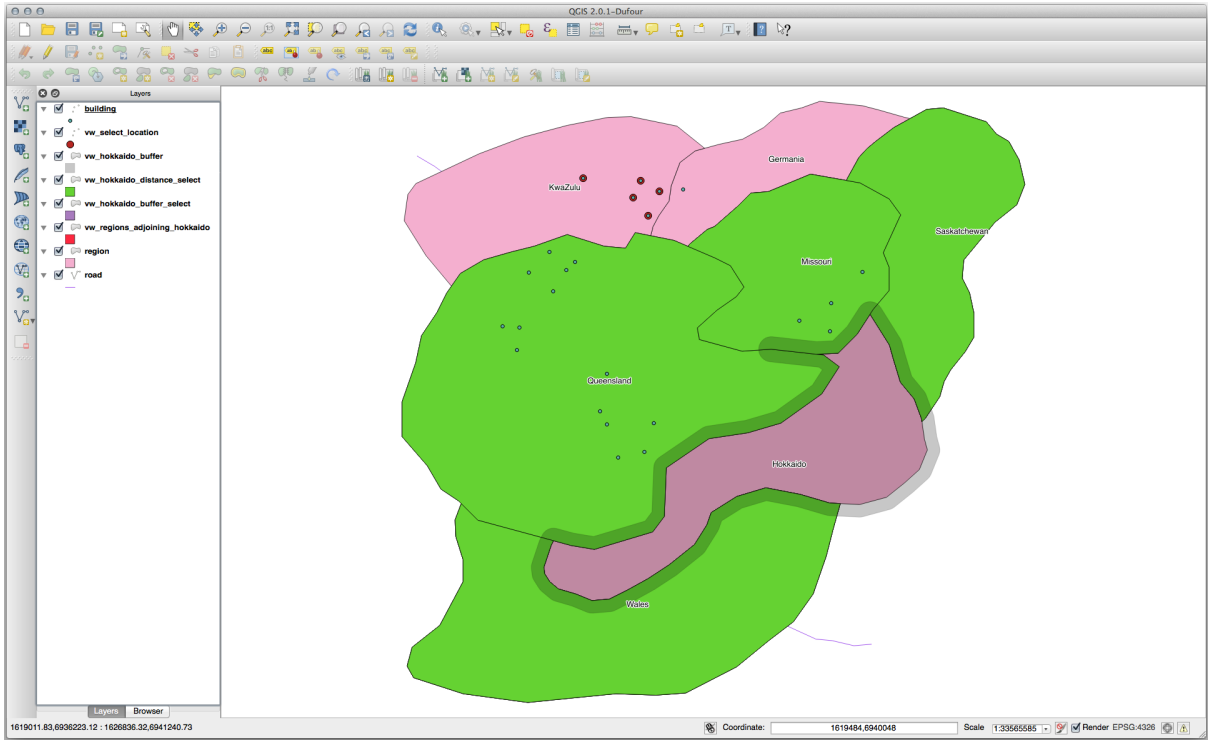
In QGIS:



Es ist auch möglich, alle Objekte innerhalb eines vorgegebenen Abstandes auszuwählen, ohne vorher einen Puffer zu erstellen:

```
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';
```

Das Ergebnis ist dasselbe und der Zwischenschritt der Pufferbildung entfällt:



Auswahl eindeutiger Werte

Anzeige einer Liste der eindeutigen Stadtnamen für alle Gebäude in der Region Queensland:

```
SELECT DISTINCT a.name
  FROM building a, region b
    WHERE st_within(a.the_geom, b.the_geom)
    AND b.name = 'Queensland';
```

Ergebnis:

```
  name
---------
 Beijing
 Berlin
 Atlanta
(3 rows)
```

Weitere Beispiele ...

```
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

Wir haben gesehen, wie man räumliche Objekte mit Hilfe der neuen Datenbankfunktionen in QGIS abfragen kann.

16.4.6 What's Next?

Als Nächstes werden wir die Strukturen komplexerer Geometrien untersuchen und wie man sie mit Hilfe von PostGIS erstellt.

16.5 Lesson: Geometrieaufbau

In diesem Kapitel gehen wir näher darauf ein, wie einfach Geometrien in SQL erstellt werden. In der Realität wird man wahrscheinlich ein GIS wie QGIS mit seinen Digitalisierungshilfen verwenden, um komplexe Geometrien zu erstellen; trotzdem ist das Verständnis für die Erstellung von Abfragen und für die Nachvollziehbarkeit wie die Daten aufgebaut ist nützlich.

Ziel dieser Lektion: Ein besseres Verständnis wie man räumliche Objekte direkt in PostgreSQL/PostGIS erstellt.

16.5.1 Erstellen von Linien

Lassen Sie uns unsere `address` Datenbank an die anderen Datenbanken anpassen; sie soll eine Einschränkung bezüglich der Geometrie, einen Index und einen Eintrag in der geometry_columns Tabelle erhalten.

16.5.2 Try Yourself

- Verändern Sie die Tabelle `streets` so, dass sie eine Geometriespalte des Typs ST_LineString enthält.
- Vergessen Sie dabei nicht, die geometry_columns Tabelle zu aktualisieren!
- Fügen Sie auch eine Einschränkung ein, die verhindert, dass andere Geometrien als Linien oder NULL hinzugefügt werden.
- Erstellen Sie einen räumlichen Index über die neue Geometriespalte

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);
```

Lassen Sie uns nun eine Linie in unsere Tabelle streets einfügen. In diesem Fall aktualisieren wir einen vorhandenen Straßen-Datensatz:

```
update streets
set the_geom = 'SRID=4326;LINESTRING(20 -33, 21 -34, 24 -33)'
where streets.id=2;
```

Sehen Sie sich die Ergebnisse in QGIS an. (Sie müssen unter Umständen mit der rechten Maustaste auf den streets-Layer im ‚Layer' Bereich klicken und dann ‚Auf den Layer zoomen' auswählen.)

Erstellen Sie noch ein Paar Einträge für Straßen - einige in QGIS und einige aus der Kommandozeile.

16.5.3 Erstellung von Polygonen

Die Erstellung von Polygonen ist genauso einfach. Dabei muss man daran denken, dass Polygone definitionsgemäß mindestens vier Stützpunkte haben, wobei der erste und letzte übereinander liegen:

```
insert into cities (name, the_geom)
values ('Tokyo', 'SRID=4326;POLYGON((10 -10, 5 -32, 30 -27, 10 -10))');
```

Bemerkung: Ein Polygon erfordert doppelte Klammern um seine Koordinatenliste. Dies erlaubt es, komplexe Polygone mit mehreren unverbundenen Flächen einzufügen. Zum Beispiel

```
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))'
        );
```

Nach dem Abschlusses dieses Schrittes, können Sie die Städte in QGIS einladen und nachvollziehen, was sich verändert hat. Öffnen Sie dazu die Attributtabelle und wählen den neuen Eintrag aus. Beachten Sie, wie die neuen Polygone sich wie ein einziges Polygon verhalten.

16.5.4 Übernung: Anbindung Städte an Personen

Gehen Sie für diese Übung wie folgt vor:

- Löschen Sie alle Daten aus der Tabelle people.
- Fügen Sie eine Spalte mit einem Fremdschlüssel in der Tabelle people ein, der eine Referenz zum Primärschlüssel der Tabelle cities herstellt.
- Nutzen Sie QGIS, um einige Städte zu erfassen.
- Verwenden Sie SQL um einige neue Datensätze zu people hinzuzufügen. stellen Sie sicher, dass jeder Datensatz eine zugehörige Straße und Stadt enthält.

Ihr aktualsiertes Schema zu people sollte in etwa so aussehen:

```
\d people

Table "public.people"
   Column   |         Type          |                       Modifiers
 -----------+-----------------------+-----------------------------------------------
  id        | integer               | not null
            |                       | default nextval('people_id_seq'::regclass)
  name      | character varying(50) |
  house_no  | integer               | not null
  street_id | integer               | not null
  phone_no  | character varying     |
  the_geom  | geometry              |
  city_id   | integer               | not null
Indexes:
  "people_pkey" PRIMARY KEY, btree (id)
  "people_name_idx" btree (name)
Check constraints:
  "people_geom_point_chk" CHECK (st_geometrytype(the_geom) =
                      'ST_Point'::text OR the_geom IS NULL)
Foreign-key constraints:
  "people_city_id_fkey" FOREIGN KEY (city_id) REFERENCES cities(id)
  "people_street_id_fkey" FOREIGN KEY (street_id) REFERENCES streets(id)
```

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,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
            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)');
```

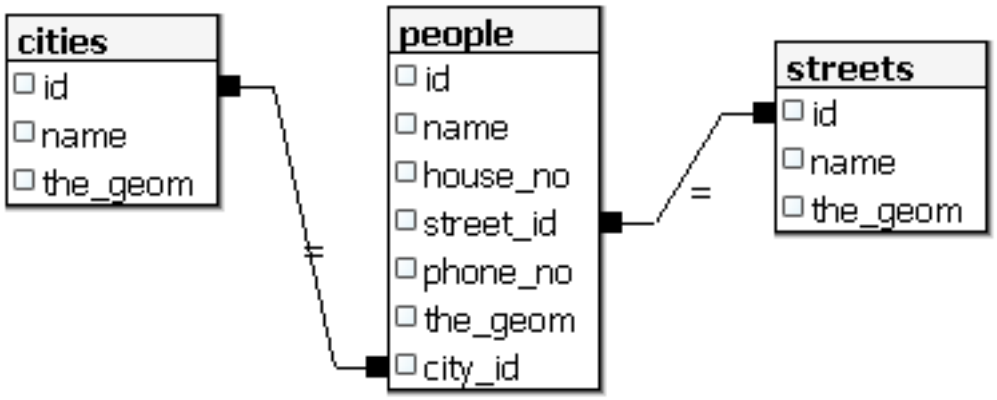
If you're getting the following error message:

```
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 Unser Schema

Aktuell sollte unser Schema wie folgt aussehen:



16.5.6 Try Yourself

Erstellen Sie Stadtgrenzen indem Sie die minimale konvexe Hülle um alle Adressen einer Stadt und einen Puffer darum ermitteln.

16.5.7 Zugriff auf Unter-Objekte

Die SFS-Model Funktionen bieten eine große Bandbreite an Optionen zum Zugriff auf Sub-Objekte der SFS Geometrien. Wenn man den ersten Stützpunkt aller Polygone in der Tabelle myPolygonTable selektieren möchte, geht man wie folgt vor:

- Umwandeln der Polygongrenze in eine Linie:

```
select st_boundary(geometry) from myPolygonTable;
```

- Auswahl des ersten Stützpunktes der resultierenden Linie:

```
select st_startpoint(myGeometry)
from (
    select st_boundary(geometry) as myGeometry
    from myPolygonTable) as foo;
```

16.5.8 Datenverarbeitung

PostGIS unterstützt alle zum OGC SFS/MM Standard konforme Funktionen. Alle diese Funktionen starten mit `ST_`.

16.5.9 Ausschneiden

Um einen Teil unsrer Daten auszuschneiden, können wir die `ST_INTERSECT()` Funktion verwenden. 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));
```

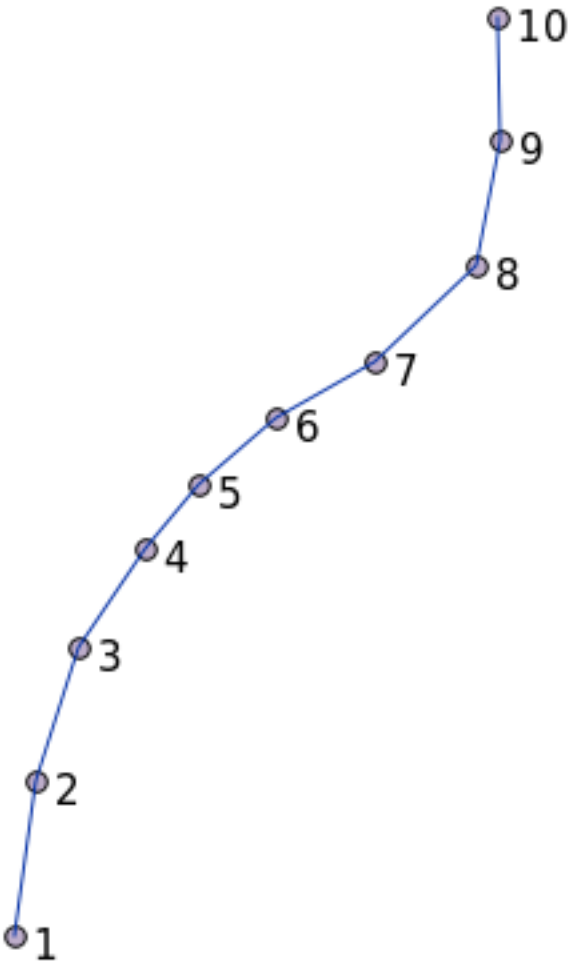
16.5.10 Geometrien aus anderen Geometrien erstellen

Man möchte aus einer gegebenen Tabelle mit Punkten eine Linie generieren. Die Ordnung der Punkte ist durch ihre `id` vorgegeben. Eine andere Methode zur Ordnung könnte ein Zeitstempel sein, den man z.B. bei der Aufzeichnung von Wegpunkten im GPS-Empfänger erhält.

Um eine Linie aus einem neuen Punktlayer ‚points' zu erstellen, kann man das folgende Kommando ausführen:

```
select ST_LineFromMultiPoint(st_collect(the_geom)), 1 as id
from (
  select the_geom
  from points
  order by id
) as foo;
```

Um die Arbeitsweise zu testen ohne einen neuen Layer zu erstellen, kann man das Kommando auf den ‚people' Layer anwenden, obwohl das kein wirklich sinnvolles Ergebnis liefert.



16.5.11 Geometriebereinigung

You can get more information for this topic in this blog entry.

16.5.12 Unterschiede zwischen Tabellen

Um den Unterschied zwischen zwei Tabellen mit derselben Struktur herauszufinden, kann man das PostgreSQL Schlüsselwort `EXCEPT` verwenden:

```
select * from table_a
except
select * from table_b;
```

Im Ergebnis erhält man alle Datensätze aus table_a die nicht in table_b gespeichert sind.

16.5.13 Tablespaces

Man kann durch die Erstellung von Tablespaces vorgeben, wo Postgres seine Daten speichern soll:

```
CREATE TABLESPACE homespace LOCATION '/home/pg';
```

Wenn man eine Datenbank erstellt, kann man vorgeben welcher Tablespace genutzt werden soll, z.B.:

```
createdb --tablespace=homespace t4a
```

16.5.14 In Conclusion

Sie haben gelernt, wie man komplexere Geometrien mit Hilfe von PostGIS Anweisungen erstellt. Dies dient vor allem der Verbesserung Ihres Hintergrundwissens bei der Arbeit mit räumlichen Datenbanken über eine GIS Oberfläche. Normalerweise müssen Sie diese Anweisungen nicht manuell eingeben. Trotzdem hilft ein generelles Verständnis ihrer Struktur bei der Nutzung von GIS, insbesondere wenn Fehler auftreten, die ohne dieses Wissen unverständlich erscheinen.

KAPITEL 17

Der QGIS-Verarbeitungsleitfaden

Dieses Modul wurde von Victor Olaya und Paolo Cavallini beigetragen.

Inhalt:

17.1 Einleitung

Diese Anleitung beschreibt die Nutzung des QGIS Verarbeitungs Frameworks. Es werden keine Kenntnisse des Verarbeitungsframeworks oder anderer damit in Verbindung stehender Anwendungen benötigt. Es wird davon ausgegangen, dass Grundlagenwissen zu QGIS vorhanden ist. Die Kapitel zur Skriptverarbeitung gehen von grundlegendem Wissen zu Python und der QGIS Python API aus.

Dieser Leitfaden ist für das Selbststudium oder zur Durchführung eines Workshops zur Verarbeitung konzipiert.

Die Beispiele in dieser Anleitung bezeihen sich auf QGIS 3.4. Sie funktionieren unter Umständen nicht mit anderen Verisonen von QGIS.

Diese Anleitung beeinhaltet einige Übungen, der Komplexität sich schrittweise erhöht. Wenn Sie das Verarbeitungsframework noch nicht verwendet haben, sollten Sie am Anfang beginnen. Wenn Sie schon Erfahrungen haben, können Sie auch Übungen überspringen. Sie sind in der Regel unabhängig voneinander. Jede Übung führt in ein neues Konzept oder neues Element ein, das im Titel und der kurzen Einführung des Abschnitts genannt wird. Das vereinfacht die Suche nach Übungen mit einem bestimmten Inhalt.

Um eine systematischere Beschreibung aller Komponenten des Frameworks zu erhalten, ist es empfehlenswert, sich die entsprechenden Kapitel im Benutzerhandbuch anzusehen. Nutzen Sie es als Unterstützung zusammen mit dieser Anleitung.

Alle Übungen in dieser Anleitung nutzen denselben freien Datensatz wie im gesamten Training Handbuch. Den Datensatz findet man unter *Data*. Die Zip-Datei enthält verschiedene zu den jeweiligen Lektionen gehöhrende Ordner. In jedem Ordner findet man eine QGIS Projektdatei. Öffnen Sie zu Beginn einer Lektion einfach die dazu gehörende Projektdatei.

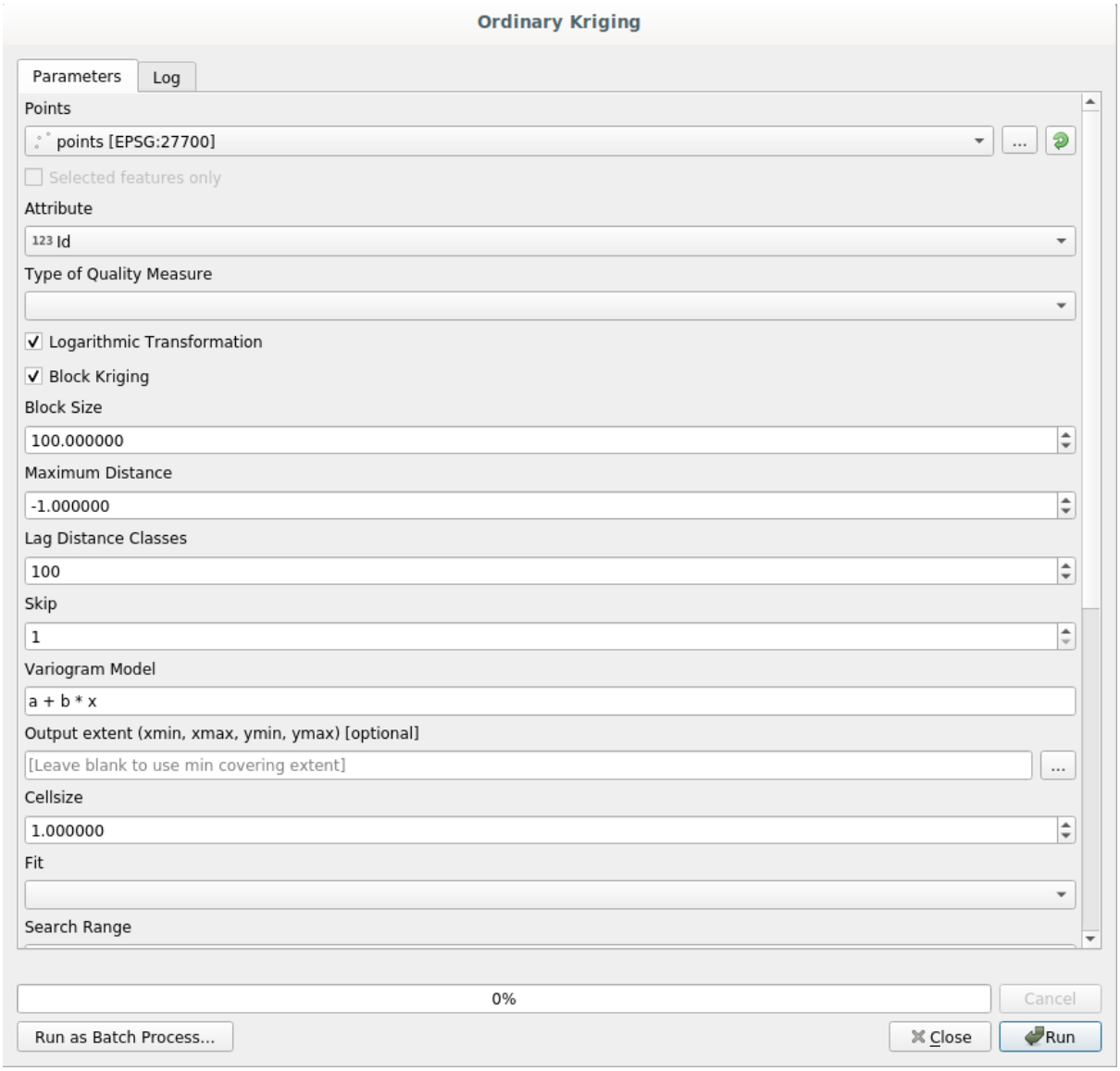
Viel Spaß!

17.2 Eine wichtige Warnung zu Beginn

Genauso wie das Handbuch einer Textverarbeitung Ihnen nicht beibringt, wie man eine Novelle oder ein Gedicht schreibt und ein CAD Tutorial Ihnen nicht zeigt wie man die Größe eines Gebäudeträgers berechnet, so lehrt diese Anleitung auch nicht räumliche Analyse. Statt dessen zeigt sie, wie man das QGIS Prozessierungsframework als mächtiges Werkzeug zur Durchführung räumlicher Analysen verwenden kann. Sie sollten sich die erforderlichen Kenntnisse zum Verständnis dieser Art von Analysen aneignen. Ohne dem macht es wenig Sinn, das Framework und seine Algorithmen zu verwenden, auch wenn es verlockend ist.

Lassen Sie uns das an einem Beispiel demonstrieren.

Gegeben sind eine Menge an Punkten mit einem bestimmten Wert für eine Variable an jedem Punkt. Man kann aus diesen Werten mit Hilfe des *Kriging* Geoalgorithmus einen Rasterlayer erstellen. Das Dialogfenster mit den Eingabeparametern des Moduls sieht in etwa so aus.



Das sieht komplex aus, oder?

Mit dem Lesen dieses Handbuches, lernen Sie, wie man dieses Modul benutzt, wie man es innerhalb einer Stapelverarbeitung verwendet, um Raster Layer aus hunderten von Punkten in einem Durchlauf zu erstellen oder was passiert, wenn im Eingabelayer einige Punkte ausgewählt sind. Die Parameter an sich werden aber nicht erklärt. Ein erfahrener Analyst mit guter Kenntnis der Geostatistik wird keine Probleme haben, die Parameter zu verstehen. Wenn

Sie keine Erfahrung damit haben und mit Begriffen wie *sill*, *range* oder *nugget* nicht vertraut sind, sollten Sie das Modul *Kriging* nicht verwenden. Es beinhaltet weitere Konzepte wie räumliche Autokorrelation oder Semivariogramme, die Ihnen dann vermutlich auch nicht bekannt oder nicht vertraut genug sind. Sie sollten diese zuerst studieren, um sie danach in QGIS anzuwenden. Wenn man dies ignoriert, erhält man falsche Ergebnisse und dürftige (und wahrscheinlich nutzlose) Analyseergebnisse.

Auch wenn nicht alle Algorithmen so komplex sind wie kriging (andererseits gibt es auch noch komplexere!), erfordern doch fast alle, die zu Grunde liegenden Ideen der verwendeten Analysen zu verstehen. Die Anwendung ohne dieses Wissen führt sehr wahrscheinlich zu schlechten Ergebnissen.

Geoalgorithmen ohne eine gutes Grundlagenwissen über räumliche Analysen einzusetzen, ist wie der Versuch einen Roman zu schreiben ohne irgendwelche Kenntnisse zu Grammatik, Rechtschreibung oder Geschichtenerzählen zu haben. Sie erhalten vielleicht ein Ergebnis, aber es wird vermutlich wertlos sein. Täuschen Sie sich bitte nicht selbst, indem Sie denken, dass Sie nur nach dem Lesen dieser Anleitung bereit sind, räumliche Analysen durchzuführen und korrekte Ergebnisse zu erhalten.

Hier ist ein guter Literaturhinweis, um mehr über die Analyse räumlicher Daten zu lernen.

Geospatial Analysis (3rd Edition): A Comprehensive Guide to Principles, Techniques and Software Tools Michael John De Smith, Michael F. Goodchild, Paul A. Longley

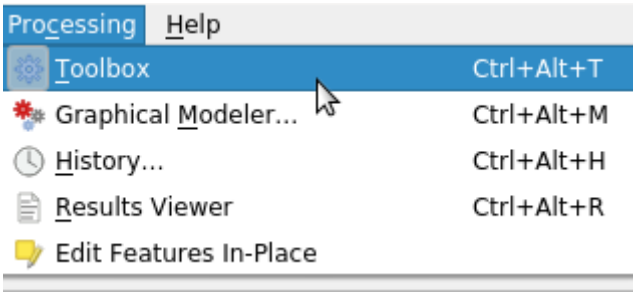
Die online Quelle ist: here

17.3 Einrichtung der Verarbeitungsumgebung

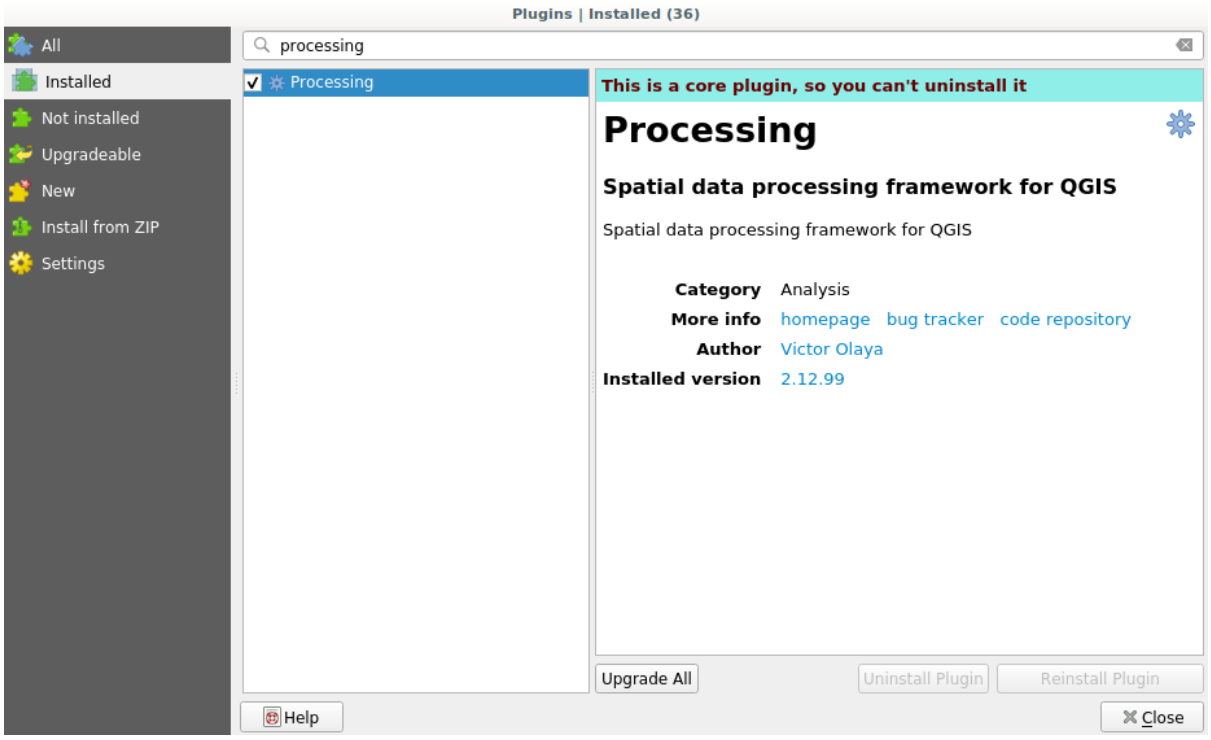
Als Erstes muss die Verarbeitungsumgebung konfiguriert werden. Da nicht viele Sachen einzustellen sind, ist das ganz einfach.

Später werden wir sehen, wie man externe Anwendungen zur Erweiterung der vorhandenen Algorithmen konfiguriert. Zuerst arbeiten wir aber nur mit der Umgebung.

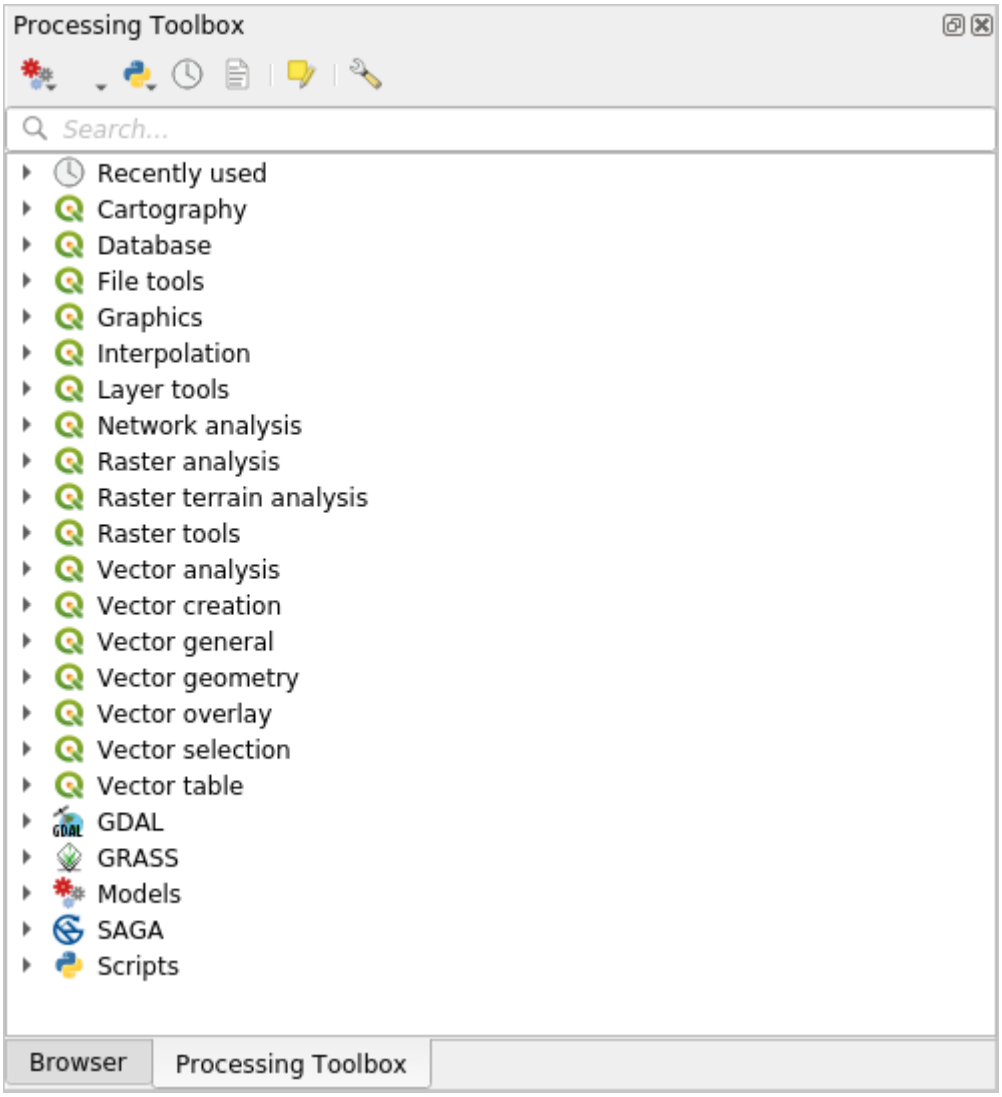
Die Verarbeitungsumgebung ist eine Standard-QGIS-Erweiterung. D.h. sie ist in QGIS enthalten und sollte in Ihrem System schon zur Verfügung stehen. Wenn sie aktiv ist, sollten Sie einen Menüpunkt *Verarbeitung* in Ihrer Menüleiste sehen. Dort können Sie auf alle Komponenten der Umgebung zugreifen.



Wenn Sie dieses Menü nicht sehen, müssen Sie die Erweiterung im Erweiterungsmanager aktivieren.



Am meisten werden wir mit den Verarbeitungswerkzeugen arbeiten. Klicken sie auf den entsprechenden Menüeintrag. Die Verabeitungswerkzeuge werden dann auf der rechen Seite des QGIS-Fensters angezeigt.



Die Verarbeitungswerkzeuge enthalten eine Liste aller verfügbaren Algorithmen, unterteilt in Gruppen die auch *Anbieter* genannt werden. Diese Anbieter können unter *Einstellungen* ► *Optionen* ► *Verarbeitung* aktiviert oder deaktiviert werden. Wir werden später in dieser Anleitung darauf eingehen.

Als Voreinstellung sind nur Anbieter aktiv, die nicht von Drittherstellern abhängig sind (d.h. Anbieter, die nur QGIS-Elemente voraussetzen). Bei Algorithmen, die externe Anwendungen benötigen, ist unter Umständen eine zusätzliche Konfiguration erforderlich. Die Konfiguration von Erweiterungen wird in einem späteren Kapitel in diesem Handbuch beschrieben.

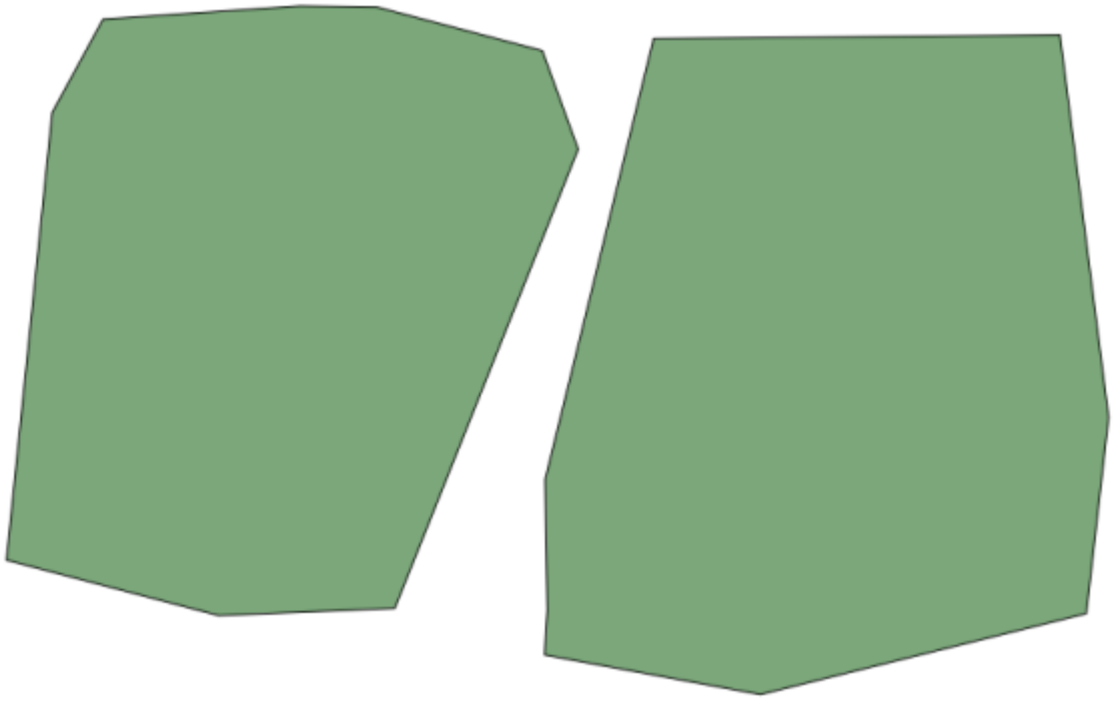
Wenn Sie bis hier gekommen sind, können Sie mit der Nutzung der Geoalgorithmen beginnen. Es muss nichts weiter konfiguriert werden. Wir werden unseren ersten Algorithmus in der nächsten Lektion starten.

17.4 Ausführung unseres ersten Algorithmus. Der Werkzeugkasten

Bemerkung: In dieser Lektion werden wir unseren ersten Algorithmus starten und schon erste Ergebnisse erhalten.

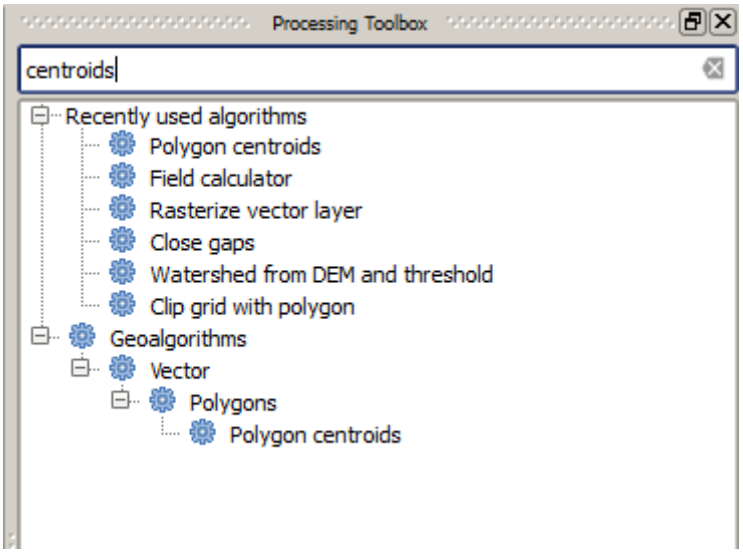
Wie schon erwähnt, kan die Verarbeitungsumgebung Algorithmen aus anderen Anwendungen ausführen. Sie enthält aber auch zugehörige Algorithmen, die zur Ausführung keine externe Software benötigen. Zur Erkundung der Verarbeitungsumgebung werden wir einen dieser enthaltenen Algorithmen starten. Genauer gesagt, wir werden die Schwerpunkte einer Reihe von Polygonen ermitteln.

Öffnen Sie zuerst das QGIS-Projekt für diese Lektion. Es enthält nur einen Layer mit zwei Polygonen

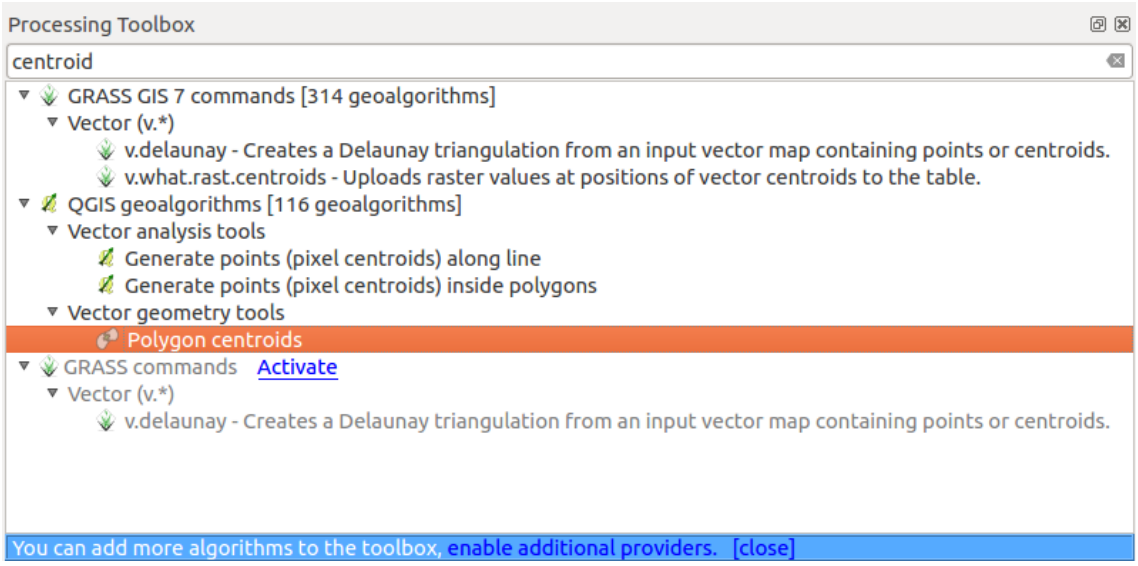


Gehen Sie nun zur Textzeile oben in den Verarbeitungswerkzeugen. Es handelt sich um ein Suchfeld. Wenn Sie Text eingeben, wird eine Liste mit Algorithmen gefiltert, die diesen Text enthalten. Bei Algorithmen, die zu Ihrer Suche passen aber zu einem inaktiven Anbieter gehören, wird zusätzlich eine Beschriftung im unteren Teil der Verarbeitungswerkzeuge angezeigt.

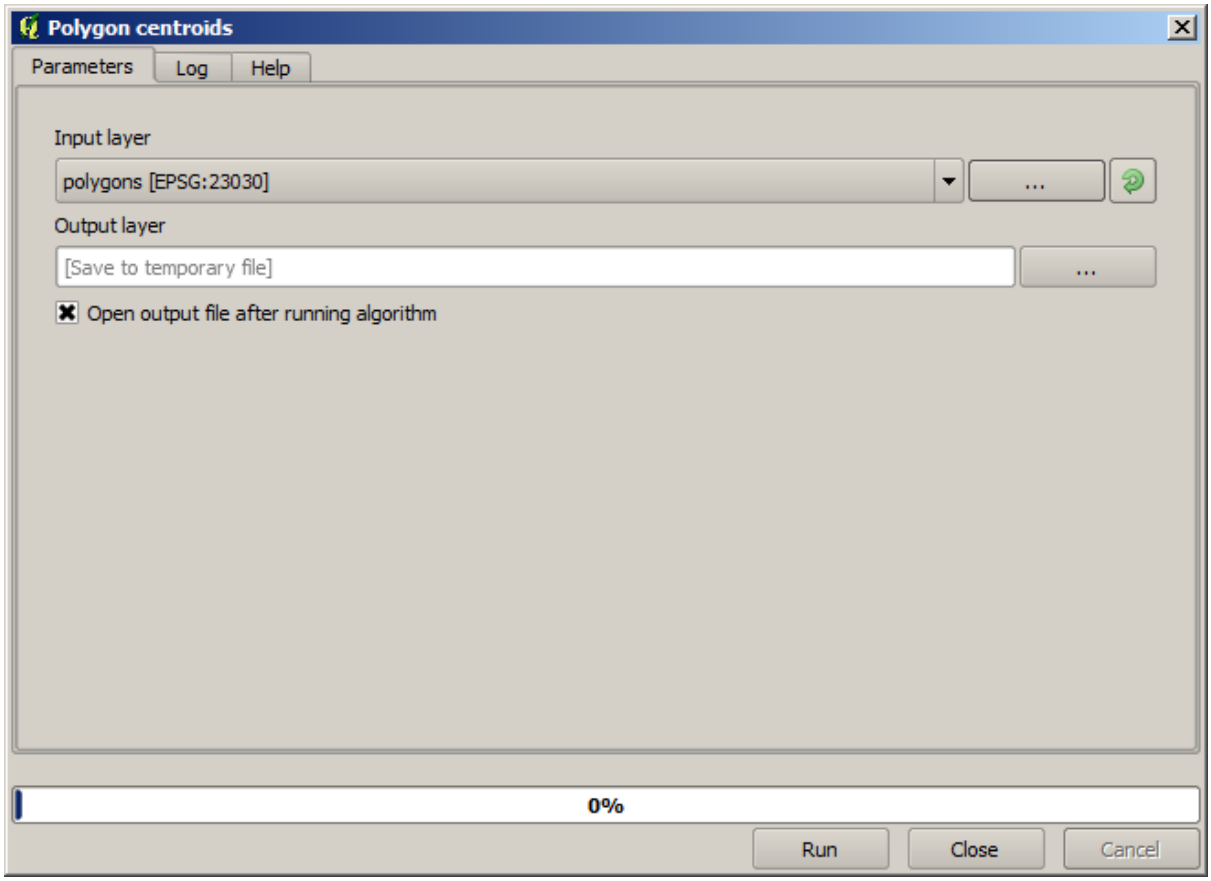
Geben Sie `centroids` ein und Sie sollten in etwa das Folgende sehen.



Das Suchfeld ist ein praktischer Weg, um einen gesuchten Algorithmus zu finden. Im unteren Teil des Dialogfensters zeigt eine zusätzliche Beschriftung, dass Algorithmen zu Ihrer Suche passen aber zu einem inaktiven Anbieter gehören. Wenn Sie auf den Link in der Beschriftung klicken, werden die Algorithmen der inaktiven Anbieter hellgrau hinterlegt angezeigt. Außerdem wird ein Link zur Aktivierung der inaktiven Anbieter angezeigt. Wir werden später sehen, wie man andere Anbieter aktiviert.



Um einen Algortihmus auszuführen, muss man nur doppelt auf den Eintrag in den Verarbeitungswerkzeugen klicken. Wenn Sie doppelt auf den Algorithmus *Polygon centroids* klicken, sehen Sie das folgende Dialogfenster.



Alle Algorithmen verwenden eine vergleichbare Benutzerschnittstelle mit Eingabeparametern, die man ausfüllen muss und einer Ausgabe für die der Speicherort gewählt werden muss. In diesem Fall ist die einzige Eingabe ein Vektorlayer mit Polygonen.

Wählen Sie den Layer *Polygons* als Eingabelayer. Es gibt zwei Optionen zur Speicherung des Ergebnisses: als Datei auf einem Laufwerk oder als temporäre Datei

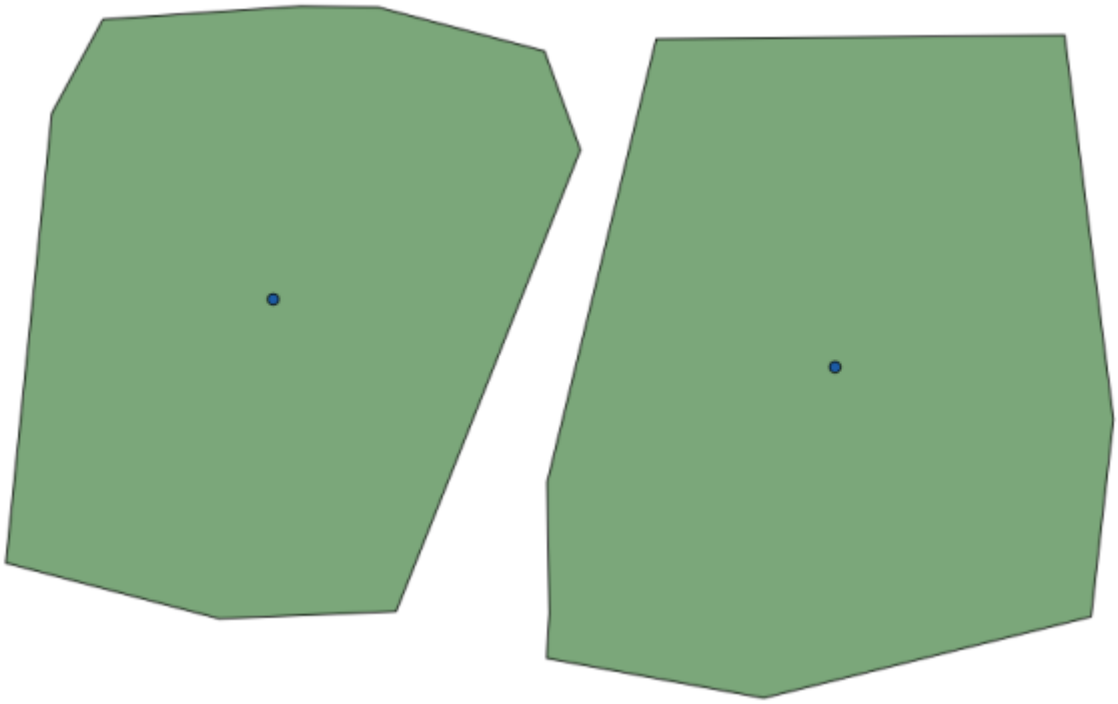
Wenn Sie eine Ausgabe vorgeben wollen und das Resultat nicht in einer temporären Datei speichern wollen, wird das Dateiformat durch den Dateityp der Ausgabedatei vorgegeben. Um ein Format zu auszuwählen, wählt man einfach den zugehörigen Dateityp (oder Sie fügen ihn bei der Eingabe des Pfades per Hand ein). Wenn der vergebene Dateityp zu keinem bekannten Dateityp passt, wird ein Standardtyp vergeben (normalerweise `.dbf` für Tabellen, `.tif` for Rasterlayer und `.shp` für Vektorlayer) und der Layer dann im dazu gehörenden Dateiformat gespeichert.

In allen Übungen in dieser Anleitung speichern wir die Ergebnisse nur temporär ab, da sie später nicht mehr benötigt werden. Sie können die Ergebnisse natürlich auch zusätzlich als Datei auf einem Laufwerk speichern.

Warnung: Temporäre Dateien werden gelöscht, wenn man QGIS beendet. Wenn Sie ein Projekt erstellen, das eine temporäre Ausgabe enthält, wird QGIS sich beim erneuten Öffnen des Projektes beschweren, da die Ausgabedatei nicht mehr existiert.

Nachdem Sie die Einstellung im Dialogfenster des Algorithmus vorgenommen haben, klicken Sie zum Starten auf *Starte*.

Sie erhalten die folgende Ausgabe.



Die Ausgabe hat dasselbe KBS wie die Eingabe. Die Geoalgorithmen gehen davon aus, dass alle Eingabelayer dasselbe KBS verwenden. Sie führen keine Reprojektion durch. Bis auf einige wenige Algorithmen (z.B. zur Reprojizierung) haben die Ausgaben dasselbe KBS. Wir werden bald mehr dazu erfahren.

Versuchen Sie selbst, andere Dateiformate zu verwenden (z.B. `shp` und `geojson` als Dateitypen). Wenn Sie nicht wollen, dass der Layer nach Abschluss der Verarbeitung in QGIS geladen wird, können Sie den Haken unterhalb des Ausgabepfades entfernen.

17.5 Mehr Algorithmen und Datentypen

Bemerkung: In dieser Lektion werden wir noch drei weitere Algorithmen ausführen. Wir werden lernen wie man andere Eingabetypen verwendet und Ausgaben so konfiguriert, dass sie automatisch in einen vorgegebenen Ordner gespeichert werden.

Für diese Übung benötigen wir eine Tabelle und einen Polygonlayer. Wir werden einen Punktlayer basierend auf Koordinaten in der Tabelle erstellen und dann die Anzahl der Punkte in jedem Polygon zählen. Wenn Sie das QGIS-Projekt zu dieser Lektion öffnen, finden Sie eine Tabelle mit X und Y Koordinaten aber keinen Polygonlayer. Keine Angst, wir werden den Layer noch mit einem Geoalgorithmus erstellen.

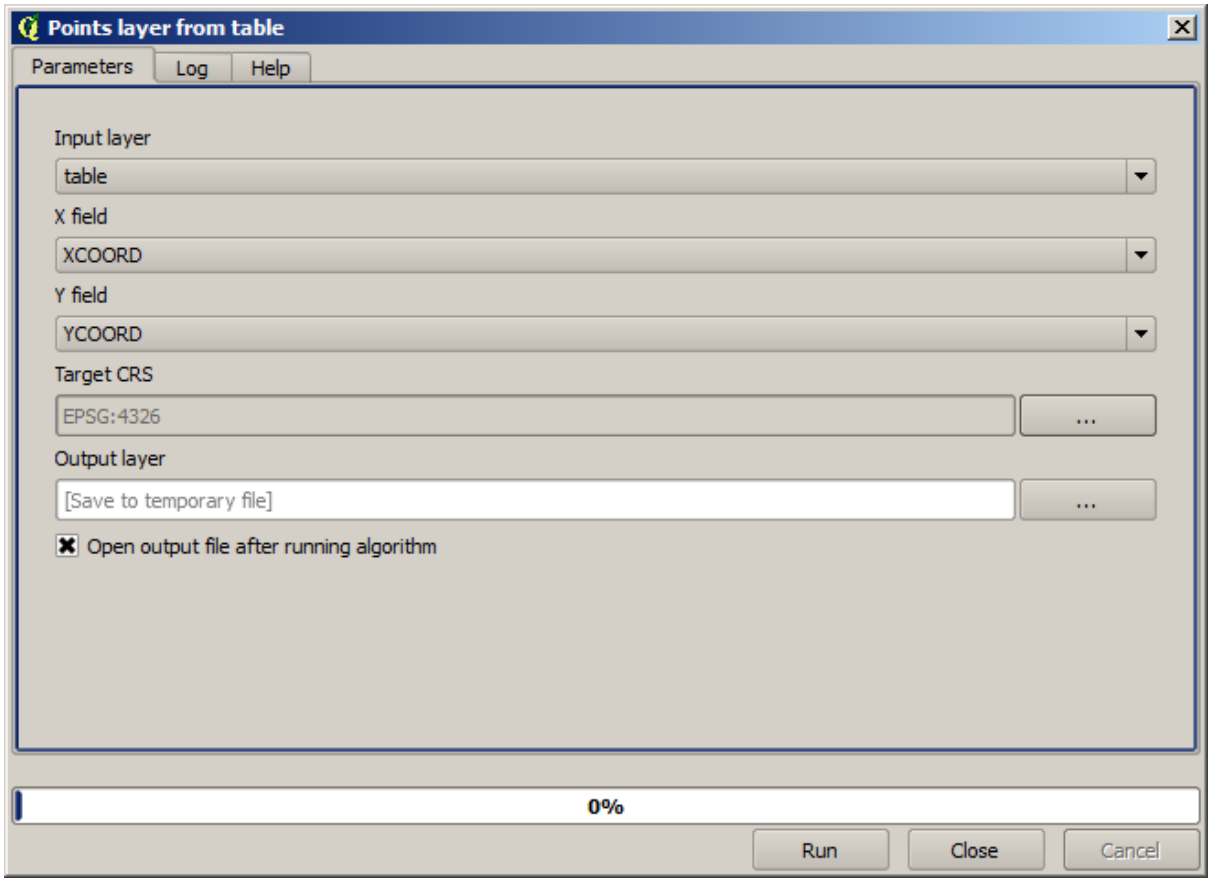
Als Erstes werden wir mit Hilfe des Algorithmus *Punktlayer aus Tabelle erzeugen* einen Punktlayer aus den Koordinaten in der Tabelle erzeugen. Sie wissen schon wie man das Suchfeld benutzt, es sollte also nicht schwer sein, den Algorithmus zu finden. Klicken Sie doppelt darauf und es erscheint der folgende Dialog.

Der Algorithmus generiert, ähnlich wie in der letzten Lektion, genau eine Ausgabedatei. Er benötigt drei Eingaben:

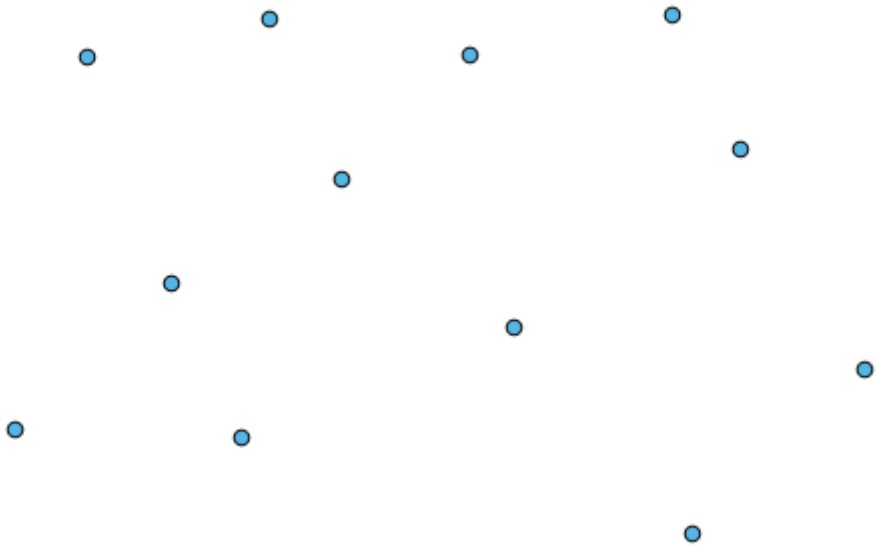
- *Eingabelayer*: die Tabelle mit den Koordinaten. Sie sollten hier die Tabelle aus dem Übungsordner verwenden.
- *X und Y Felder*: diese zwei Parameter gehören zum ersten Eintrag. Das Auswahlfeld zeigt die Felder an, die in der gewählten Tabelle vorhanden sind. Wählen Sie das Feld *XCOORD* für den Parameter *X-Feld* und das Feld *YCOORD* für den Parameter *Y-Feld*.
- *Ziel KBS*: Da der Algorithmus keine Eingabelayer verwendet, kann kein KBS auf Basis der Eingabe vergeben werden. Statt dessen muss das KBS der Koordinaten in der Tabelle manuell ausgewählt werden. Klicken Sie auf

das Auswahlfeld, um die QGIS KBS Auswahl zu öffnen. Wählen Sie EPSG:4326 als Ziel KBS. Wir benutzen dieses KBS, da die Koordinaten in der Tabelle in diesem KBS vorliegen.

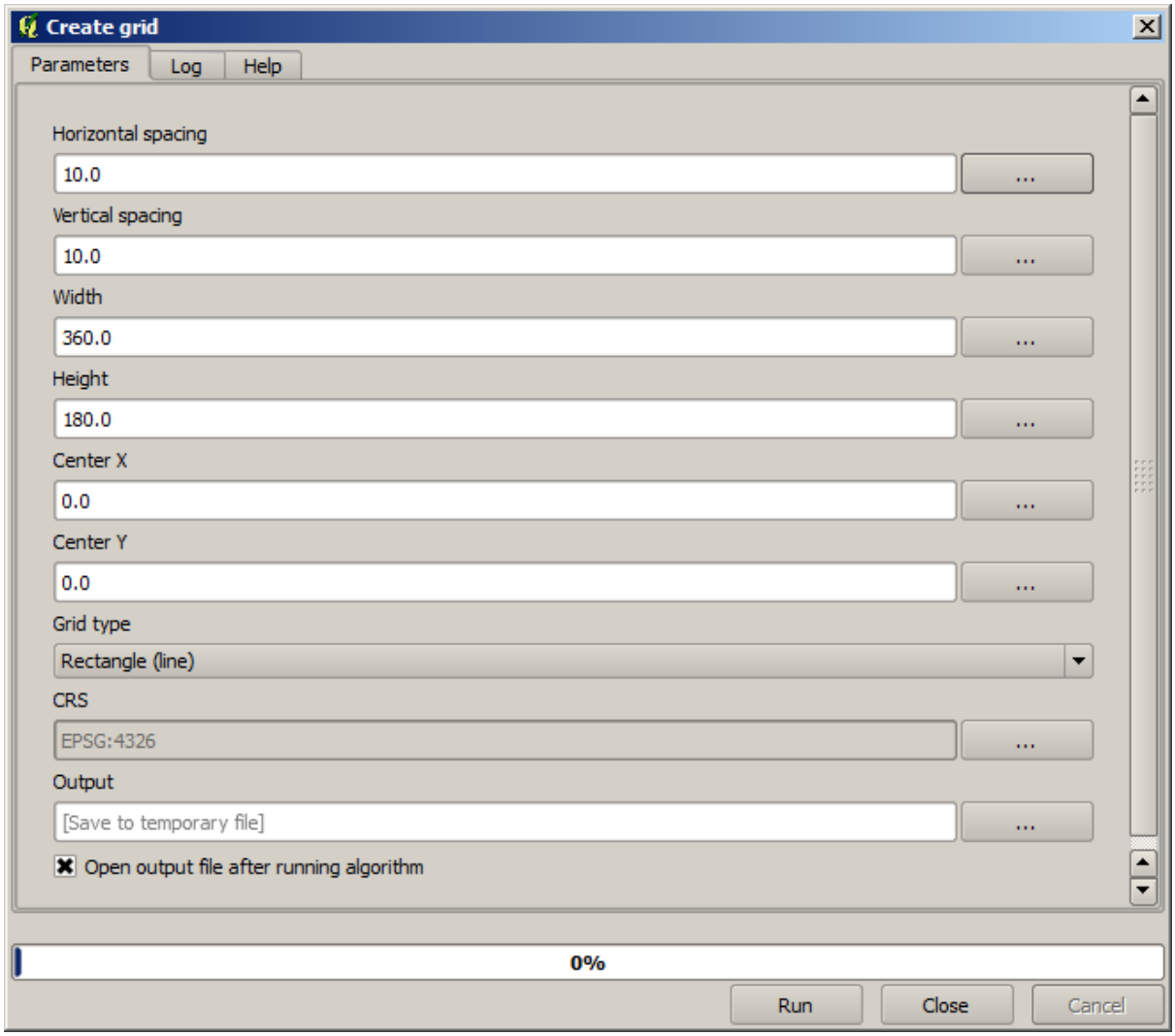
Der Dialog sollte in etwa so aussehen.



Klicken Sie jetzt auf die *Starte* Schaltfläche, um den folgenden Layer zu erhalten (Sie müssen unter Umständen die Zoomstufe ändern, um alle neu erstellten Punkte zu sehen):

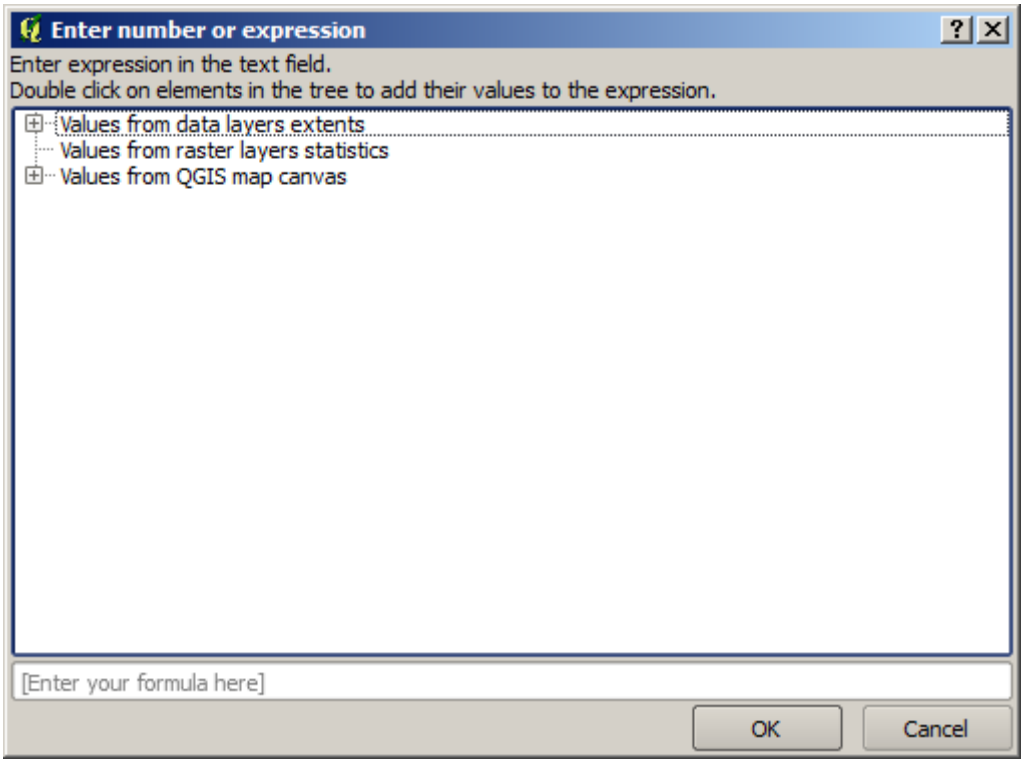


Als nächstes benötigen wir den Polygonlayer. Wir werden mit dem Algoritumus *Gitter erzeugen* ein regelmäßiges Polygongitter erzeugen. Im folgenden Dialog sind die Algorithmus Parameter enthalten.



Warnung: Die Optionen sind in neueren Versionen von QGIS einfacher; man muss nur die Minimal- und Maximalwerte für X und Y eingeben (vorgeschlagene Werte: -5.696226,-5.695122,40.24742,40.248171)

Die zur Erstellung des Gitters erforderlichen Eingaben sind alle Zahlen. Wenn man einen numerischen Wert eingibt, hat zwei Möglichkeiten: die direkte Eingabe in das Eingabefeld oder der Knopf auf der rechten Seite der den folgenden Dialog öffnet.



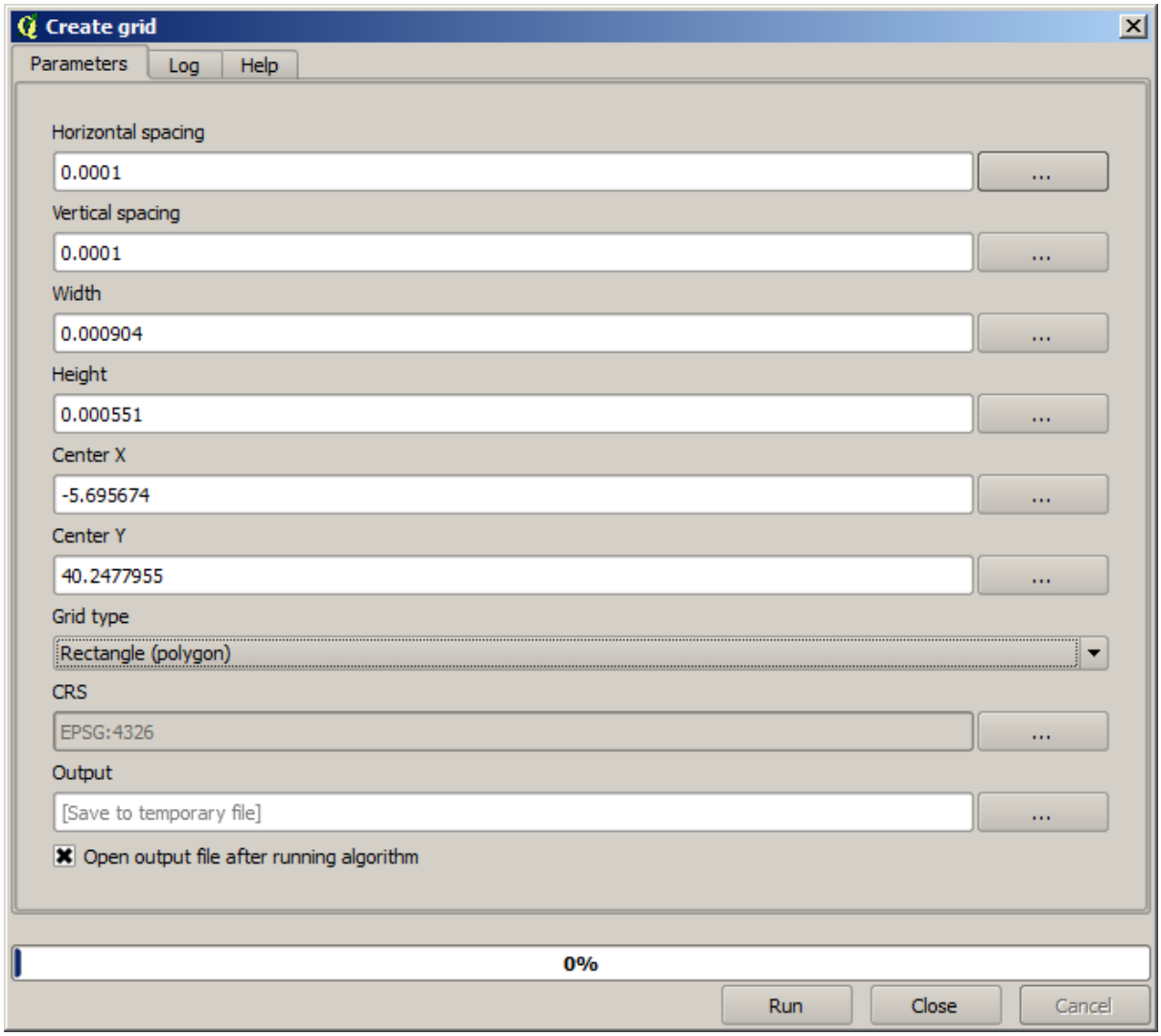
Der Dialog enthält einen einfachen Rechner, mit dem Sie Ausdrücke wie `11 * 34,7 + 4,6` eingeben können, das Ergebnis wird berechnet und in das zugehörige Feld des Parameter Dialogs eingetragen. Es sind außerdem Konstanten und Werte aus anderen Layer verfügbar, die verwendet werden können.

In diesem Fall wollen wir ein Gitter erstellen, das die Ausdehnung des Eingabepunktlayers umfasst. Wir müssen die Koordinaten des Punktlayers verwenden, um die Koordinaten des Zentrums des Gitters und seine Breite und Höhe zu bestimmen. Versuchen Sie es selbst, mit ein bisschen Mathematik, der Hilfe des Rechners und der Konstanten aus dem Eingabepunktlayer.

Wählen Sie *Rechteck (Polygon)* im Feld *Gittertyp*.

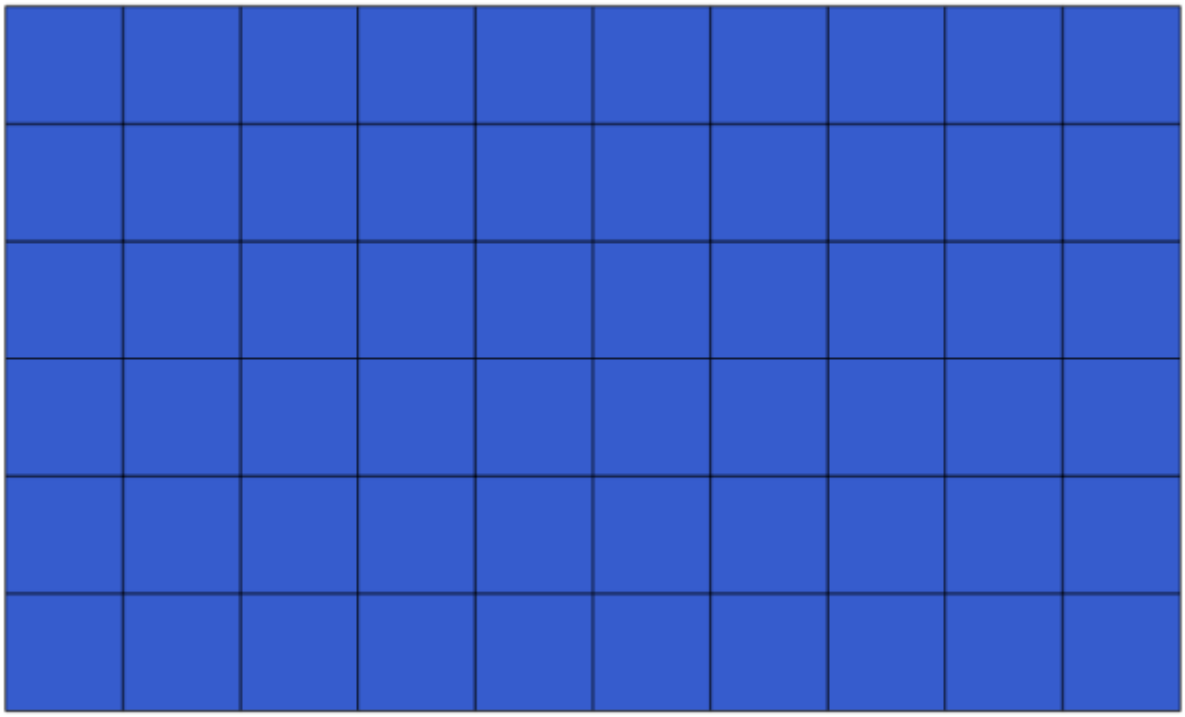
Wie beim letzten Algorithmus müssen wir auch das KBS angeben. Wählen sie wie vorher EPSG:4326 als Ziel KBS aus.

Am Ende sollte der Parameterdialog in etwa so aussehen:

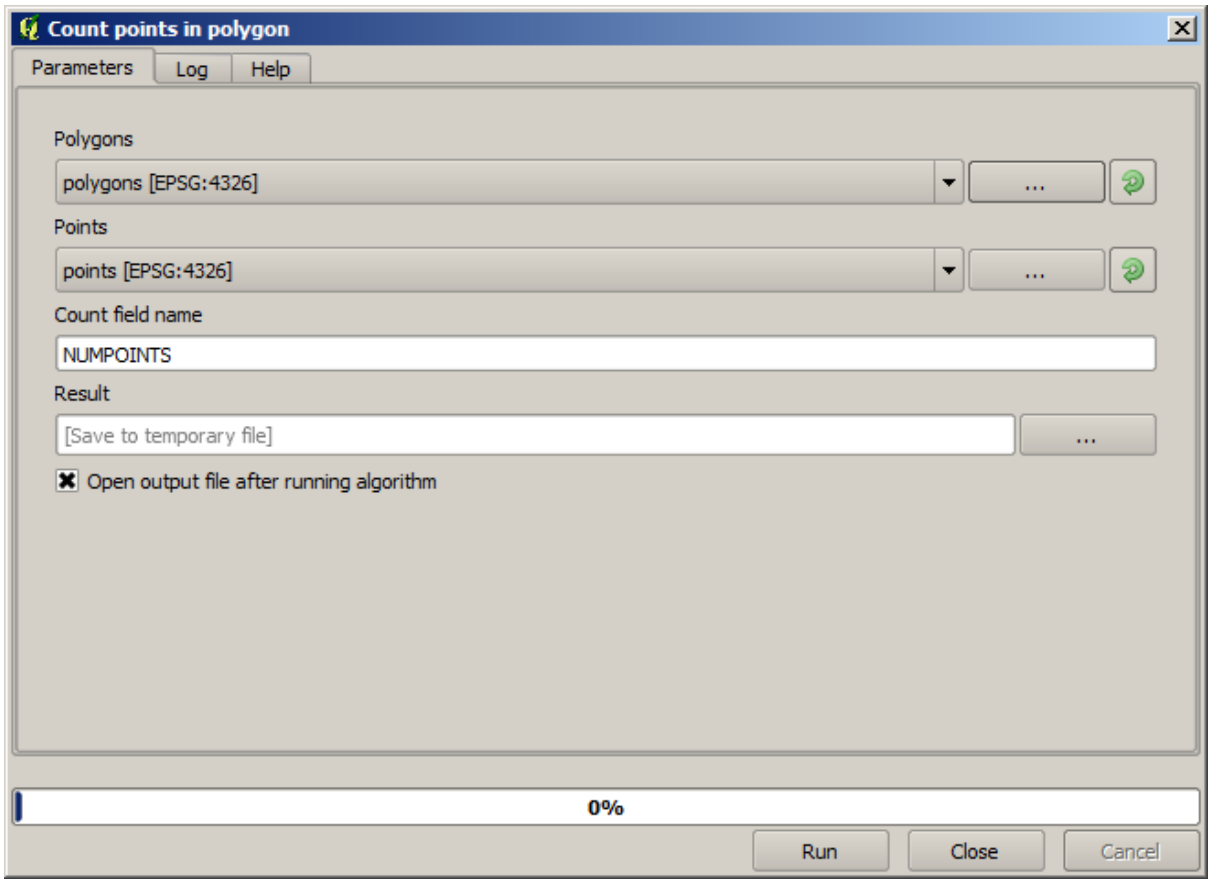


(Es ist besser einen Abstand bei Breite und Höhe hinzuzufügen: Horizontaler Abstand: 0,0001, Vertikaler Abstand: 0,0001, Breite: 0,001004, Höhe: 0,000651, Zentrum X: -5,695674, Zentrum Y: 40,2477955) Das X Zentrum ist ein bisschen knifflig: -5,696126+((-5,695222+ 5,696126)/2)

Drücken Sie *Starte* und Sie erhalten den Gitterlayer.

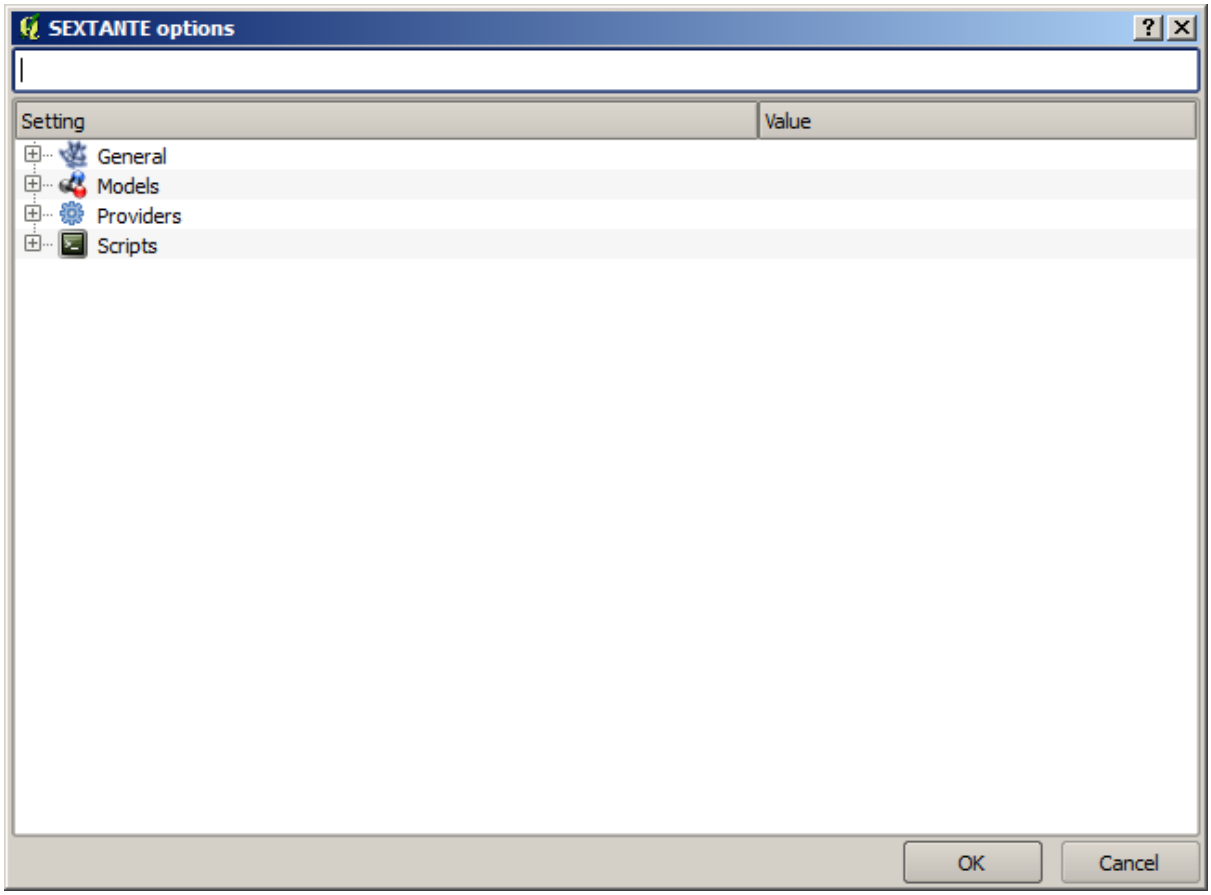


Der letzte Schritt ist das Zählen der Punkte in jedem Rechteck des Gitternetzes. Wir werden dazu den Algorithmus *Punkte in Polygon zählen* verwenden.

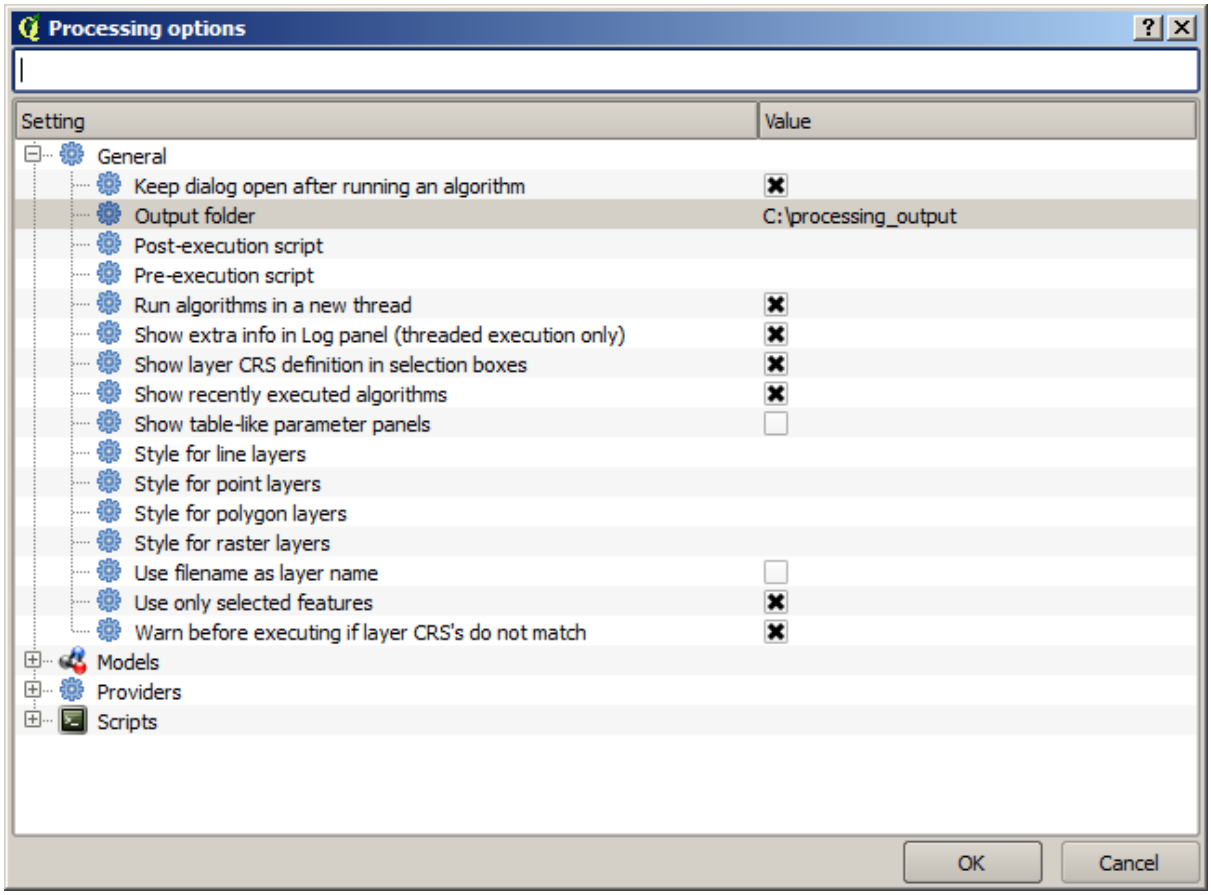


Damit haben wir das gewünschte Ergebnis erreicht.

Bevor wie die Lektion beenden, folgt noch ein kleiner Tipp, der das dauerhafte Speichern von Daten vereinfacht. Wenn Sie alle Ausgabedateien in einem gegebenen Ordner speichern wollen, müssen Sie den Ordnernamen nicht jedes Mal neu eingeben. Gehen Sie statt dessen zur Schaltfläche *Optionen* der Verarbeitungswerkzeuge. Damit wird der Einstellungsdialog geöffnet.



In der Gruppe *Allgemein* finden Sie den Eintrag *Ausgabeverzeichnis*. Geben Sie hier den Ordner Ihrer Ausgabedateien an.



Wenn Sie nun einen Algortihmus starten, reicht der Dateiname anstelle des gesamten Pfades aus. Im obigen Beispiel würde die Eingabe von `graticule.shp` als Ausgabepfad dazu führen, dass das Ergebnis unter `D:\processing_output\graticule.shp` gespeichert werden würde. Man kann weiterhin einen vollständigen Pfad verwenden, wenn man das Ergebnis in einem anderen Ordner speichern möchte.

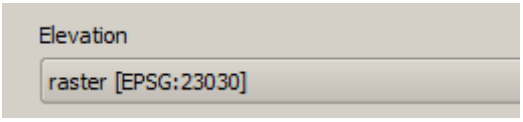
Probieren Sie den *Gitter erzeugen* Algorithmus selbst weiter aus. Verwenden Sie verschiedene Gittergrößen und Gitterarten.

17.6 KBS - reprojizieren

Bemerkung: In dieser Lektion werden wir diskutieren, wie die Verarbeitungswerkzeuge KBS verwenden. Wir werden auch einen sehr nützlichen Algorithmus kennenlernen: reprojizieren.

KBS führen bei Nutzern der QGIS Verarbeitungswerkzeuge oft zu Verwirrung. Hier folgen einige generelle Regeln, wie KBS durch Geoalgorithmen bei der Erstellung neuer Layer behandelt werden.

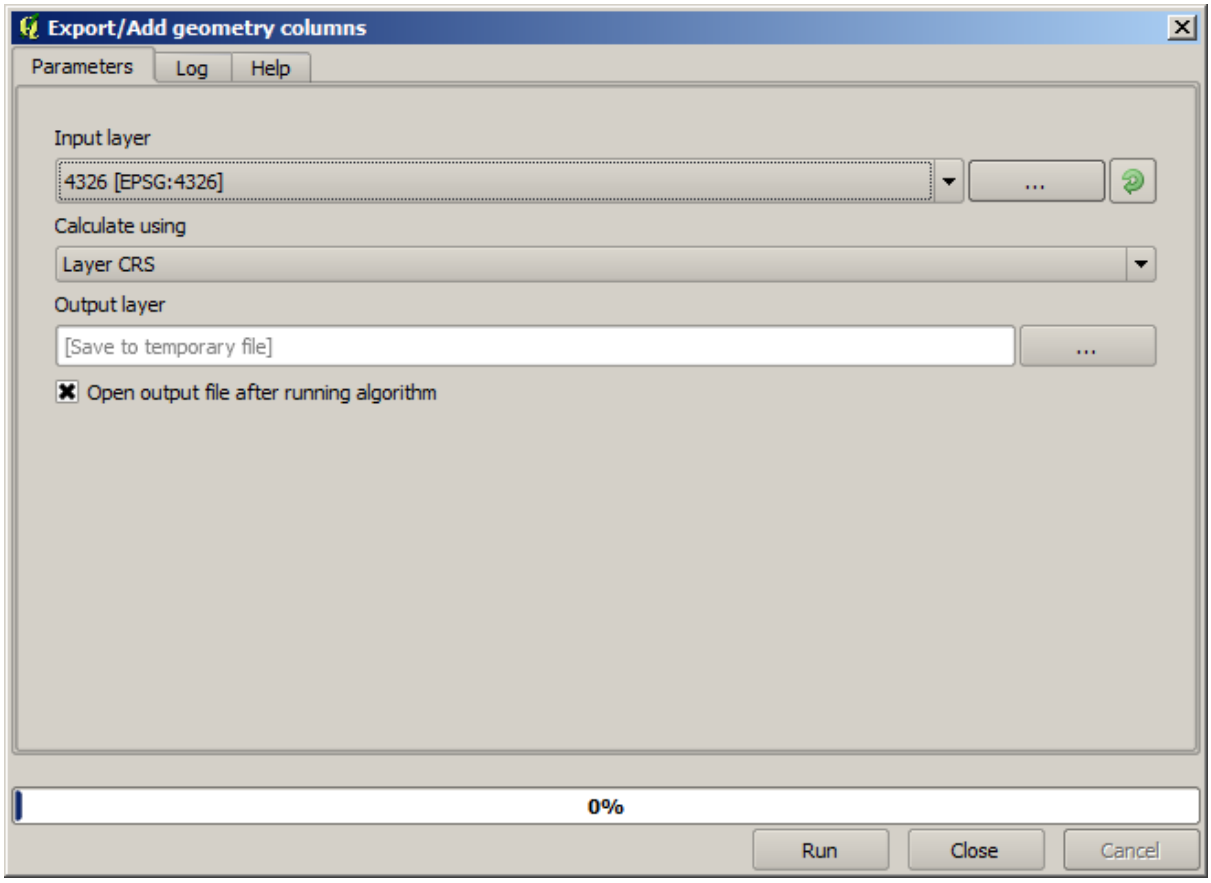
- Wenn Eingabelayer vorhanden sind, wird das KBS des ersten Layers verwendet. Es wird angenommen, dass die anderen Layer - so wie es sein sollte - dasselbe KBS verwenden. Wenn Sie Layer mit verschiedenen KBS verwenden, wird QGIS dazu eine Warnung ausgeben. Achten Sie darauf, dass das KBS der Eingabelayer zusammen mit dem Layernamen im Parameter Dialog angezeigt wird.



- Wenn kein Eingabelayer vorhanden ist, wird das KBS des Projektes verwendet, es sei denn der Algorithmus enthält ein explizites KBS Feld (so wie in der letzten Lektion beim Algorithmus zur Erzeugung des Gitternetzes)

Öffnen Sie das Projekt zu dieser Lektion und Sie werden zwei Layer mit Namen 23030 und 4326 sehen. Sie enthalten beide dieselben Punkte aber in verschiedenen KBS (EPSG:23030 und EPSG:4326). Sie erscheinen an derselben Stelle in der Karte, weil QGIS in Echtzeit eine Reprojektion in das KBS des Projektes (EPSG:4326) vornimmt. Es handelt sich aber nicht um dieselben Layer.

Öffnen Sie den Algorithmus *Geometrieattribute hinzufügen*.



Dieser Algorithmus fügt neue Spalten in die Attributtabelle eines Vektorlayers ein. Der Inhalt der Spalten hängt vom Geometrietyp des Layers ab. Im Fall von Punkten, werden neue Spalten mit den X- und Y-Koordinaten jedes Punktes hinzugefügt.

In der Liste der verfügbaren Layer unter Eingabelayer werden beide Layer mit ihren jeweiligen KBS angezeigt. Das bedeutet, dass die Layer verschieden behandelt werden, obwohl sie an derselben Stelle angezeigt werden. Wählen Sie den 4326 Layer.

Der andere Parameter steuert, wie der Algorithmus die Koordinaten ermittelt, die in den Ergebnislayer eingefügt werden. Die meisten Algorithmen verwenden diese Option nicht. Sie nutzen direkt die vorhandenen Koordinaten. Wählen Sie *Layer-KBS*, um die Koordinaten so wie sie sind zu verwenden. Auf diese Art arbeiten fast alle Geoalgorithmen.

Wir erhalten einen neuen Layer mit exakt denselben Punkten wie in den anderen beiden Layern. Wenn Sie mit der rechten Maustaste auf den Layer klicken und seine Eigenschaften öffnen, sehen Sie, dass er dasselbe KBS wie der Eingabelayer verwendet, d.h. EPSG:4326. Wenn der Layer in QGIS geladen wird, fragt QGIS nicht nach dem KBS - es ist schon bekannt.

Beim Öffnen der Attributtabelle des neuen Layers sehen wir zwei neue Felder, die die X- und Y-Koordinaten für jeden Punkt enthalten.

	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

Die Koordinatenwerte gehören zum KBS des Layers, so wie wir es bei der Erstellung eingestellt hatten. Aber auch wenn wir eine andere Option gewählt hätten, wäre das Ausgabe KBS des Layers gleich, da das KBS des Eingabelayers als Vorgabe für das KBS des Ausgabelayers verwendet wird. Die Wahl einer anderen Option würde zwar die Koordinatenwerte in der Attributtabelle ändern, aber nicht das KBS des Ergebnislayers an sich.

Führen Sie nun dieselbe Berechnung mit dem anderen Layer aus. Sie werden sehen, das der im Ergebnis erzeugte Layer exakt an derselben Stelle wie die anderen Layer angezeigt wird. Er hat das KBS EPSG:23030, so wie sein Eingabelayer.

Wenn Sie sich seine Attributtabelle ansehen, bemerken Sie, dass die Werte von denen des ersten Layers abweichen.

	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

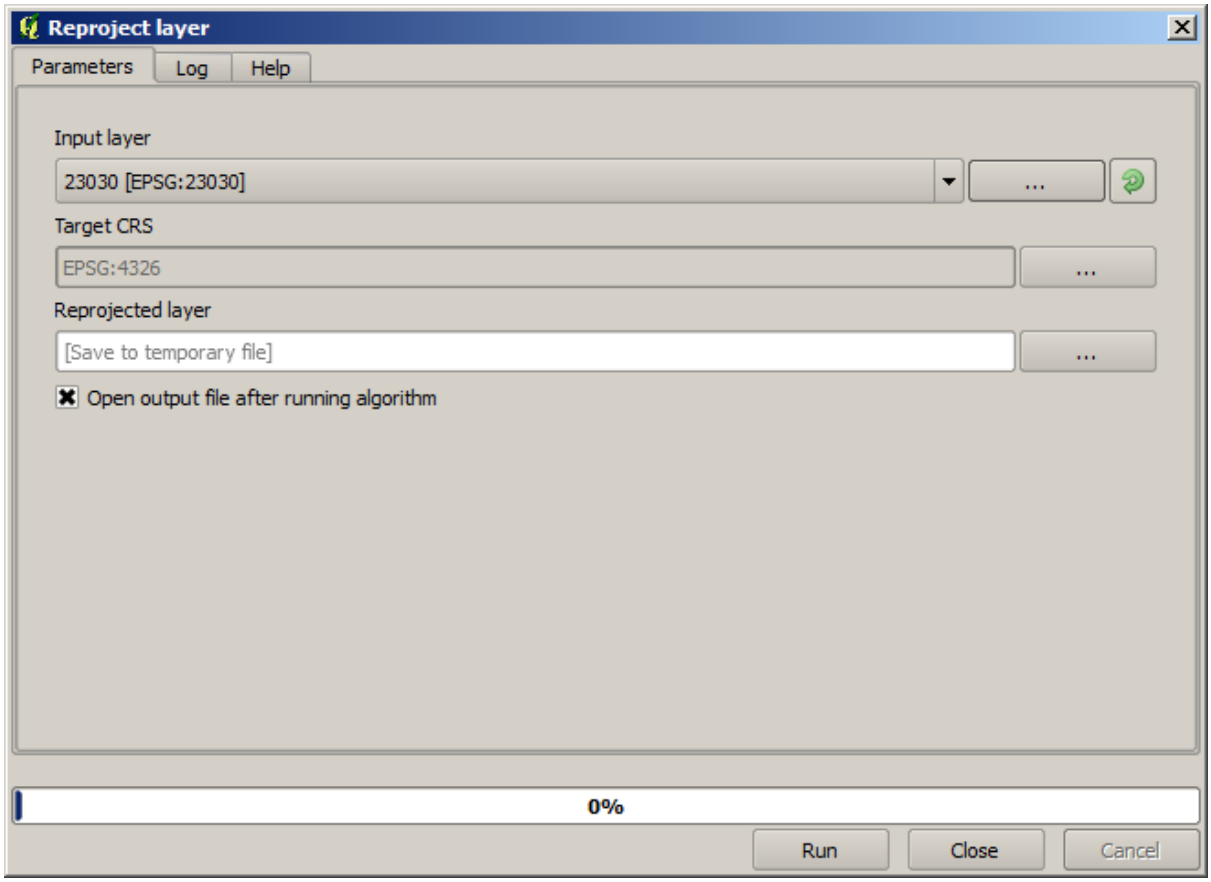
Das liegt daran, dass die Originaldaten voneinander abweichen (sie nutzen verschiedene KBS) und die Koordinaten von dort stammen.

Wass sollte man daraus lernen? Der Hauptgedanke hinter diesen Beispielen ist, dass die Geoalgorithmen Layer so verwenden wie sie sind. Sie ignorieren vollständig die Reprojektion die QGIS unter Umständen zur Darstellung benutzt. Mit anderen Worten: vertrauen Sie nicht dem, was Sie in der QGIS Karte sehen. Denken Sie immer daran, dass die Originaldaten verwendet werden. Das ist in diesem Fall nicht so wichtig, da wir jeweils nur einen Layer gleichzeitig verwenden. Bei Algorithmen, die mehrere Layer benutzen (z.B. der Zuschneiden Algorithmus) kann es so aussehen, als ob sie übereinander liegen. In Wirklichkeit sind sie aber weit voneinander entfernt, da sie verschiedene KBS verwenden.

Algorithmen führen keine Reprojektion durch (mit Ausnahme des reprojizieren Algorithmus wie wir gleich sehen werden). Sie müssen darauf achten, dass die Layer dasselbe KBS haben.

Ein interessantes Modul, das sich mit KBS befasst, ist die Reprojektion. Es stellt einen Sonderfall dar, da es einen Eingabelayer entgegen nimmt (den zu reprojizierenden) aber sein KBS nicht für den Ausgabelayer verwendet.

Öffnen Sie den Algorithmus *Layer reprojizieren*.



Wählen Sie einen der Layer als Eingabelayer und EPSG:23029 als Ziel KBS. Starten Sie den Algorithmus und Sie erhalten einen neuen Layer, identisch zum Eingabelayer, aber mit einem anderen KBS. Er wird im Arbeitsbereich an derselben Stelle wie die anderen Layer angezeigt, da QGIS ihn zur Laufzeit reprojiziert. Seine eigentlichen Koordinaten sind aber andere. Sie können das mit Hilfe des Algorithmus *Geometrieattribute hinzufügen* überprüfen. Die Koordinaten sind verschieden zu den Koordinaten in den beiden anderen Attributtabellen der anderen Layer.

17.7 Auswahl

Bemerkung: In dieser Lektion werden wir sehen, wie die Verarbeitungsalgorithmen Auswahlen in Vektorlayern behandeln, die zur Eingabe verwendet werden. Wir werden lernen, wie man eine Auswahl mit einem bestimmten Typ eines Algorithmus erstellt.

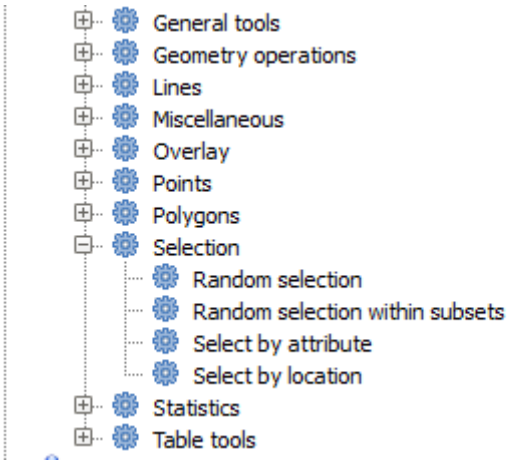
Anders als in anderen QGIS Erweiterungen, finden wir in den Verbeitungswerkzeugen keine Auswahlmöglichkeit wie „nur ausgewählte Objekte benutzen“. Das Verhalten bezüglich der Auswahl wird für die gesamte Erweiterung und alle dazu gehörenden Algorithmen vorgegeben, nicht für jede Ausführung eines Algorithmus. Es gilt für Algorithmen die folgende einfache Regel für Vektorlayer.

- Wenn Teile des Layer ausgewählt sind, werden nur die selektierten Objekte verwendet.
- Wenn keine Auswahl vorhanden ist, werden alle Objekte verwendet.

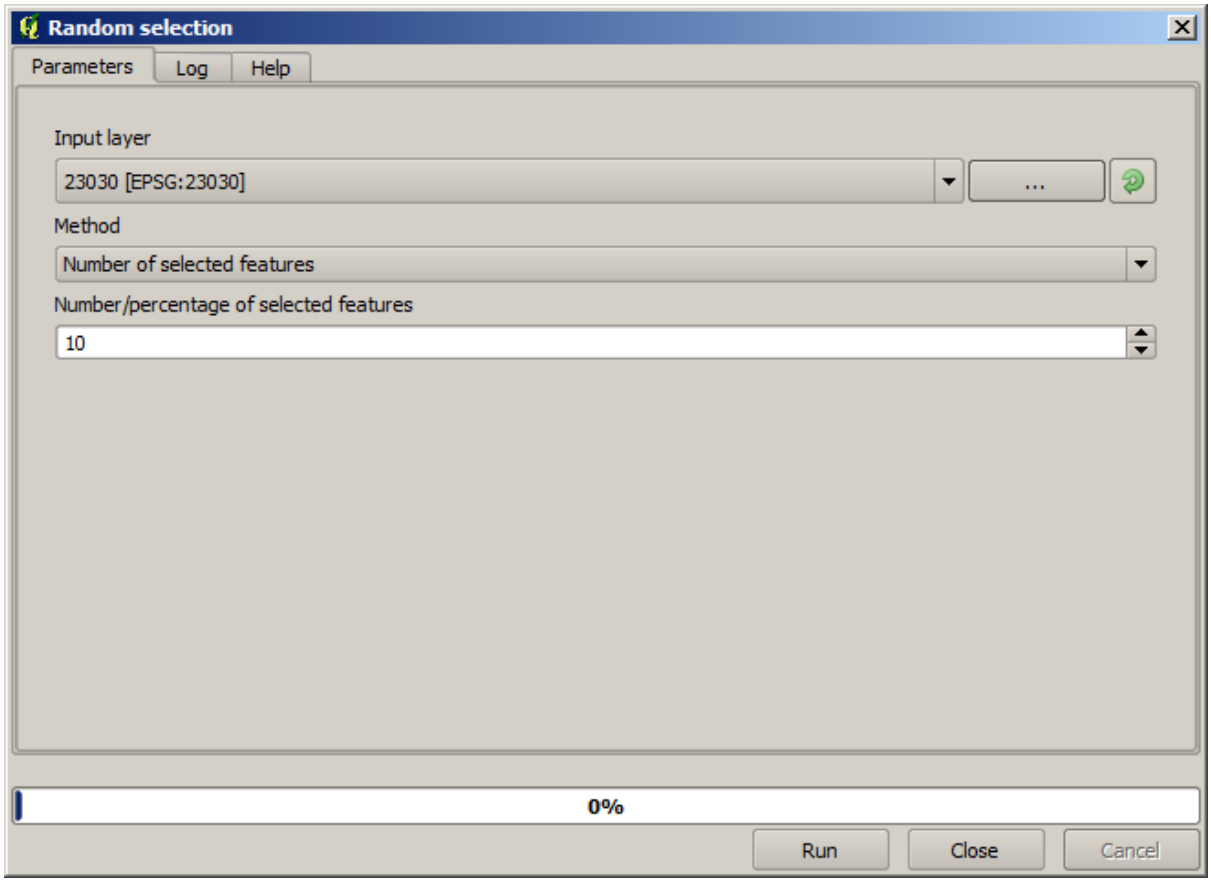
Bitte beachten Sie, dass sie dieses Verhalten unter *Optionen* ► *Verarbeitung* ► *Allgemein* anpassen können.

Sie können das selbst durch Auswahl einiger Punkte aus der letzten Übung und Start der Reprojektion testen. Der reprojizierte Layer enthält dann nur die Punkte die ausgewählt waren. Wenn Sie keine Auswahl getroffen hatten, enthält der reprojizierte Layer alle Punkte des Originallayers.

Um eine Auswahl zu erstellen, können Sie alle verfügbarer Methoden und Tools in QGIS benutzen. Sie können sogar einen Geoalgorithmus dazu verwenden. Algorithmen zur Erstellung einer Auswahl findet man unter *Vektorauswahl*



Öffnen Sie den Algorithmus *Zufällige Auswahl*.



Wenn man die Vorgabeeinstellungen so belässt, werden 10 Punkte des aktuellen Layers ausgewählt.

Wir sehen, dass der Algorithmus keine Ausgabe erstellt, er modifiziert den Eingabelayer (nicht den Layer an sich, aber die Auswahl der Objekte des Layers). Das ist ein unübliches Verhalten, da alle anderen Algorithmen neue Layer erzeugen und die Eingabelayer nicht verändern.

Die Auswahl ist nicht Teil der Daten an sich, sondern nur etwas das in QGIS existiert. Daher können diese Auswahl Algorithmen nur für in QGIS geöffnete Layer verwendet werden. Die Dateiauswahl, die als Parameter enthalten ist, kann daher nicht verwendet werden.

Die Auswahl die wir eben getroffen haben, genauso wie die meisten der anderen Selektionen, die durch die Auswahl Algorithmen erzeugt werden, können auch in QGIS manuell erstellt werden. Sie fragen sich daher vielleicht, warum man die Algorithmen verwenden soll. Auch wenn es momentan noch keinen Sinn ergibt, wir werden später sehen, wie man Modelle und Skripte erstellt. Wenn man eine Auswahl in der Mitte eines Modells (das einen Verarbeitungs Workflow definiert) erstellen möchte, kann nur ein Geoalgorithmus in das Modell eingefügt werden. Andere QGIS Elemente und Operationen können nicht eingefügt werden. Das ist der Grund, warum einige Verarbeitungsalgorithmen Funktionalitäten duplizieren.

Fürs Erste reicht es sich zu merken, dass Auswahlen mit Hilfe von Geoalgorithmen erstellt werden können. Algorithmen verwenden bei einer getroffenen Auswahl nur die selektierten Objekte und ansonsten alle Objekte.

17.8 Ausführung eines externen Algorithmus

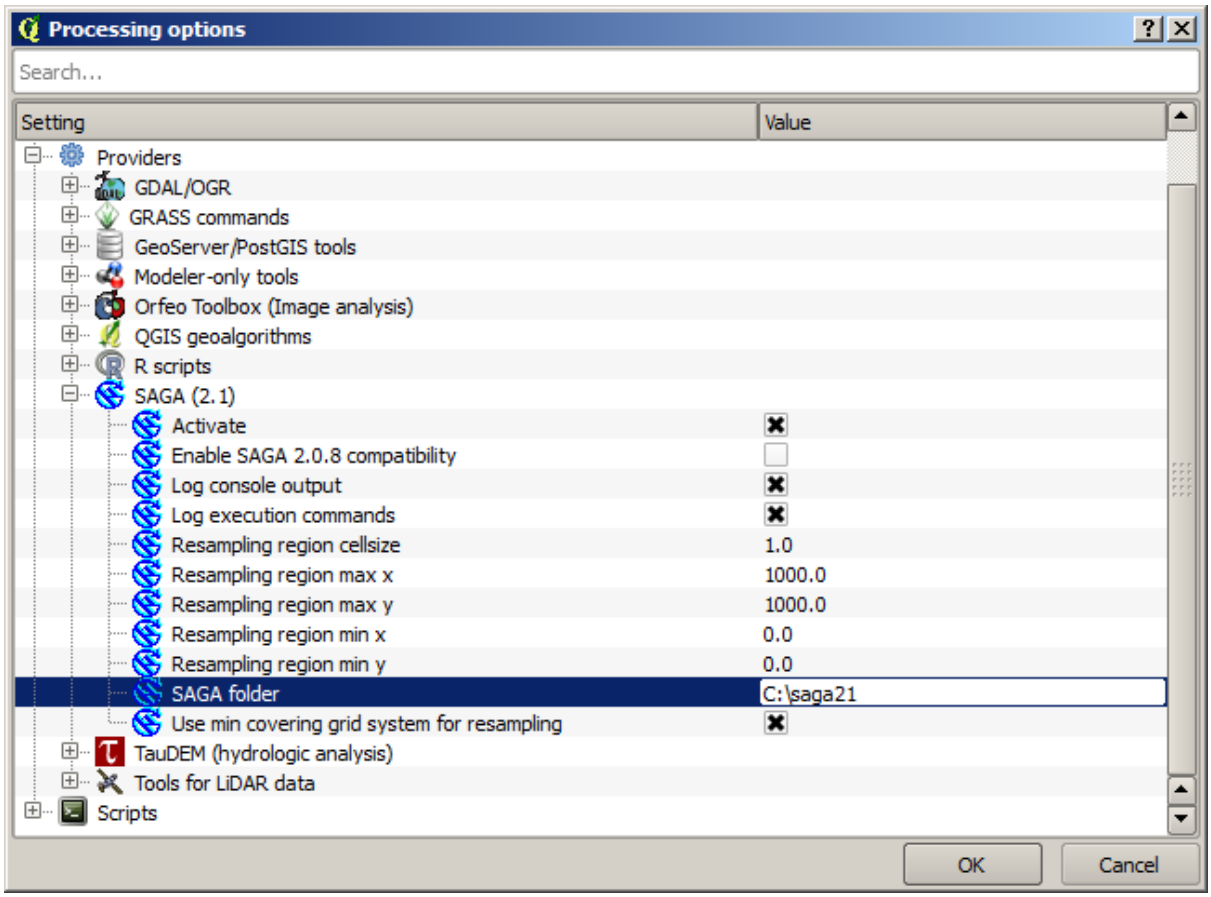
Bemerkung: In dieser Lektion werden wir lernen, wie man Algorithmen von Drittherstellern, insbesondere SAGA einem der Hauptanbieter von Algorithmen, verwendet.

Alle Algorithmen, die wir bisher verwendet haben, sind Teil der Verarbeitungsumgebung. Das bedeutet, dass es sich um *native* Algorithmen handelt, die innerhalb der Erweiterung enthalten sind und von QGIS genau wie die Erweiterung selbst gestartet werden. Eine der großartigsten Möglichketien von QGIS ist, dass es Algorithmen von externen Anwendungen verwenden kann und damit die Möglichkeiten dieser Anwendungen erweitern kann. Solche Algorithmen sind in der Verarbeitungsumgebung eingebunden, so dass man sie einfach aus QGIS heraus starten kann und QGIS Daten damit verwenden kann.

Einige der Algorithmen, die Sie in der vereinfachten Ansicht sehen, erfordern die Installation von Drittanwendungen auf Ihrem System. Ein besonders interessanter Anbieter von Algorithmen ist SAGA (System for Automated Geospatial Analysis). Als Erstes müssen wir alles so einrichten, dass QGIS SAGA korrekt aufruft. Das ist nicht schwer, aber es ist wichtig, zu verstehen wie es funktioniert. Jede externe Anwendung hat seine eigene Konfiguration. Wir gehen hier vor allem auf SAGA ein, dass wir hauptsächlich verwenden werden. In späteren Kapiteln behandeln wir weitere Anwendungen.

Wenn Sie unter Windows arbeiten, ist es am besten den QGIS-Installer zu verwenden. Er sorgt dafür, dass die erforderlichen Abhängigkeiten einschließlich SAGA installiert werden. Wenn Sie ihn verwendet haben, ist nichts

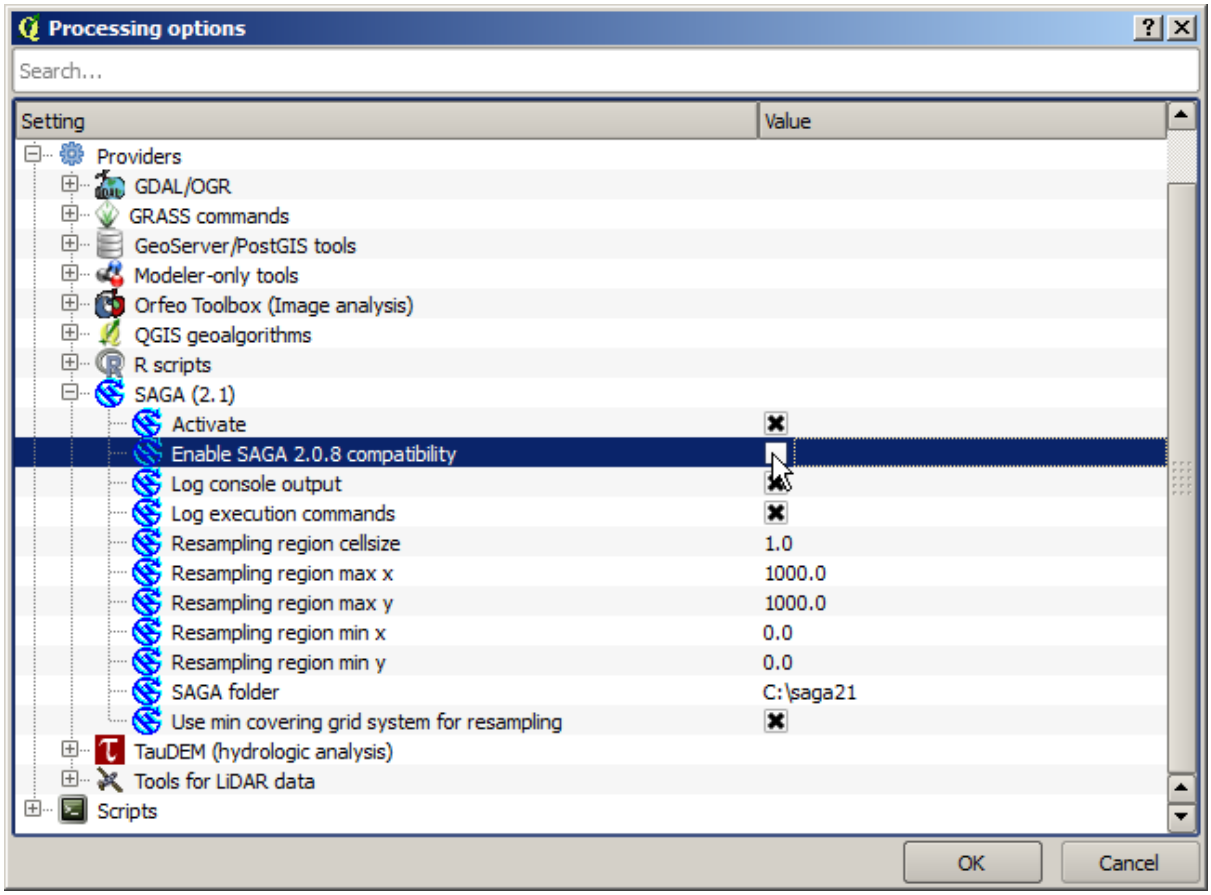
weiter zu tun. Sie können den Einstellungsdialog öffnen und zur Gruppe *Datenanbieter/SAGA* wechseln.



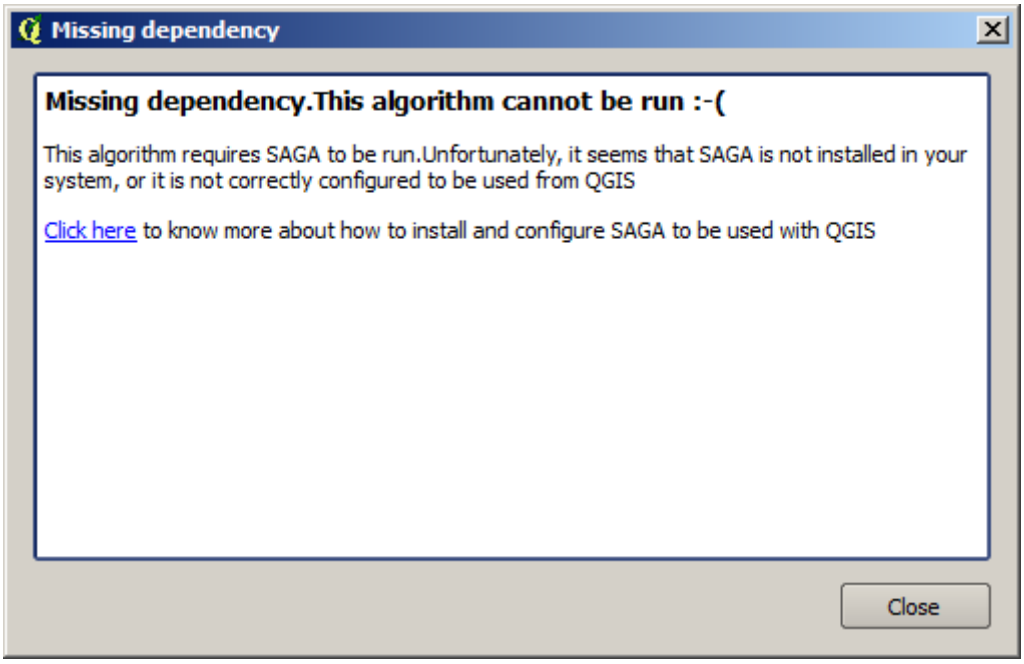
Der SAGA Pfad sollte schon eingerichtet sein und auf den Ordner zeigen, in dem SAGA installiert ist.

Wenn Sie QGIS nicht mit Hilfe des standalone-installers installiert haben, müssen Sie den Pfad zu Ihrer SAGA-Installation (muss schon vorhanden sein) hier eingeben. Die erforderliche Version von SAGA ist 2.1 [dies ändert sich je nach Veröffentlichungzeitpunkt einer SAGA Version].

Wenn Sie Linux verwenden, brauchen Sie den Pfad zu Ihrer SAGA Installation nicht in den Verarbeitungsoptionen ändern. Statt dessen müssen Sie SAGA installieren und dabei sicher stellen, dass der SAGA Ordner in der PATH Variable enthalten ist, so dass das Programm von der Kommandozeile aus gestartet werden kann (öffnen Sie eine Konsole und geben Sie `saga_cmd` zum Test ein). Unter Linux ist die Sollversion für SAGA ebenfalls 2.1 aber in einigen Installationen (wie z.B. der OSGeo Live DVD) ist unter Umständen nur die Verison 2.0.8 verfügbar. Es kann sein, dass in diesen Fällen auch Packete in der Version 2.1 verfügbar sind. Sie werden häufig nicht verwendet, da sie noch nicht so stabil laufen. Wenn Sie lieber die verbreitetere und stabilere Version 2.0.8 verwenden möchten, können Sie das in den Verarbeitungsoptionen in der *SAGA* Gruppe einstellen.

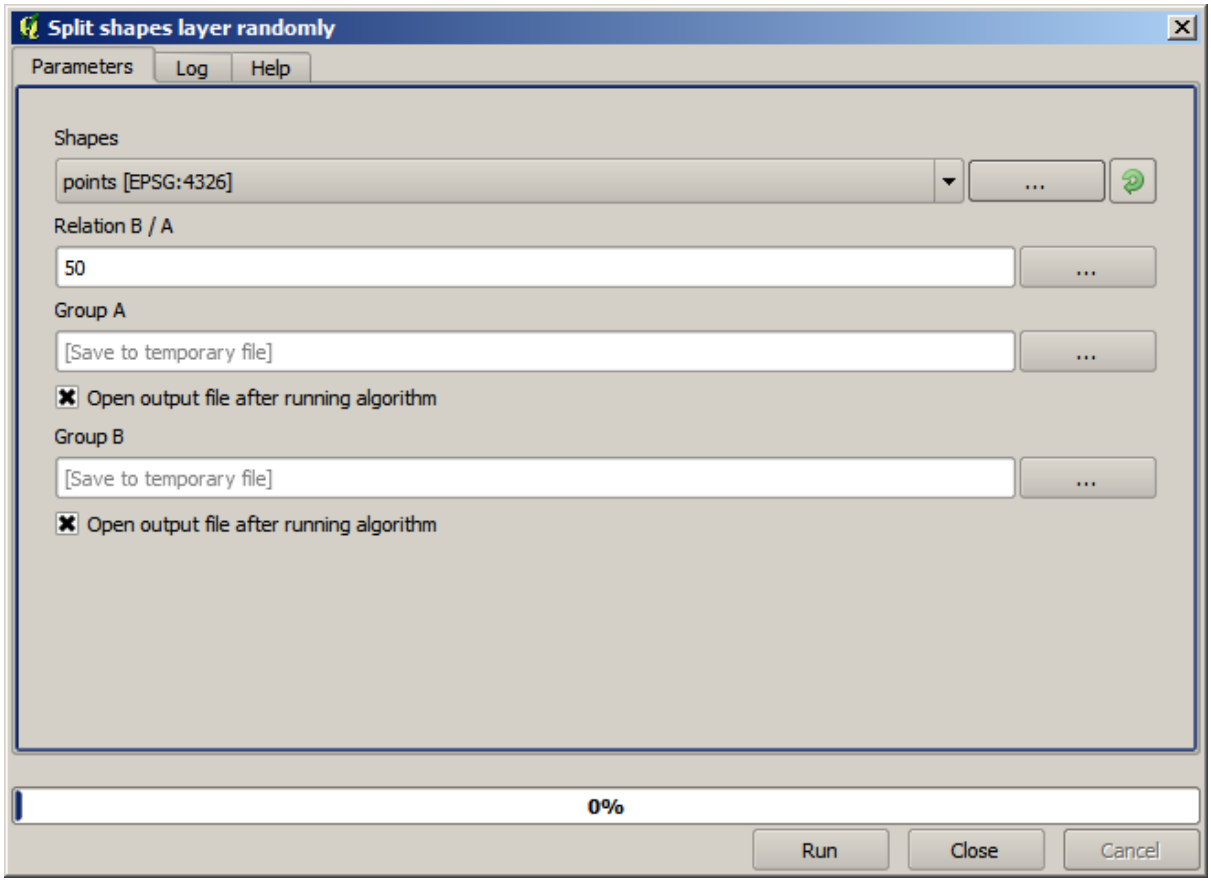


Nachdem SAGA installiert ist, können Sie einen SAGA Algorithmus wie jeden anderen Algorithmus durch Doppelklick starten. Da wir eine vereinfachte Schnittstelle benutzen, wissen wir nicht, welche Algorithmen auf SAGA oder eine andere externe Anwendung aufsetzen. Wenn eine zugehörige Anwendung nicht installiert ist, erhalten wir nach Doppelklick auf den Algorithmus in etwa die folgende Meldung.

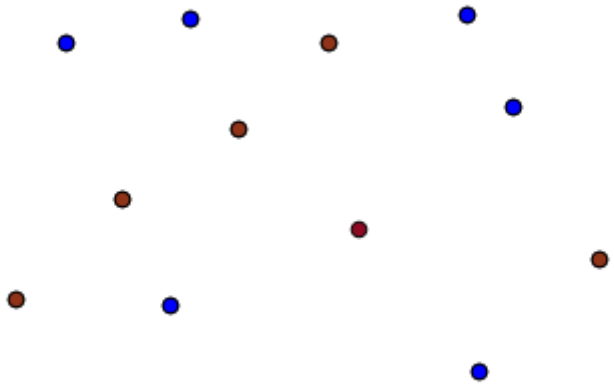


Wir gehen davon aus, dass SAGA korrekt installiert ist. Es öffnet sich dann an Stelle des Fensters mit der Fehlermeldung der Parameterdialog.

Wir testen einen SAGA–basierten Algorithmus: *Split shapes layer randomly*.



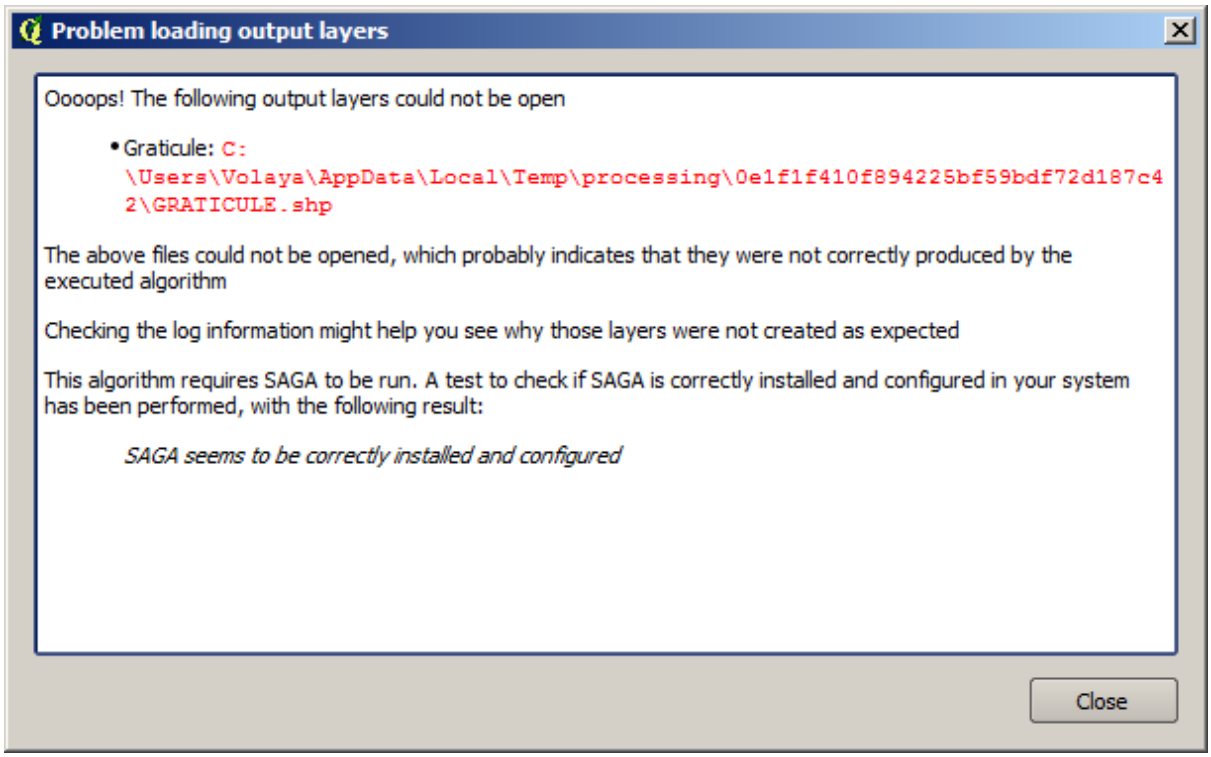
Benutzen Sie den Punktlayer im zur Lektion gehörenden Projekt. Wenn Sie die Vorgabeparameter verwenden, erhalten Sie in etwa die folgende Ausgabe (die Teilung erfolgt zufällig, d.h. es wird ggf. eine abweichende Ausgabe erzeugt).



Der Eingabelayer wurde in zwei Layer mit jeweils derselben Anzahl an Punkten geteilt. Das Ergebnis wurde von SAGA berechnet und nachfolgend von QGIS übernommen und ins QGIS Projekt eingefügt.

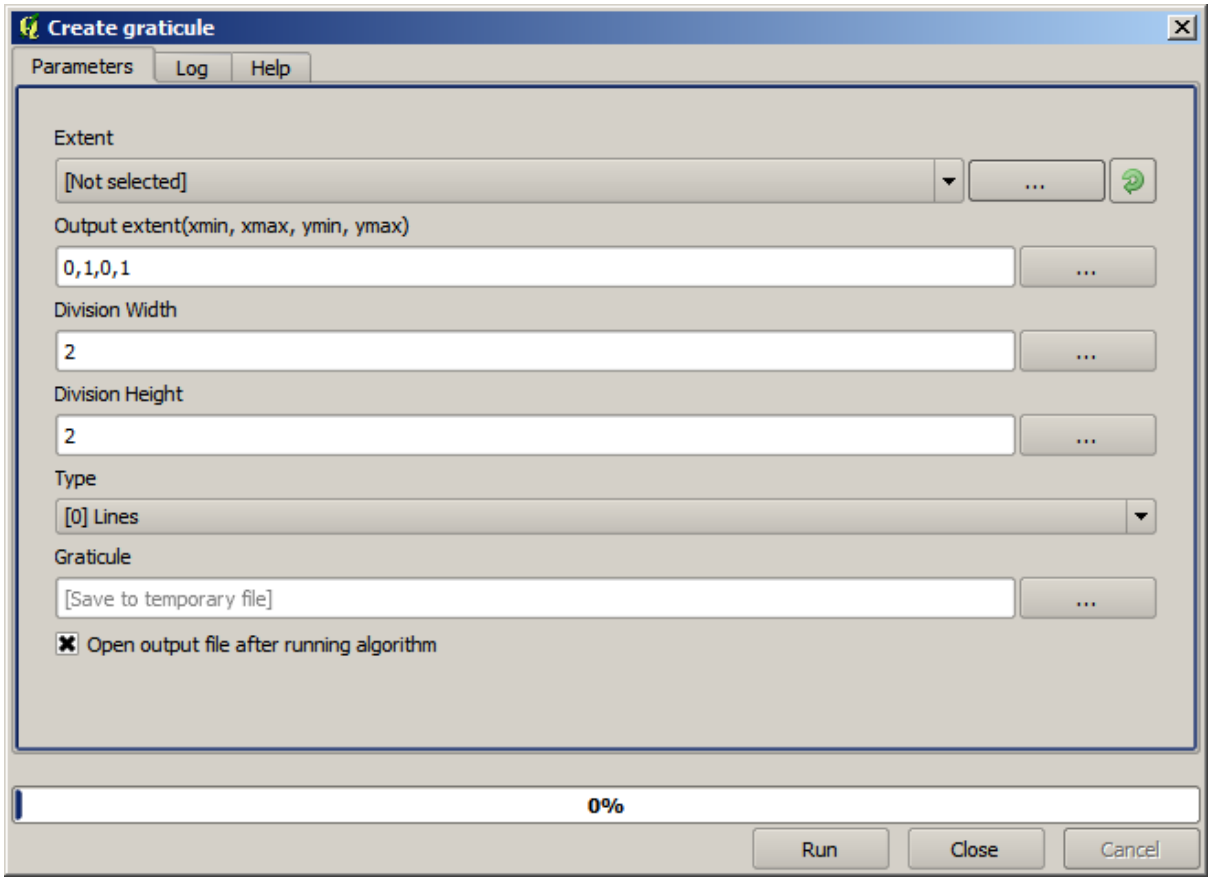
Wenn alles klappt, werden Sie keinen Unterschied zwischen diesem SAGA-basierten Algorithmus und den anderen vorher verwendeten merken. Es kann allerdings passieren, dass SAGA nicht in der Lage ist, die von QGIS erwartete

Datei zu erstellen. In diesem Fall kommt es zu Problemen bei der Übernahme des Ergebnisses in das QGIS Projekt und es erscheint eine Fehlermeldung ähnlich der folgenden.



Diese Art von Fehler kann selbst dann auftreten, wenn SAGA (oder jede andere Anwendung, die man aus den Verarbeitungswerkzeugen aufruft) korrekt installiert ist. Es ist wichtig zu lernen, wie man dieser Fehlerart umgeht. Wir werden diese Art von Fehler nun erzeugen.

Öffnen Sie den Algorithmus *Create graticule* und verwenden dabei die folgenden Werte.



Wir nutzen Werte für Breite und Höhe, die größer als die vorgegebene Ausdehnung des Gitters sind, so dass SAGA keine Ausgabe erzeugen kann. Mit anderen Worten: die Werte der Parameter sind falsch vergeben, werden aber vor der Übergabe an SAGA nicht geprüft. Da SAGA das Gitter nicht erzeugen kann, wird es den erwarteten Layer nicht erzeugen und man erhält die obige Fehlermeldung.

Bemerkung: Ab SAGA >= 2.2.3 werden falsche Eingabewerte automatisch korrigiert, so dass man in diesem Fall keine Fehlermeldung erhält. Um einen Fehler zu provozieren, kann man statt dessen negative Werte für die Parameter Divison verwenden.

Das Verständnis dieser Art von Problemen hilft dabei, zu verstehen was vorsichgeht und wie man eine Lösung findet. Man erkennt aus der Fehlermeldung, dass ein Test ausgeführt wird, ob die Verbindung mit SAGA richtig funktioniert bzw. ob der Algorithmus in der erwarteten Weise ausgeführt wird. Das gilt nicht nur für SAGA, sondern auch für alle anderen externen Anwendungen.

In der nächsten Lektion lernen wir das Verarbeitungsprotokoll kennen. Hier werden Informationen über die Ausführung von Geoalgorithmen aufgezeichnet. Wir werden sehen, wie man weitere Details zu Problemen dieser Art erhält.

17.9 Das Prozessierungsprotokoll

Bemerkung: Diese Lektion beschreibt das Verarbeitungsprotokoll.

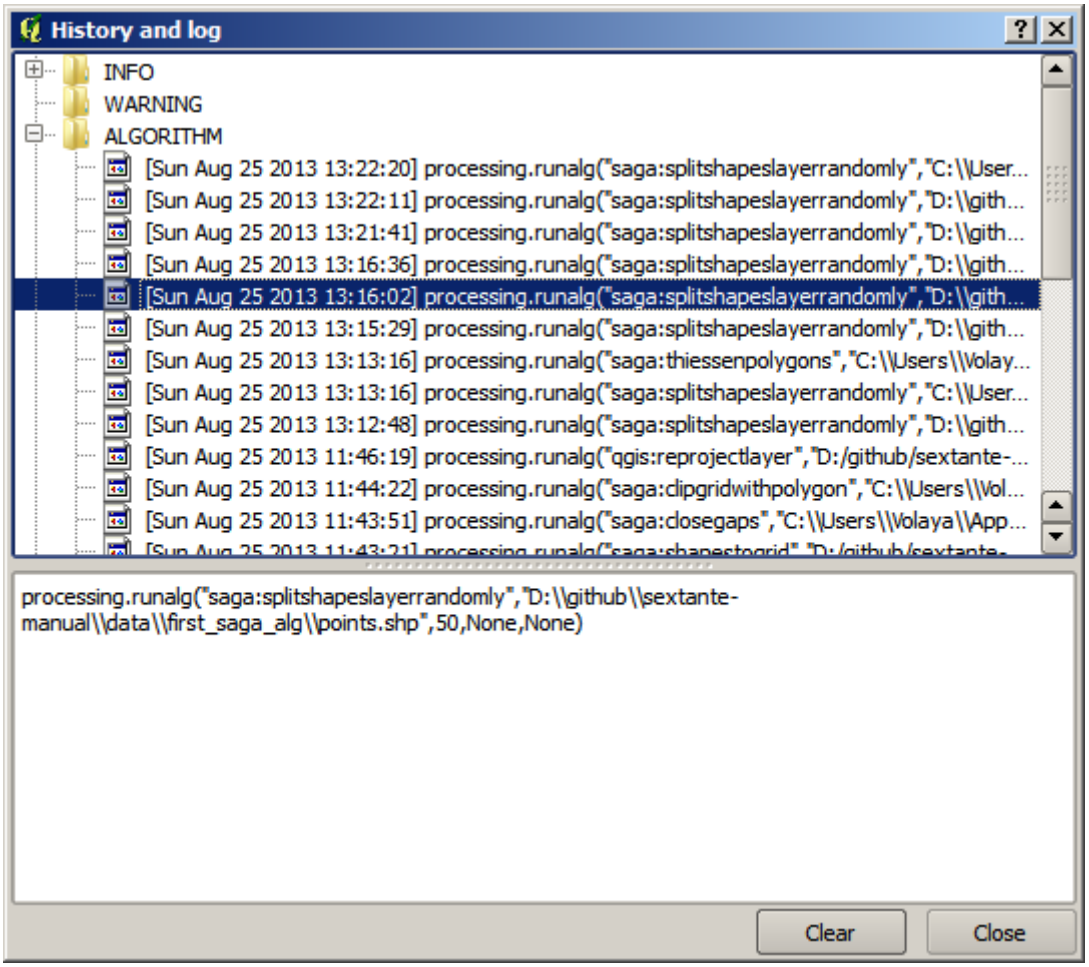
Alle Analysen, die in der Verarbeitungsumgebung ausgeführt werden, werden von QGIS protokolliert. Damit erhält man zusätzliche Informationen, was bei der Ausführung der Verarbeitungswerkzeuge getan wurde. Es erleichert die Lösung von auftretenden Problemen und erlaubt die Wiederholung vorhergehender Operationen, da die Protokollierung auch interaktive Elemente einschließt.

Öffnen Sie das Protokoll durch Klick auf den Ballon auf der rechten Seite der Statusleiste in QGIS. Einige Algorithmen hinterlassen hier Nachrichten zu ihrer Ausführung. Algorithmen, die eine externe Anwendung aufrufen, hinterlassen in der Regel ein Protokoll der Konsolenausgabe der Anwendung und damit Informationen über deren Ausführung. Beim Betrachten des Protokolls sehen wir die Ausgabe des SAGA Algorithmus, den wir vorher gestartet hatten (und der nicht funktionierte, da die Eingabedaten nicht korrekt waren).

Das hilft uns zu verstehen, was passiert ist. Erfahrene Benutzer sind in der Lage, die Ausgabe zu analysieren und herauszufinden, warum der Algorithmus mit einem Fehler beendet wurde. Wenn Sie kein erfahrener Nutzer sind, kann die Ausgabe durch andere zur Fehlerdiagnose verwendet werden. Es könnte sich um ein Problem bezüglich der Installation von externer Software oder um ein Problem mit den verwendeten Daten handeln.

Selbst wenn ein Algorithmus ausgeführt werden konnte, können Warnungen im Protokoll erstellt werden, wenn das Resultat falsch sein könnte. Wenn z.B. ein Algorithmus zur Interpolation mit nur einer sehr geringen Anzahl von Punkten ausgeführt wird, erzeugt der Algorithmus vermutlich ein Ergebnis - es ist aber wahrscheinlich nicht richtig, da mehr Punkte verwendet werden sollten. Wenn einige Aspekte von Algorithmen nicht ganz klar sind, ist es sinnvoll, sich die Warnungen regelmäßig anzusehen.

Im Menü *Verarbeitung* unter *Protokoll* finden wir den Punkt *Algorithm*. Jeder Aufruf eines Algorithmus, egal ob von der Konsole (wird später in dieser Anleitung beschrieben) oder der grafischen Oberfläche, wird hier als Konsolenaufruf aufgezeichnet. Man erhält damit eine komplette Historie aller Aufrufe von Algorithmen der aktuellen Arbeitssitzung. Die Historie sieht in etwa wie folgt aus:



Das ist sehr nützlich, sobald man anfängt mit der Konsole zu arbeiten. Man erlernt die Syntax der Algorithmen in der Konsole. Wir werden uns damit beschäftigen, wenn wir Analysen von der Konsole aus betrachten.

Die Historie ist interaktiv, d.h. man kann jeden vorher ausgeführten Algorithmus durch Doppelklick auf den Eintrag in der Historie erneut starten. Damit kann man ganz einfach bereits ausgeführte Arbeitsschritte wiederholen.

Probieren Sie z.B. Folgendes aus. Öffnen Sie die Daten zum ersten Kapitel der Anleitung und starten den dort beschriebenen Algorithmus. Gehen Sie jetzt zur Historie und dort zum letzten Eintrag in der Liste, der zu dem eben ausgeführten Algorithmus gehört. Klicken Sie doppelt darauf. Es sollte ein neues Ergebnis erstellt werden, genauso als wenn Sie den normalen Dialog aus der Werkzeugkiste verwendet hätten.

17.9.1 Fortgeschritten

Sie können den Algorithmus auch modifizieren. Kopieren Sie ihn einfach, öffnen Sie die *Erweiterungen* ► *Python-Konsole*, klicken Sie auf *Import Klasse* ► *Import Verarbeitungsklasse*, dann fügen Sie ihn ein, um die Analyse erneut auszuführen; ändern Sie den Text nach Belieben. Um die resultierende Datei anzuzeigen, geben Sie `iface.addVectorLayer('/Pfad/Dateiname.shp', 'Layername in der Legende', 'ogr')` ein. Andernfalls können Sie :kbd: *processing.runandload* verwenden.

17.10 Der Rasterrechner. Nullwerte

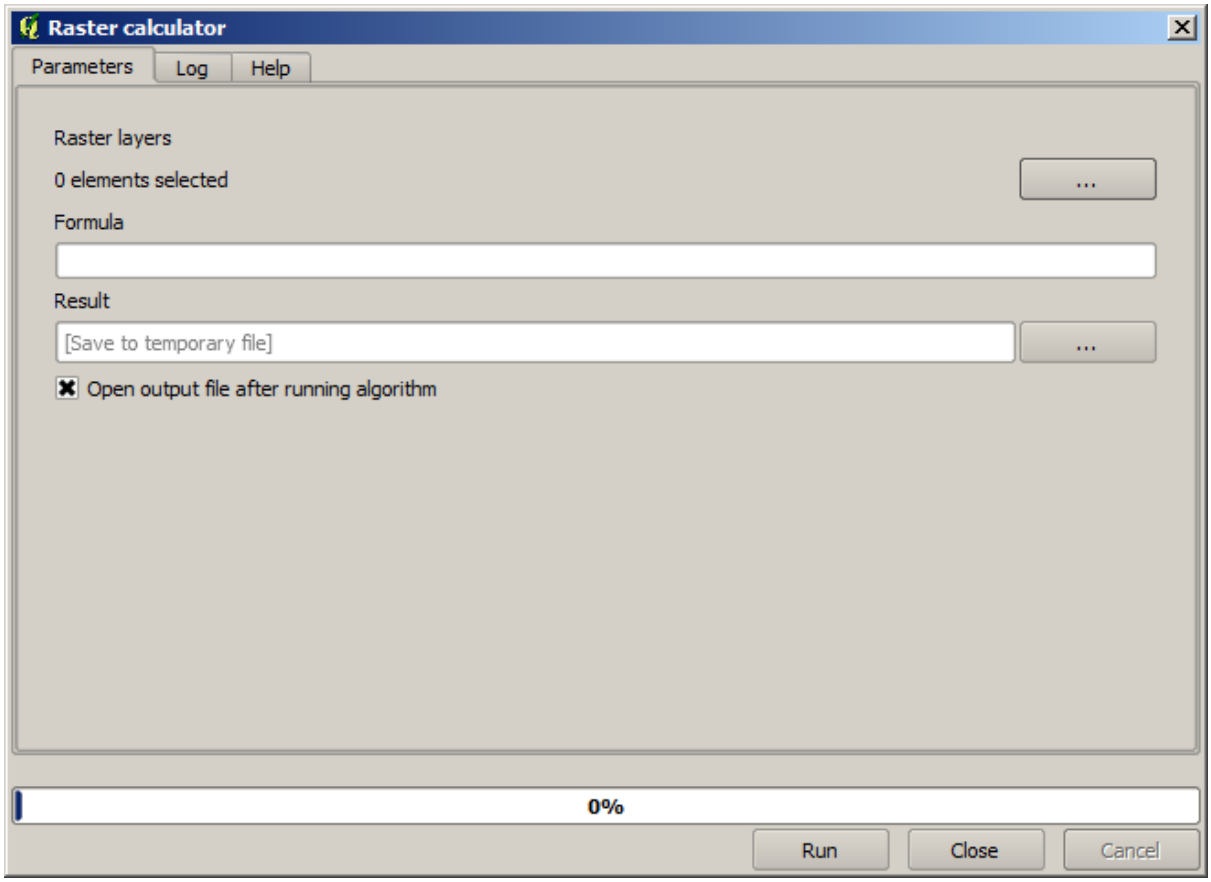
Bemerkung: In dieser Lektion werden wir lernen, wie man den Rasterrechner verwendet, um Operationen mit Rasterlayern auszuführen. Wir werden außerdem erläutern, worum es sich bei Leerwerten handelt und wie der Rechner und andere Algorithmen damit umgehen.

Der Rasterrechner ist einer der mächtigsten Algorithmen, die zur Verfügung stehen. Er ist sehr flexibel und vielseitig und kann für viele verschiedene Berechnungen verwendet werden. Er wird sehr schnell ein wichtiger Teil Ihrer Werkzeugkiste werden.

In dieser Lektion werden wir einige Berechnungen mit dem Rasterrechner ausführen. Die meisten davon sind eher einfach. Wir erfahren wie er verwendet wird und wie er sich in einigen speziellen Situationen verhält. Das Verständnis hierzu ist wichtig, um später die erwarteten Ergebnisse bei der Verwendung des Rechners zu erhalten und bestimmte Techniken, die in der Regel mit dem Rechner verwendet werden, zu verstehen.

Öffnen Sie das zu dieser Lektion gehörende QGIS Projekt. Sie werden sehen, dass es mehrere Rasterlayer enthält.

Öffnen Sie nun die Verarbeitungswerkzeuge und dort den Raster calculator.



Bemerkung: Die Schnittstelle unterscheidet sich in neueren Versionen.

Der Dialog enthält 2 Parameter.

- Die für die Analyse genutzten Layer. Hierbei handelt es sich um eine Mehrfacheingabe, d.h. man kann so viele Layer auswählen, wie man möchte. Klicken Sie auf den Knopf auf der rechten Seite und wählen die Layer aus, die Sie verwenden möchten.

- Die anzuwendende Formel. Die Formel verwendet die ausgewählten Layer, die mit Buchstaben als Variablen benannt werden (`a, b, c...`) oder `g1, g2, g3...` Das bedeutet, die Formel `a + 2 * b` ist dasselbe wie `g1 + 2 * g2`, sie berechnet die Summe der Werte des ersten Layers und dem zweifachen der Werte des zweiten Layers. Die Ordnung der Layer ist dieselbe wie im Auswahldialog.

Warnung: Der Rechner unterscheidet zwischen Groß- und Kleinschreibung.

Am Anfang ändern wir die Einheit des DEM von Meter auf Fuß. Wir benötigen die folgende Formel:

```
h' = h * 3.28084
```

Wählen Sie das DEM im Layer Feld und geben als Formel `a * 3.28084` ein.

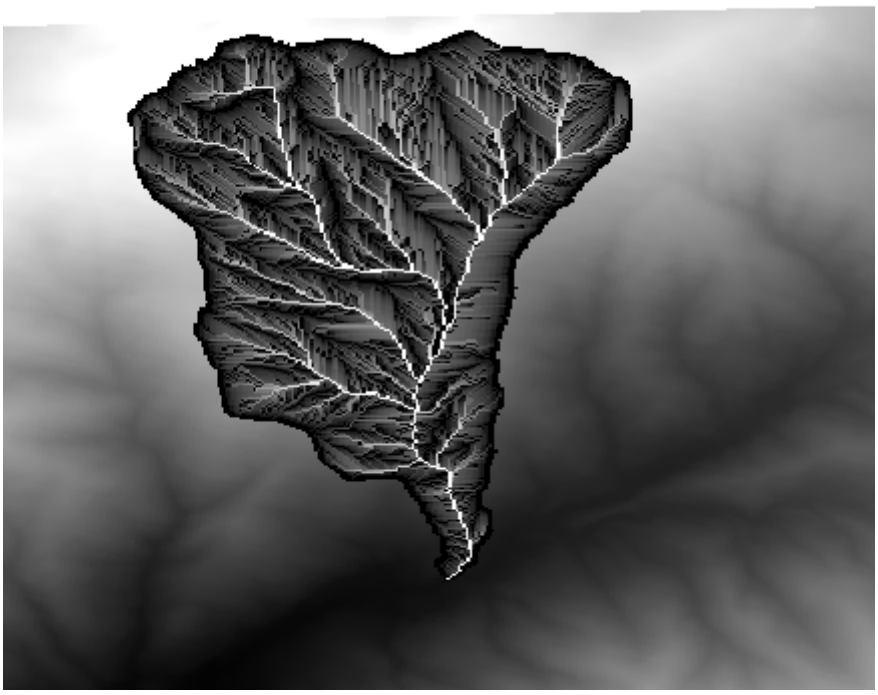
Warnung: Für Benutzer, die nicht englisch verwenden: nutzen Sie immer „.“ und nicht „,“.

Klicken Sie auf *Starte* um den Algorithmus zu starten. Wir erhalten einen mit der Quelle vergleichbaren Layer aber mit abweichenden Werten. Der Eingabelayer hat für alle Zellen gültige Werte, so dass der letzte Parameter keine Auswirkung hat.

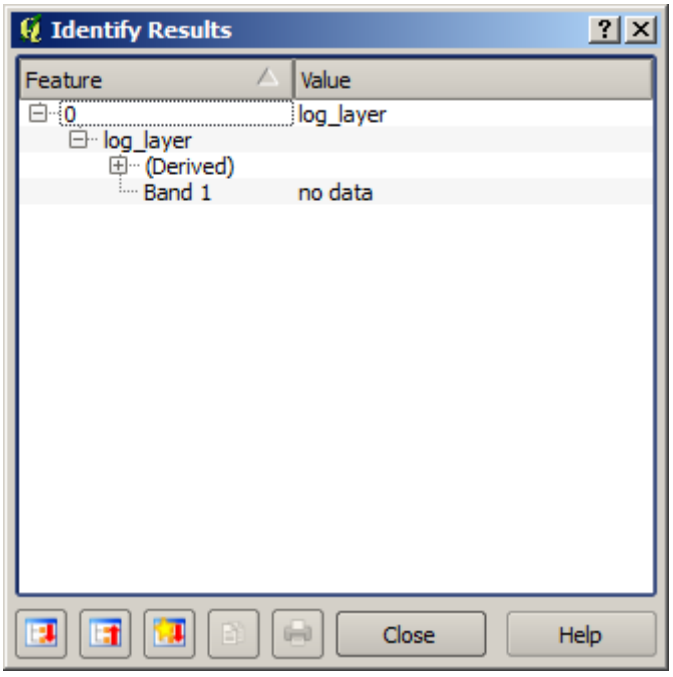
Wir werden nun eine weitere Berechnung durchführen, dieses Mal mit dem Layer *accflow*. Dieser Layer enthält Werte des akkumulierten Abflusses - ein hydrologischer Parameter. Er enthält diese Werte nur innerhalb der Fläche eines gegebenen Einzugsgebietes und außerhalb Leerwerte. Wie man sieht, ist die Ausgabe aufgrund der Verteilung der Werte nicht sehr informativ. Die Verwendung des logarithmierten akkumulierten Abflusses liefert eine deutlich informativere Repräsentation. Wir verwenden zur Berechnung den Rasterrechner.

Öffnen Sie den Dialog des Algorithmus noch einmal, wählen den Layer *accflow* als Eingabelayer und geben die folgende Formel vor: `log(a)`.

Wir erhalten den folgenden Layer.



Wir können die Werte des Layers mit Hilfe des *Abfrage* Werkzeuges an jedem Punkt ermitteln. Wenn wir außerhalb des Layers klicken, sehen wir, das der Layer nur Leerwerte enthält.



Für die nächste Übung werden wir zwei Layer statt einem verwenden. Wir werden ein DEM erhalten, das nur gültige Höhenwerte innerhalb des mit dem zweiten Layer vorgegebenen Einzugsgebietes enthält. Öffnen Sie den Dialog des Rasterrechners und wählen beide Layer als Eingabe aus. Geben Sie die folgende Formel vor:

```
a/a * b
```

`a` bezieht sich auf den Layer des akkumulierten Abflusses (er erscheint als erster in der Liste) und `b` bezieht sich auf das DEM. Im ersten Teil der Formel teilen wir den Layer des akkumulierten Abflusses mit sich selbst. Dabei erhalten wir einen Wert von 1 innerhalb und einen Leerwert außerhalb des Einzugsgebietes. Dieser Wert wird anschließend mit dem DEM multipliziert, um den Höhenwert innerhalb des Einzugsgebietes (`DEM * 1 = DEM`) und einen Leerwert außerhalb (`DEM * no_data = no_data`) zu erhalten.

Das ist der sich ergebene Layer.

Diese Technik wird häufig zur *Maskierung* von Werten eines Rasterlayers verwendet. Sie ist immer dann nützlich, wenn man Berechnungen auf Regionen durchführen möchte, die von der willkürlichen rechteckigen Form abweichen.

Wenn man z.B. das Histogramm der Höhenwerte eines Rasterlayers bestimmt, macht das erst mal nicht so viel Sinnn. Wenn man dagegen nur die Werte eines Einzugsgebietes berücksichtigt (wie im obigen Beispiel), erhält man ein sinnvolles Resultat mit Informationen über die Struktur des Einzugsgebietes.

Neben der Behandlung der Leerwerte gibt es weitere interessante Dinge an dem Algorithmus. Wenn wir uns die Ausdehnung der Layer ansehen, die wir multipliziert haben (man kann sich die Eigenschaften der Layer duch Doppelklick auf einen Layer im Inhaltsverzeichnis ansehen), sehen wir, dass sie nicht identisch sind. Das liegt daran, dass die Ausdehnung des Layers des akkumulierten Abflusses kleiner als die Ausdehnung des DEM ist.

Das bedeutet, dass diese Layer nicht zueinander passen. Sie können nicht direkt miteinander multipliziert werden. Sie müssen vorher umgerechnet werden, so dass ihre Größe und Ausdehnung homogen wird. Wir haben aber nichts dergleichen im Vorfeld getan. QGIS macht das bei Bedarf automatisch. Die Ausdehung der Ausgabe entspricht der minimalen Ausdehnung der Eingabelayer. Die Zellengröße entspricht der minimalen Größe der Eingabelayer.

In diesem Fall (wie in den meisten Fällen) erhalten wir das gewünschte Ergebnis. Man sollte die im Hintergund ablaufenden Operationen immer bedenken, da sie unter Umständen das Ergebnis beeinflussen können. Wenn dieses Verhalten ausgeschlossen werden soll, kann im Vorfeld eine manuelle Anpassung der Eingabelayer erfolgen. In späteren Kapiteln werden wir mehr über das Verhalten von Algorithmen bei Verwendung mehrerer Rasterlayer lernen.

Wir beenden diese Lektion mit einer weiteren Übung zur Maskierung. Wir werden die Hangneigung für alle Flächen in einer Höhe zwischen 1000 und 1500 Metern berechnen.

In diesem Fall haben wir keinen Layer, den wir als Maske verwenden können. Wir erstellen diesen Layer mit Hilfe des Rasterrechners.

Starten Sie den Rasterrechner mit dem DEM als Eingabelayer und der folgenden Formel

```
ifelse(abs(a-1250) < 250, 1, 0/0)
```

Wie man sieht, kann der Rasterrechner nicht nur für einfache Rechenoperationen sondern auch für bedingte Berechnungen wie im Beispiel verwendet werden.

Als Ergebnis erhalten wir eine 1 für Zellen, die im Wertebereich liegen und einen Leerwert für Zellen außerhalb des Wertebereiches.

Der Leerwert entsteht aus dem Ausdruck 0/0. Da es sich um einen unbestimmten Wert handelt, fügt SAGA einen NaN Wert ein (Not a Number), der wie ein Leerwert behandelt wird. Mit diesem Trick können wir einen Leerwert für Zellen vergeben, ohne zu wissen, welchem Wert der Leerwert für eine Zelle entspricht.

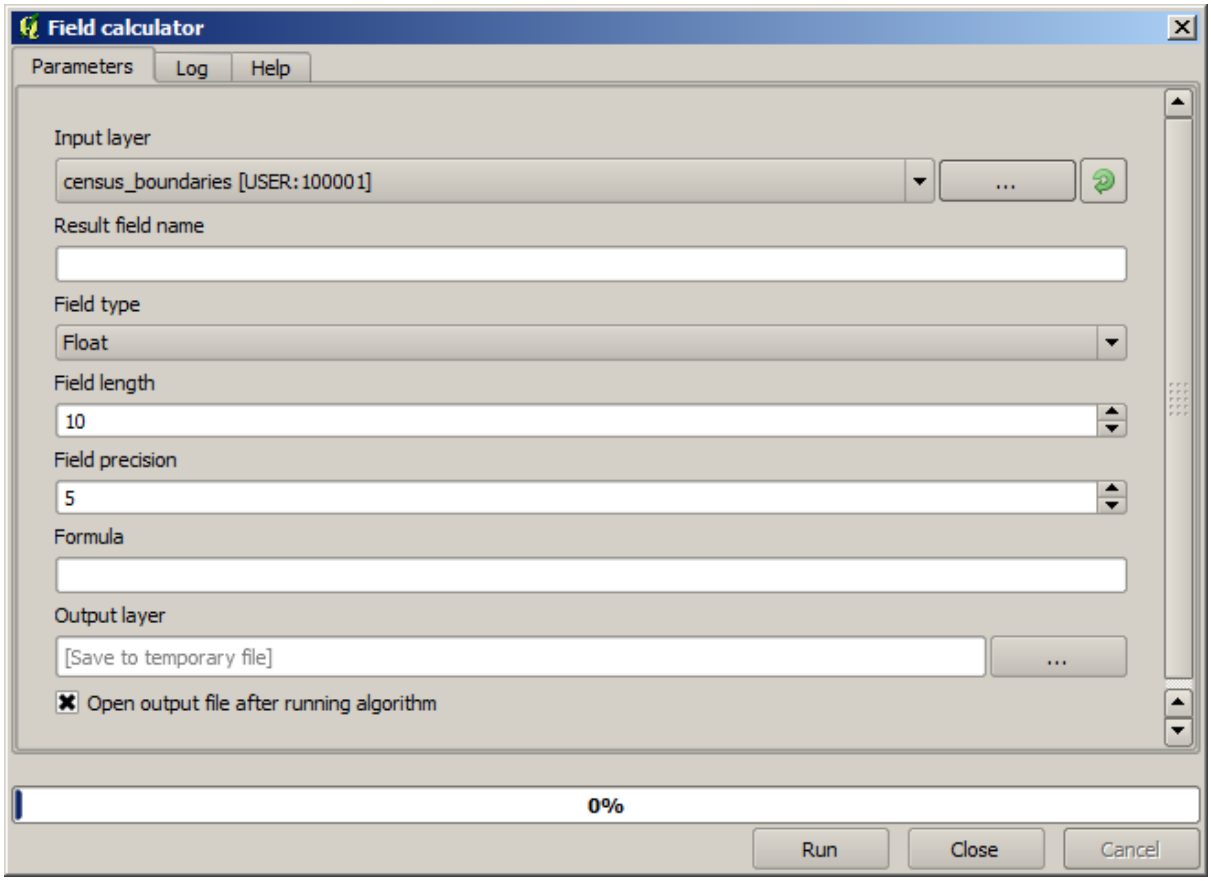
Der Layer muss nun nur noch mit dem Neigungslayer multipiliziert werden und wir erhalten unser Ergebnis.

Das kann alles in einer einzigen Operation im Rasterrechner erfolgen. Wir belassen die Ausführung als Übung für den Leser.

17.11 Vektorberechner

Bemerkung: In dieser Lektion lernen wir, wie man neue Attribute zu einem Vektorlayer basierend auf einem mathematischen Ausdruck mit Hilfe des Vektorrechners hinzufügt.

Wir wissen bereits, wie man den Rasterrechner verwendet, um neue Rasterlayer mit Hilfe von mathematischen Ausdrücken zu erstellen. Einen ähnlichen Algorithmus gibt es auch für Vektorlayer. Er erstellt eine neuen Layer mit denselben Attributen des Eingabelayers und einem zusätzlichen Attribut, das sich aus der Auswertung des Ausdrucks ergibt. Der Algorithmus wird *Feldrechner* genannnt und verwendet den folgenden Dialog zur Parametereingabe.



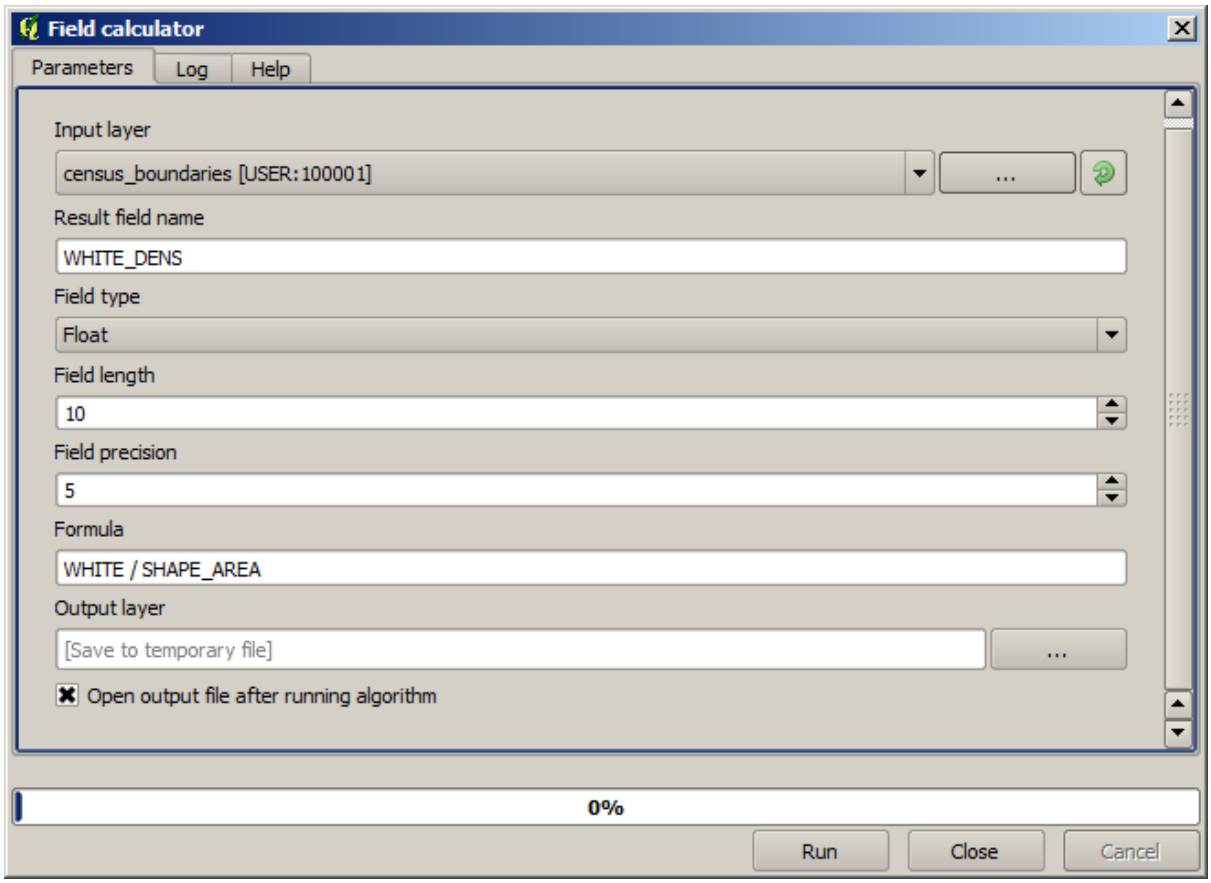
Bemerkung: In neueren Versionen hat sich die Eingabeschnittstelle stark verändert. Sie ist umfangreicher und einfacher zu nutzen.

Es folgen einige Beispiele zur Nutzung des Algorithmus.

Wir werden zuerst den Anteil der weißen Menschen in jedem Polygon, d.h. in einem Zensusgebiet, bestimmen. Wir können dazu die Felder `WHITE` und `SHAPE_AREA` aus der Attributtabelle verwenden. Wir dividieren die Werte und multiplizieren sie anschließend mit einer Million (um auf die Dichte je Quadratkilometer zu kommen). Wir können die folgende Formel verwenden.

```
( "WHITE" / "SHAPE_AREA" ) * 1000000
```

Der Dialog zur Paramtereingabe sollte wie folgt ausgefüllt sein.



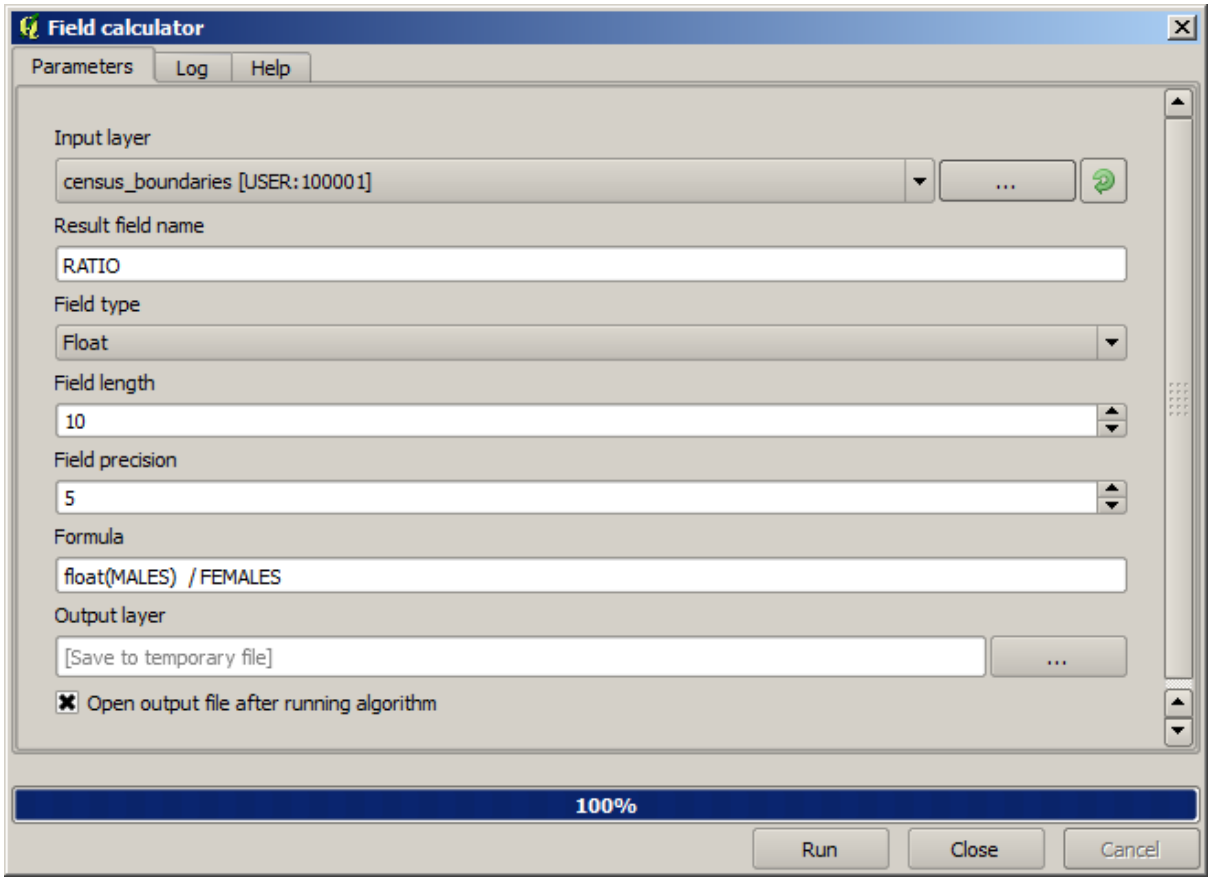
Dadurch wird ein neues Feld `WHITE_DENS` erstellt.

Als nächstes berechnen wir das Verhältnis von `MALES` zu `FEMALES` in einem neuen Feld. Wir ermitteln damit inwieweit die männliche Bevölkerung gegenüber der weiblichen vorwiegt.

Geben Sie die folgende Formel ein

```
"MALES" / "FEMALES"
```

Dieses Mal sollte das Parameterfenster wie folgt aussehen. Drücken Sie nun auf die *OK* Schaltfläche.

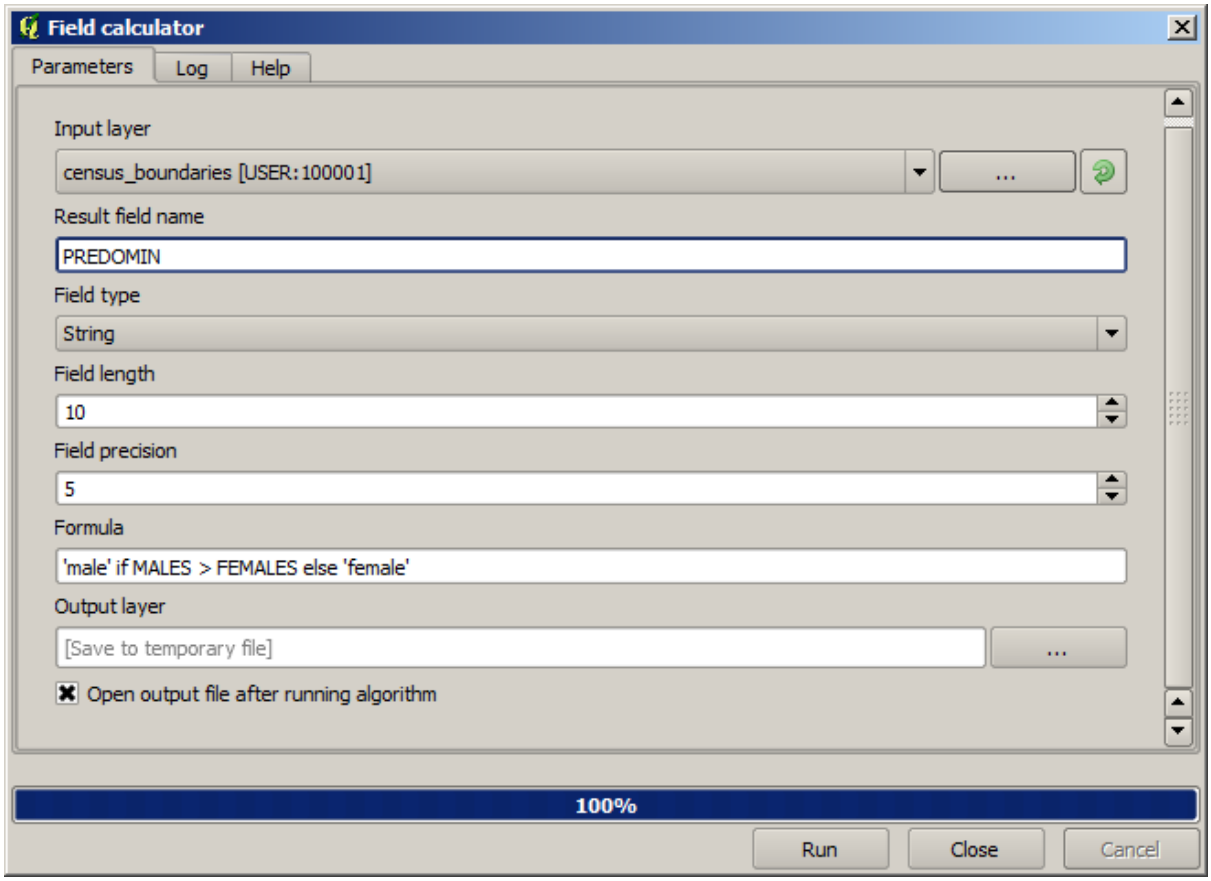


In früheren Versionen wurde das Ergebnis nur als Ganzzahl ausgegeben, da beide Eingabefelder ganzzahlig sind. In diesem Fall muss man folgende Formel verwenden: `1.0 * "MALES" / "FEMALES"`. Damit wird angezeigt, dass das Ergebnis eine Fließkommazahl sein soll.

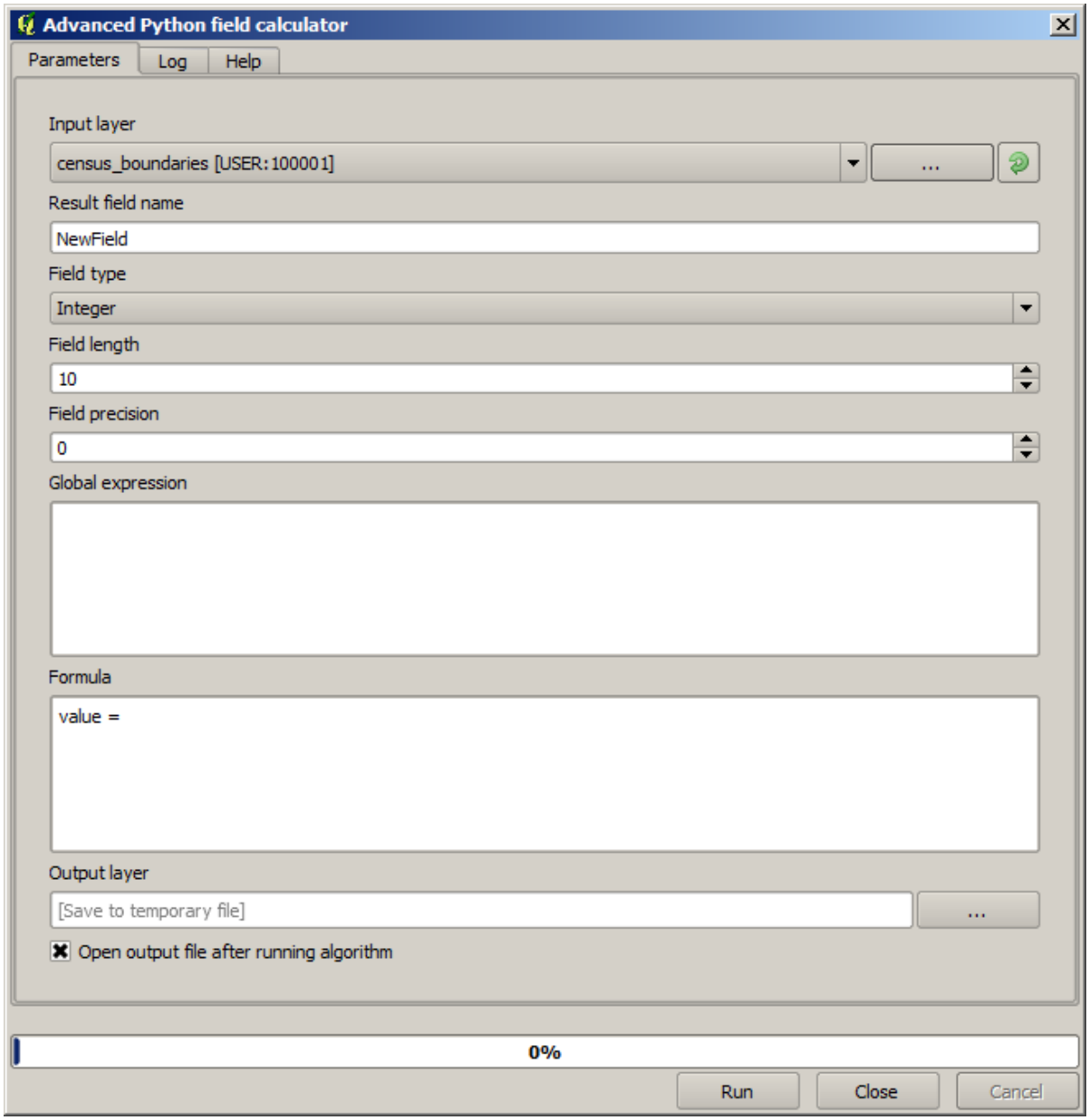
Wir können bedingte Funktionen verwenden, um ein neues Feld zu erzeugen, das anstelle des Verhältnisses mit `male` oder `female` gefüllt wird. Das kann mit der folgenden Formel erfolgen:

```
CASE WHEN  "MALES" > "FEMALES"  THEN 'male' ELSE 'female' END
```

Das Parameterfenster sollte wie folgt aussehen.



Unter *Funktionseditor* steht ein Feldrechner mit Python zur Verfügung. Hierauf gehen wir jetzt nicht näher ein.



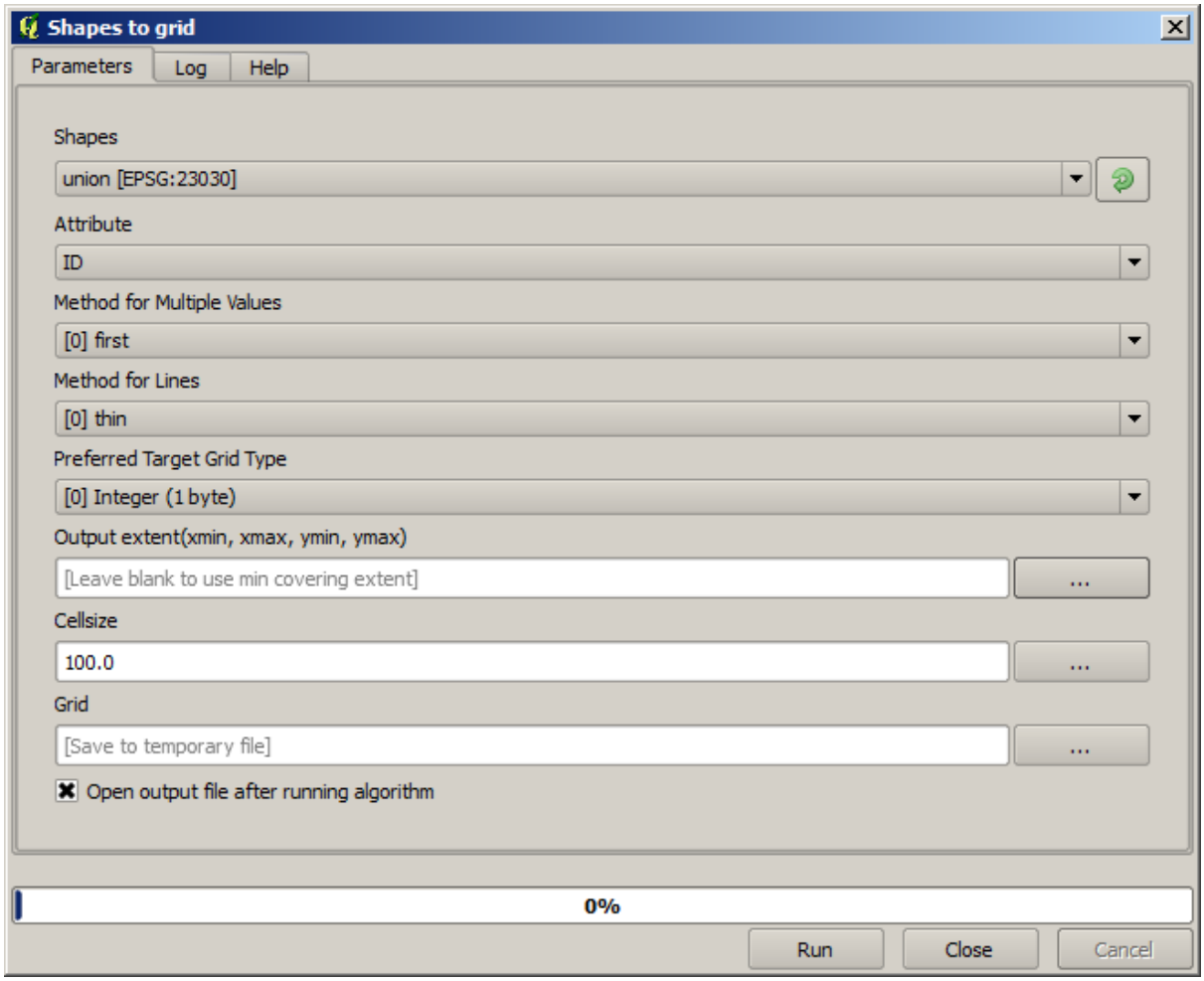
17.12 Definierung von Ausschnitten

Bemerkung: In dieser Lektion werden wir sehen, wie man Ausdehnungen definiert, die von einigen Algorithmen, v.a. den Rasteralgorithmen benötigt werden.

Einige Algorithmen benötigen eine Ausdehnung, um die Fläche zu definieren, die von der Analyse berücksichtigt wird und auch die Ausdehnung des Ausgabelayers bestimmt.

Wenn eine Ausdehnung erforderlich ist, kann sie manuell mit Hilfe der vier Werte die sie definieren vorgegeben werden (min X, min Y, max X, max Y). Es gibt aber auch andere praktikablere und interessantere Möglichkeiten, dies zu tun. Wir werden sie in dieser Lektion kennenlernen.

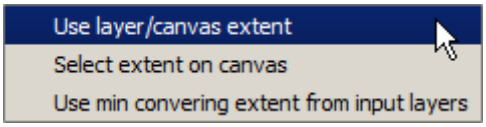
Als Erstes öffnen wir einen Algorithmus, der die Definition einer Ausdehnung voraussetzt. Öffnen Sie den Algorithmus *Rastern*, der einen Rasterlayer aus einem Vektorlayer erstellt.



Alle Parameter bis auf die letzten beiden werden benutzt, um zu bestimmen, welcher Layer gerastert werden soll und wie der Prozess der Rasterung ablaufen soll. Die letzen beiden Paramter bestimmen dagegen die Charakteristik des Ausgabelayers. Sie bestimmen die bedeckte Fläche (die unter Umständen von der Fläche des Eingabevektorlayers abweicht) und die Auflösung/Zellengröße (wird nicht vom Vektorlayer beeinflusst, da Vektorlayer keine Zellengröße verwenden).

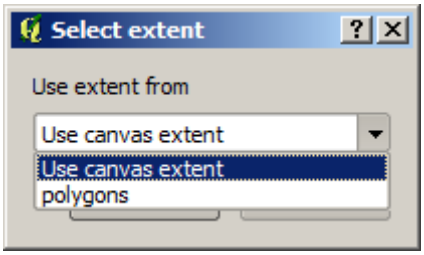
Als Erstes geben wir die 4 bestimmenden oben beschriebenen Werte getrennt durch Komma ein.

Hierfür benötigen wir keine weitere Erklärung. Obwohl das die flexibelste Möglichkeit ist, ist sie manchmal am wenigsten praktikabel. Es gibt es noch andere Alternativen. Man greift über die Schaltfläche auf der rechten Seite des Eingabefeldes darauf zu.



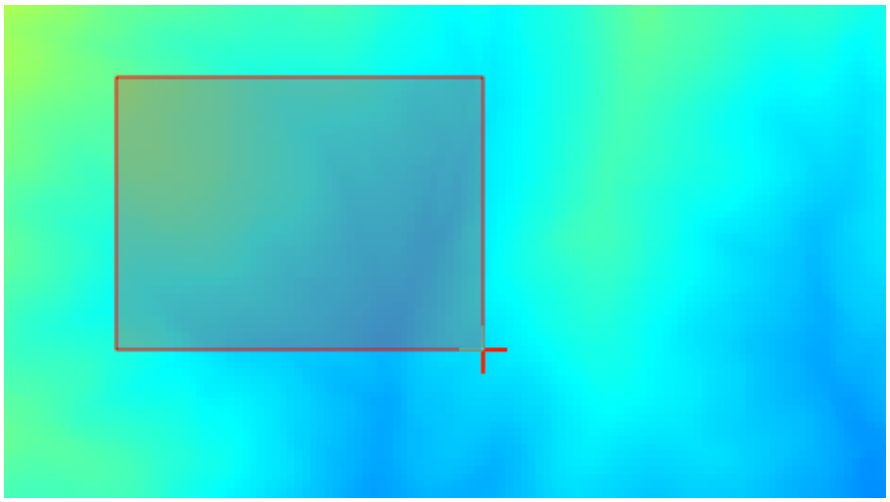
Sehen wir uns an, welche Möglichkeiten es gibt.

Die erste Option lautet *Use layer/canvas extent*, wie im Auswahl Dialog unten zu sehen ist.



Hiermit wird die Ausdehnung des Arbeitsbereiches (die Ausdehnung die der aktuellen Zoomstufe entspricht) oder eines beliebigen verfügbaren Layers gewählt. Wählen Sie den Eintrag, klicken auf *OK* und das Textfeld wird automatisch mit den zugehörigen Werten gefüllt.

Die zweite Option lautet *Ausdehnung auf Kartenausschnitt wählen*. In diesem Fall verschwindet das Auswahlfeld und man kann den gewünschten Ausschnitt im QGIS Arbeitsbereich aufziehen.

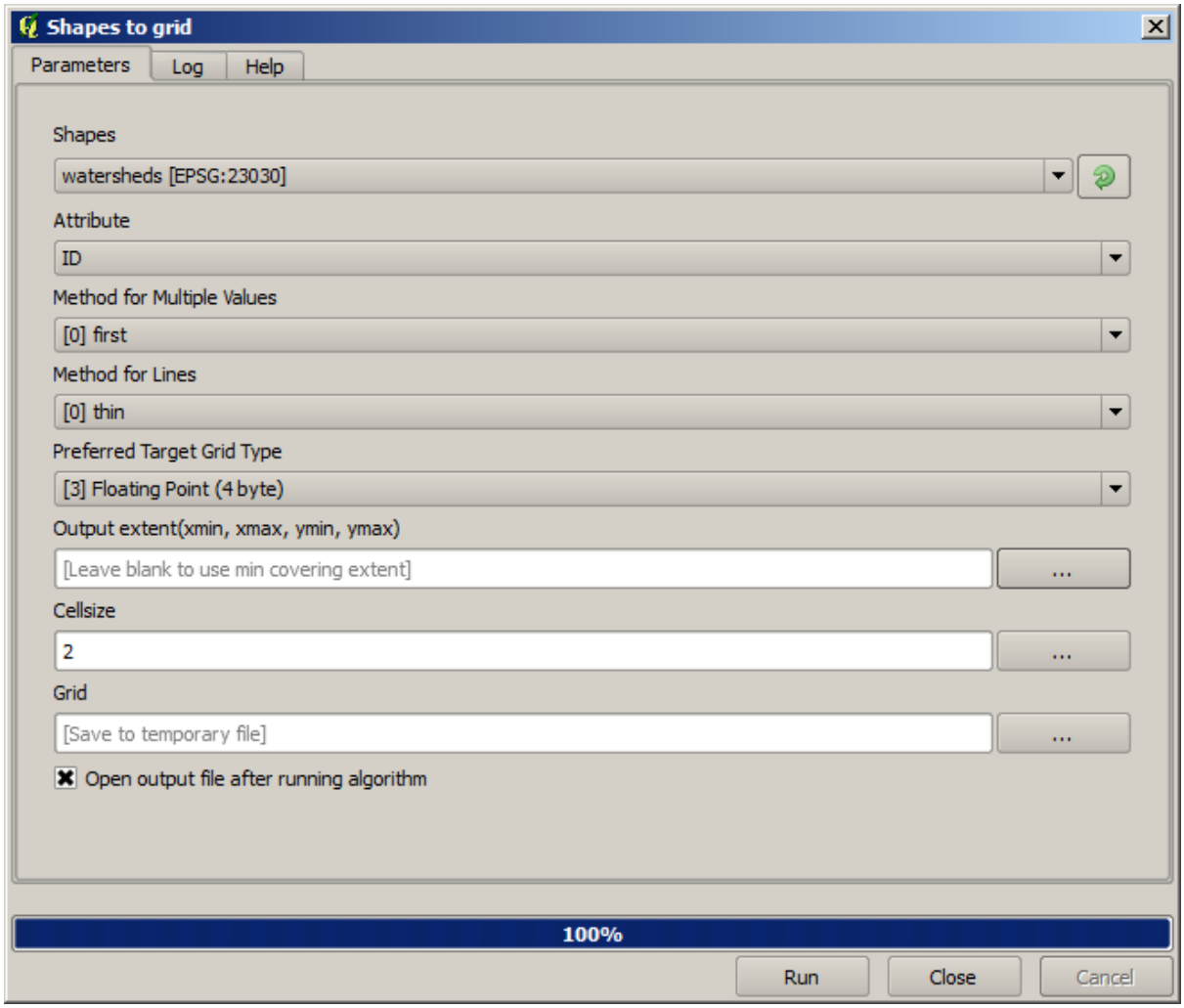


Nachdem man den Ausschnitt aufgezogen hat, erscheint erneut das Dialogfenster und das Textfeld ist mit den zum Ausschnitt gehörenden Werten gefüllt.

Die letzte Option und Standardvorgabe lautet *Use min covering extent from input layers*. Damit wird die minimale Ausdehnung aller Eingabelayer verwendet, eine händische Eingabe in das Textfeld muss nicht mehr erfolgen. Im Fall eines einzelnen Eingabelayers, wie in unserem Beispiel, wird hiermit dasselbe Ergebnis erzielt wie mit *Use layer/canvas extent*, das wir schon beschrieben haben. Wenn dagegen mehrere Eingabelayer verwendet werden, entspricht die Ausdehnung keinem einzelnen Layer. Die Ausdehnung wird dann aus allen Eingabelayern berechnet.

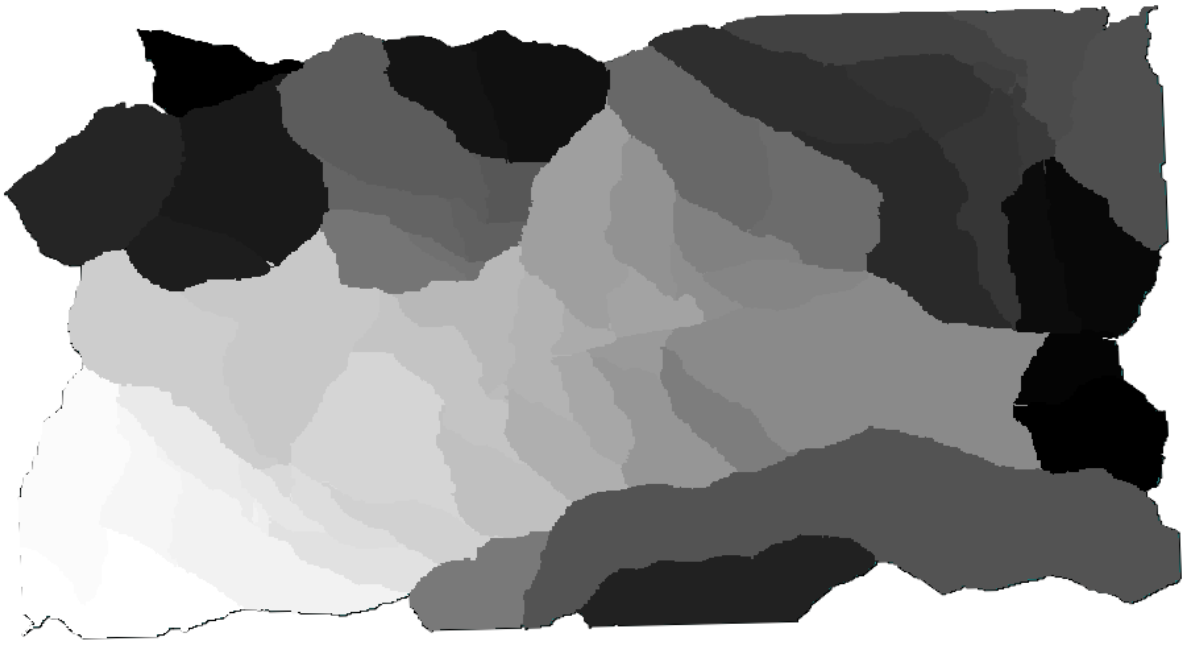
Wir werden für unseren Rasterungsalgorithmus diese letzte Möglichkeit verwenden.

Füllen Sie die Parameter im Dialog wie gezeigt aus und drücken auf *Starte*.



Bemerkung: In diesem Fall ist es besser, *Integer (1 byte)* anstelle von *Floating point (4 byte)* zu verwenden, da *NAME* ganzzahlig mit einem Maximalwert=64 ist. Die Berechnung läuft dann schneller ab und die Ergebnisdatei ist kleiner.

Wir erhalten einen gerasterten Layer, der exakt die Fläche des originalen Vektorlayers abdeckt.



In manchen Fällen ist die letzte Option *Use min covering extent from input layers* nicht verfügbar. Das passiert, wenn der Algorithmus keine Eingabelayer sondern nur Parameter mit anderem Typ verwendet. In diesem Fall müssen Sie die Werte händisch eingeben oder eine der anderen Möglichkeiten verwenden.

Beachten Sie, das eine Auswahl nicht bei der Berechnung der Ausdehnung berücksichtigt wird auch wenn die Rasterung nur mit den gewählten Objekten erfolgt. Die Ausdehnung bezieht sich auf alle Objekte des Layers. In so einem Fall, ist es günstiger einen Layer aus der Auswahl zu erstellen.

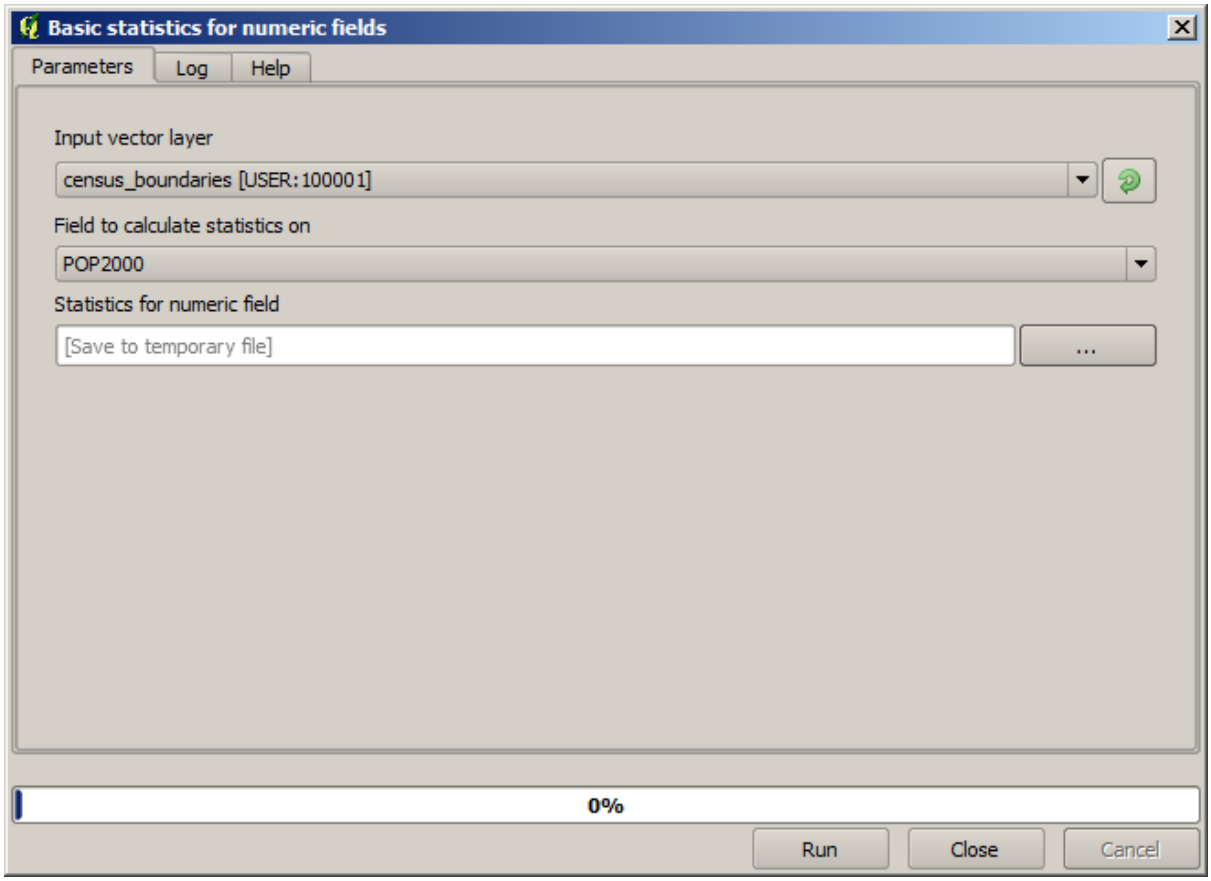
17.13 HTML-Ausgaben

Bemerkung: In dieser Lektion lernen wir, wie QGIS mit Ausgaben im HTML Format umgeht, die zur Produktion von Textausgaben und Grafiken verwendet wird.

Alle Ausgaben, die wir bisher erzeugt haben, waren Layer (Raster- oder Vektorlayer). Es gibt aber auch Algorithmen, die Ausgaben in Form von Text und Grafiken erzeugen. Alle diese Ausgaben erfolgen in HTML-Dateien, die in der sogenannten *Ergebnisanzeige*, einem weiteren Teil der Verarbeitungsumgebung, angezeigt werden.

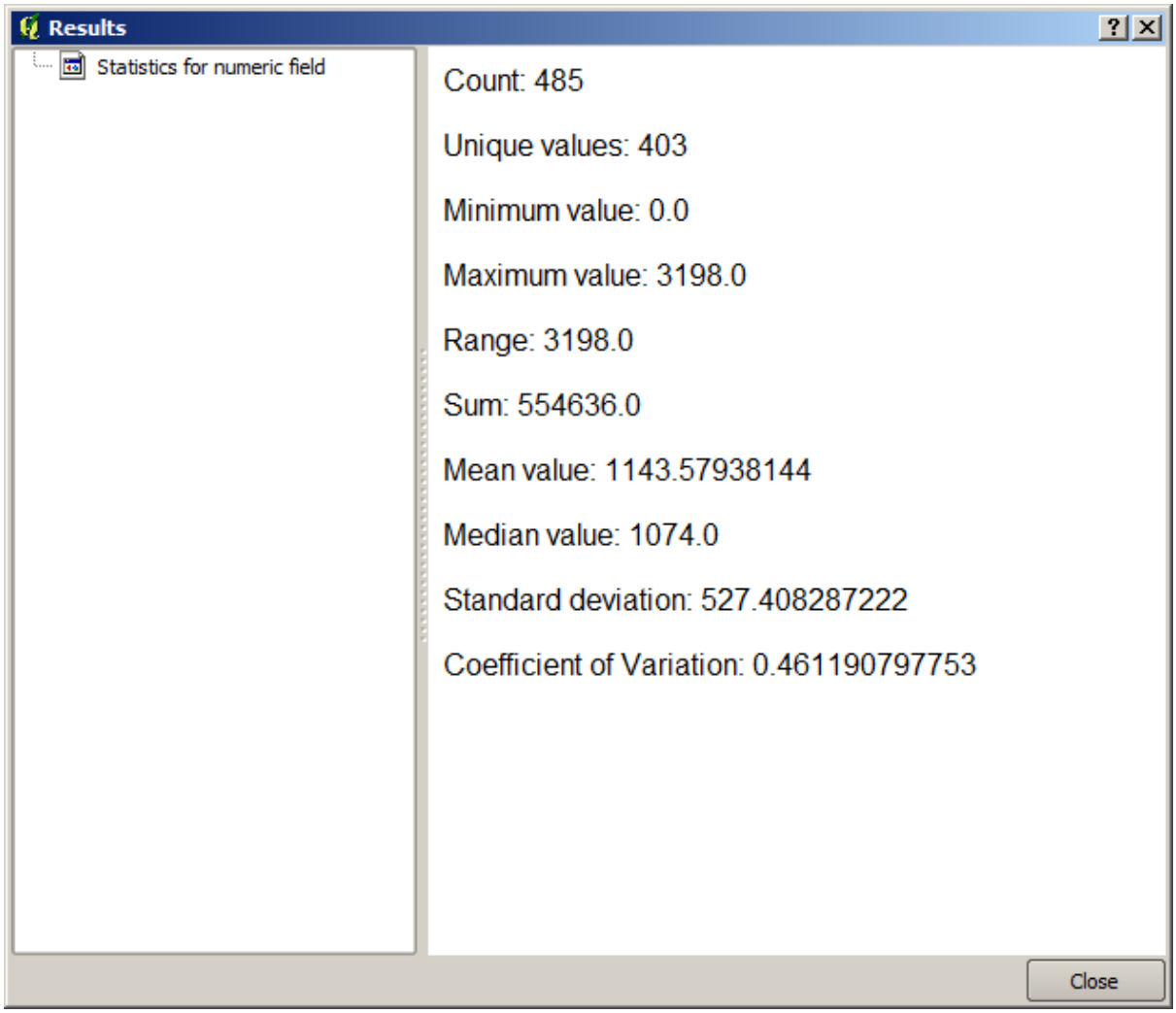
Sehen wir uns einige dieser Algorithmen an, um zu verstehen wie sie arbeiten.

Öffnen Sie das Projekt, dass die Daten für diese Lektion enthält. Öffnen Sie dann den Algorithmus *Grundstatistik für Felder*.



Der Algorithmus ist relativ einfach, man muss nur den zu verwendenden Layer und ein Feld (ein numerisches) auswählen. Die Ausgabe hat den Typ HTML aber die zugehörige Box arbeitet auf dieselbe Art wie bei der Ausgabe von Vektor- oder Rasterlayern. Sie können einen Dateipfad eingeben oder das Feld freilassen, um in eine temporäre Datei zu speichern. In diesem Fall sind nur die Dateierweiterungen `html` und `htm` erlaubt, so dass man das Ausgabeformat nicht ändern kann.

Starten Sie den Algorithmus mit dem einzigen Layer im Projekt und dem Feld *POP2000* als Eingabewerte. Nach der Ausführung des Algorithmus und dem Schließen des Parameterdialogs erscheint ein neues Fenster wie hier gezeigt.



Das ist die *Ergebnisanzeige*. Alle HTML Resultate die in der aktuellen Sitzung generiert wurden sind so leicht verfügbar und man kann sie bei Bedarf immer schnell aufrufen. Genauso wie bei temporären Layern werden die Resultate bei Schließen von QGIS gelöscht. Wenn Sie in einem Pfad gespeichert haben, wird die Datei nicht gelöscht. Sie wird aber beim nächsten Start von QGIS nicht in der *Ergebnisanzeige* angezeigt.

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 model designer, you will see that it will allow us to use those values as numeric inputs for other algorithms.

17.14 Erstes Analysebeispiel

Bemerkung: In dieser Lektion werden wir einige echte Analysen ausführen. Wir benutzen die Werkzeugkiste, so dass wir uns noch mehr mit der Verarbeitungsumgebung vertraut machen.

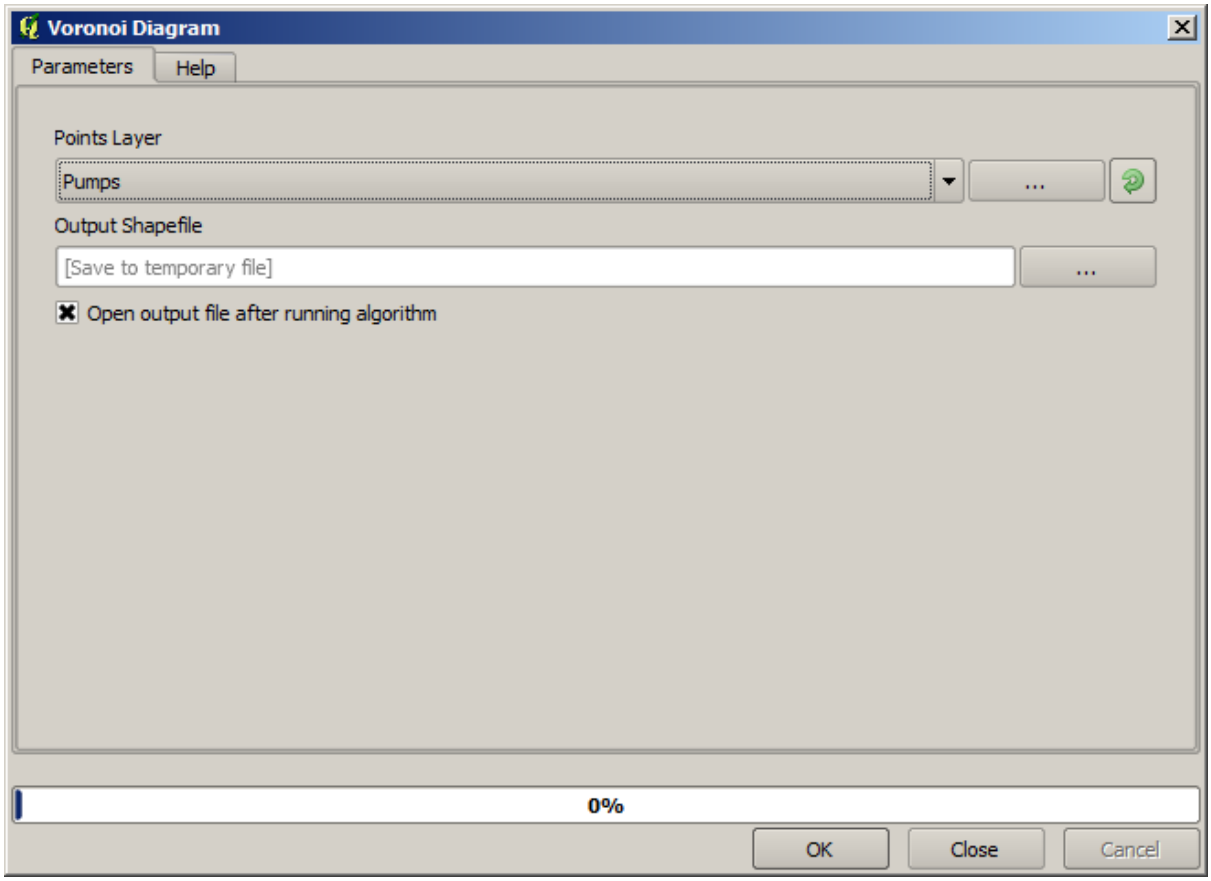
Da jetzt alles richtig konfiguriert ist und wir externe Algorithmen verwenden können, steht uns ein mächtiges Werkzeug zur Durchführung räumlicher Analysen zur Verfügung. Es ist an der Zeit ein echtes Beispiel mit echten Daten zu bearbeiten.

Wir werden den gut bekannten Datensatz verwenden, den John Snow 1854 in seiner wegweisenden Arbeit (https:

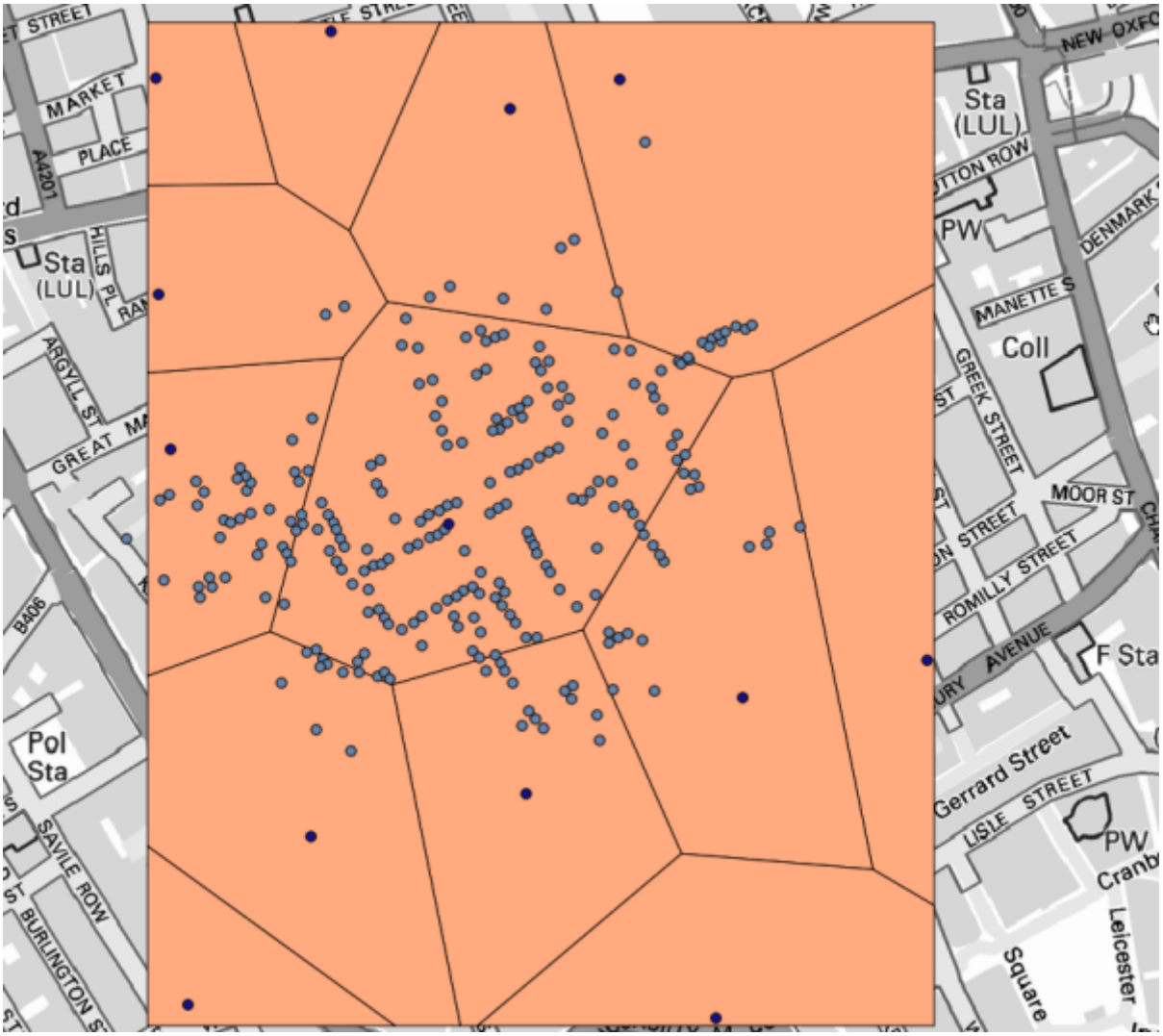
//en.wikipedia.org/wiki/John_Snow_%28physician%29) nutzte und werden zu einigen interessanten Ergebnissen kommen. Die Analyse der Daten ist ziemlich offensichtlich und man benötigt dazu keine anspruchsvollen GIS-Techniken, um zu guten Resultaten und Schlussfolgerungen zu gelangen. Es ist aber eine gute Möglichkeit, zu zeigen, wie man räumliche Fragestellungen mit Hilfe verschiedener Verarbeitungswerkzeuge analysieren und lösen kann.

Der Datensatz enthält shape-Dateien mit den Todesfällen an Cholera und der Lage von Pumpen sowie eine nach OSM gerenderte Karte im TIFF Format. Öffnen Sie das zu dieser Lektion gehörende QGIS Projekt.

Der erste Schritt ist die Erstellung eines Voronoi Diagramms (bzw. Thiessen-Polygone) für den Layer der Pumpen, um die Einflusszone für jede Pumpe zu ermitteln. Der Algorithmus *Voronoi-Polygone* kann dazu verwendet werden.

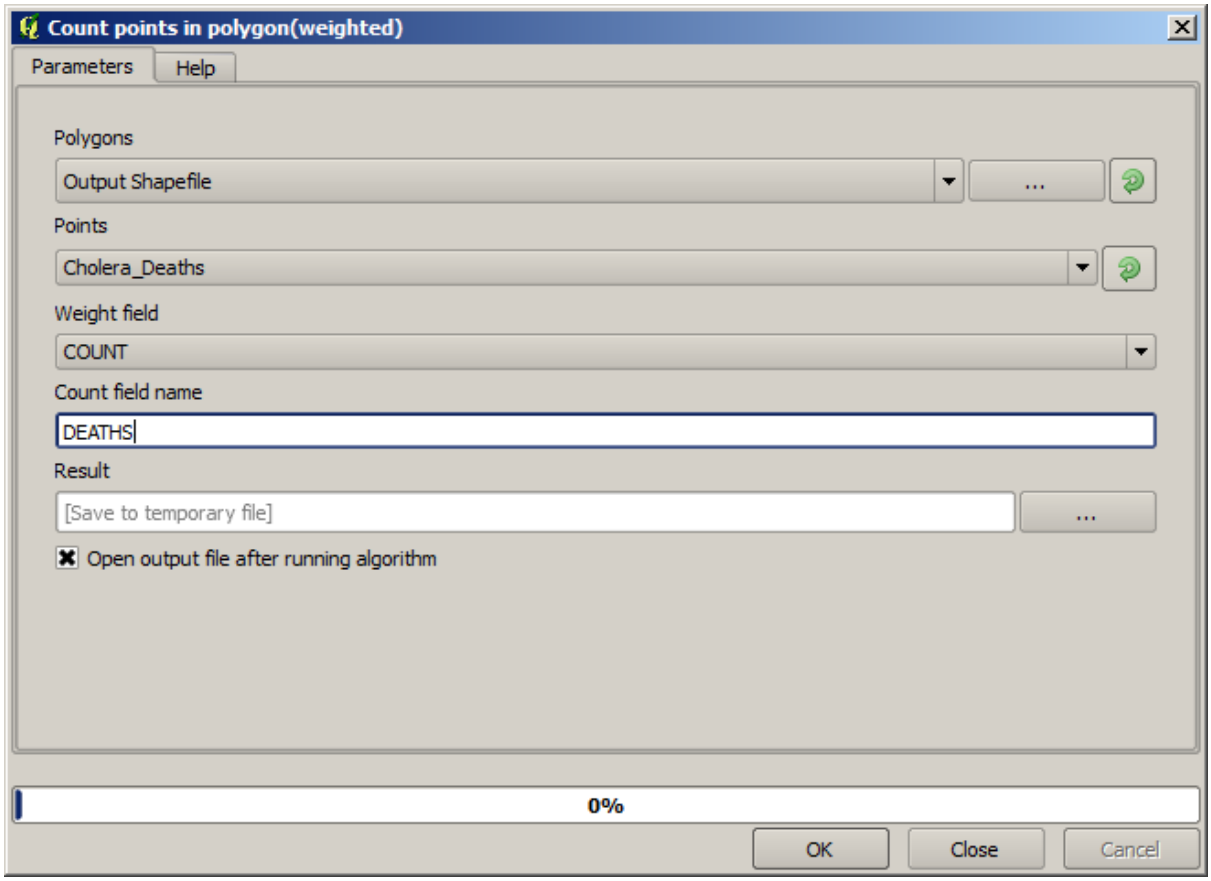


Ganz einfach - aber wir erhalten schon erste interessante Informationen.

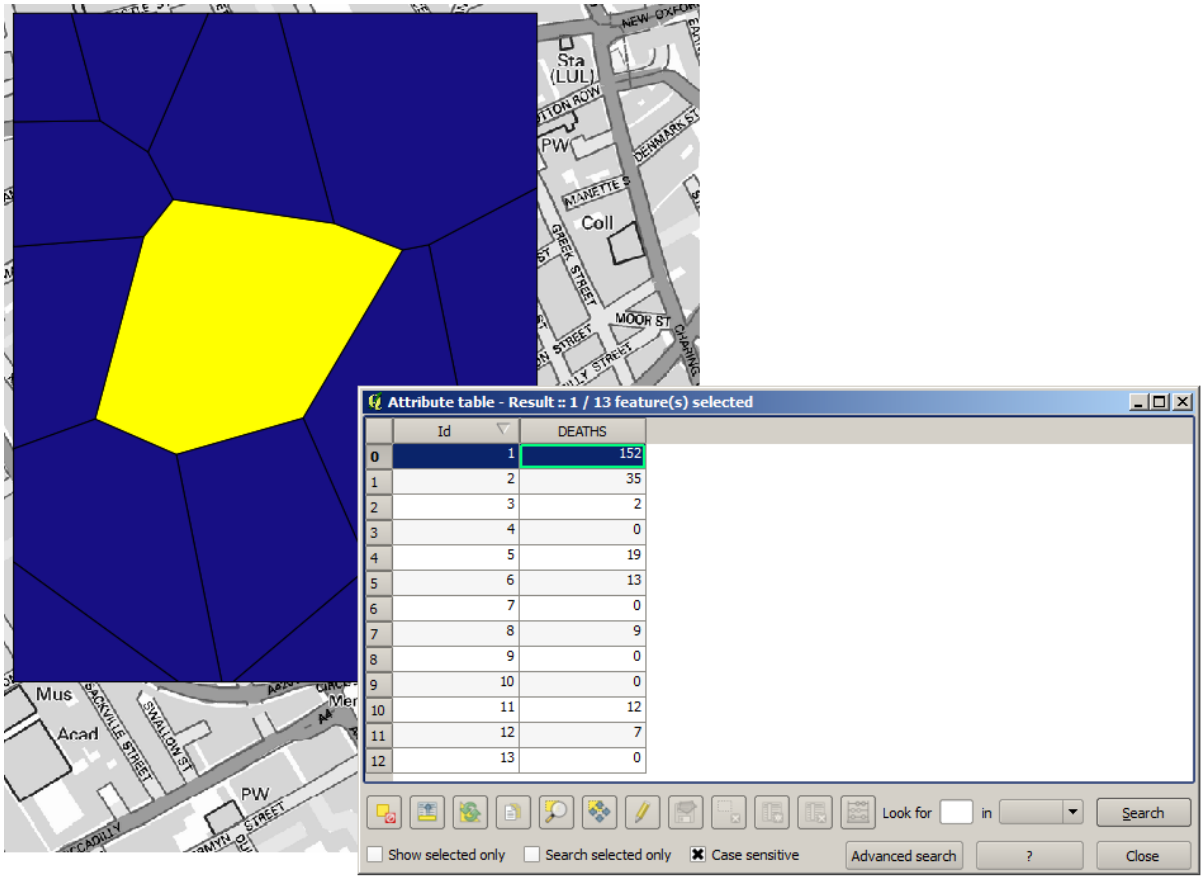


Die meisten Fälle liegen offensichtlich innerhalb eines der Polygone.

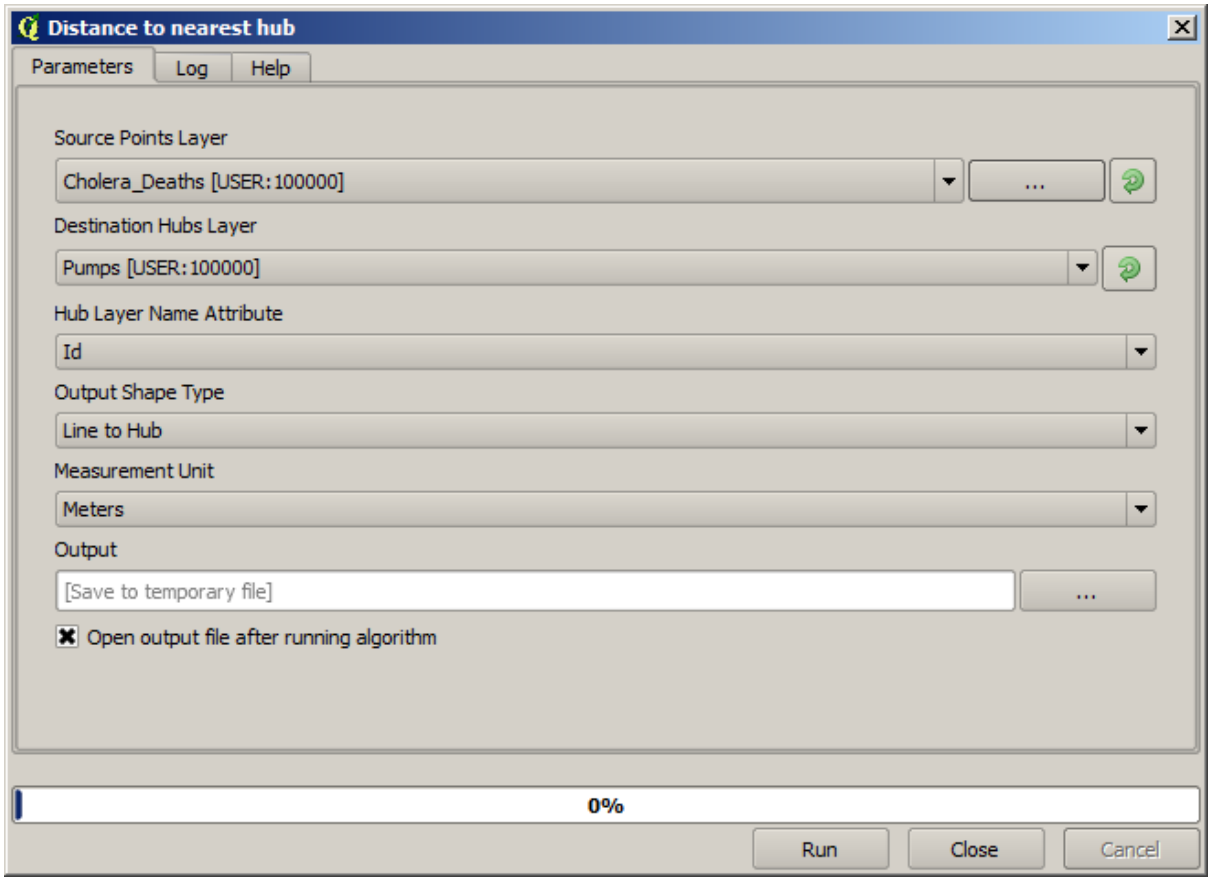
Um ein genaueres Ergebnis zu erhalten, zählen wir die Todesfälle in jedem Polygon. Jeder Punkt steht für ein Gebäude in dem Todesfälle auftraten. Da die Anzahl der Todesfälle in der Attributtabelle gespeichert ist, reicht es nicht, nur die Punkte zu zählen. Wir benötigen eine gewichtete Zählung. Dazu können wir das Werkzeug *Punkte in Polygon zählen* verwenden.



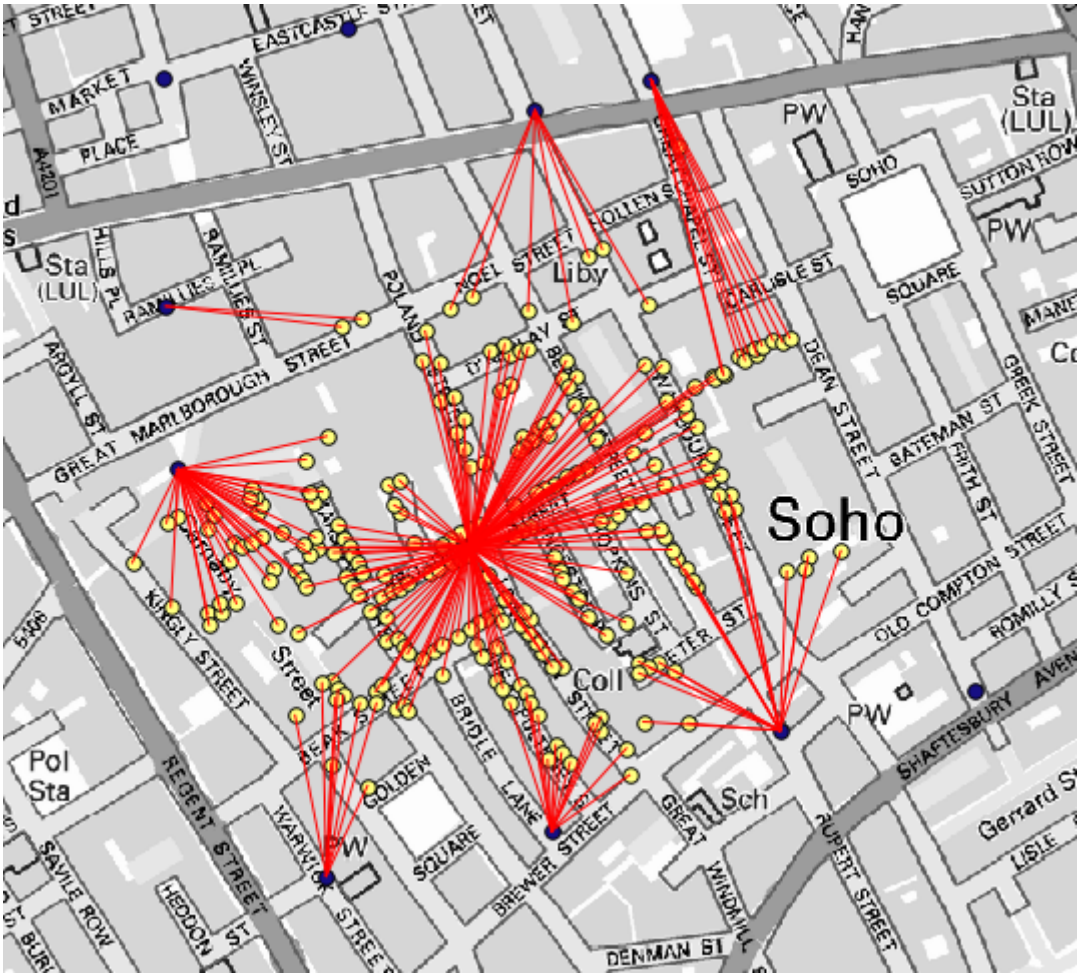
Das neue Feld wird *DEATHS* genannt und wir nutzen das Feld *COUNT* als Wichtungsfeld. Die Ergebnistabelle zeigt eindeutig, dass die Anzahl der Todesfälle im zur ersten Pumpe gehörenden Polygon größer ist, als in den anderen Polygonen.



Ein anderer guter Weg zur Visualisierung der Abhängigkeit jedes Punktes aus dem Layer `Cholera_deaths` zu den Punkten im Layer `Pumps` ist die Erstellung einer Linie zu der am nächsten gelegenden Pumpe. Dies kann mit Hilfe des Werkzeuges *Abstand zum nächsten Knoten (Linie zu Knoten)* und der unten aufgeführten Konfiguration erfolgen.

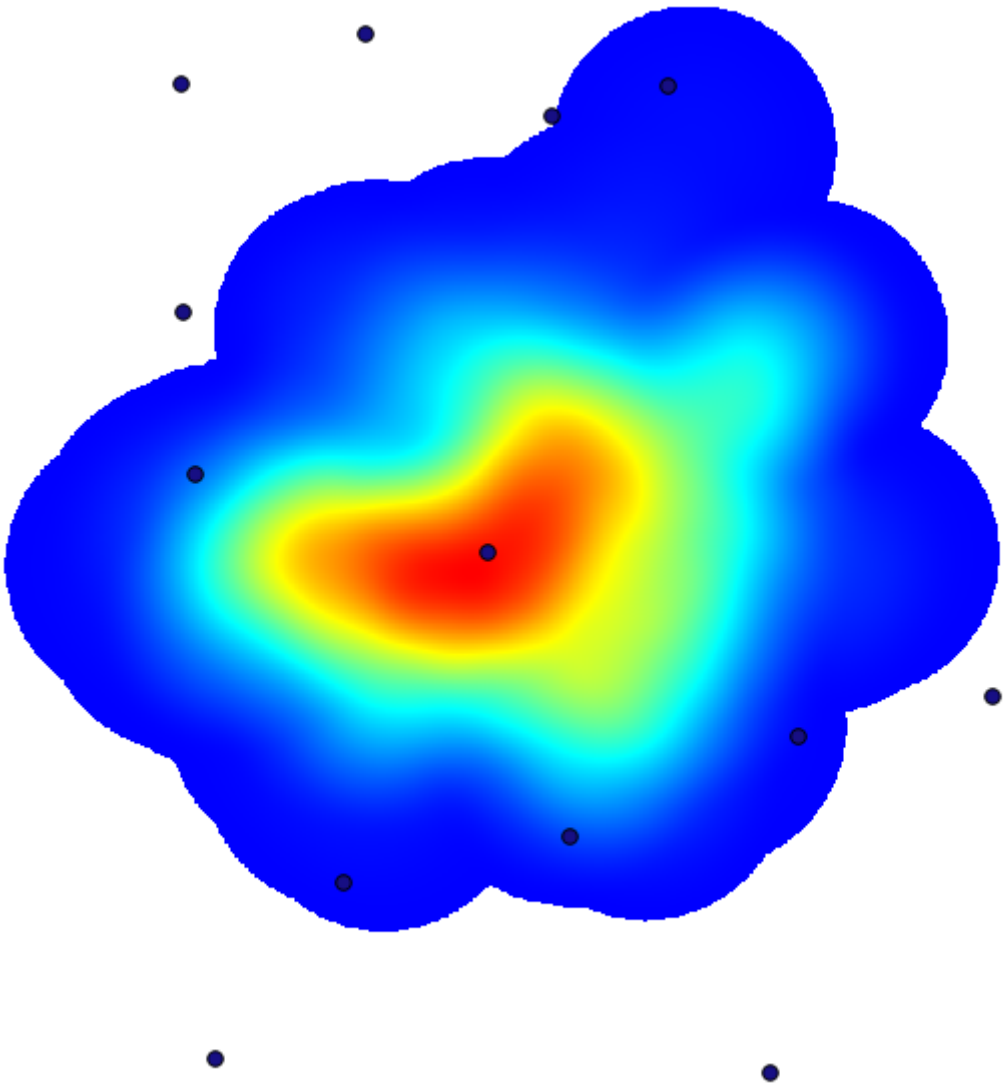


Das Ergebnis sieht wie folgt aus:

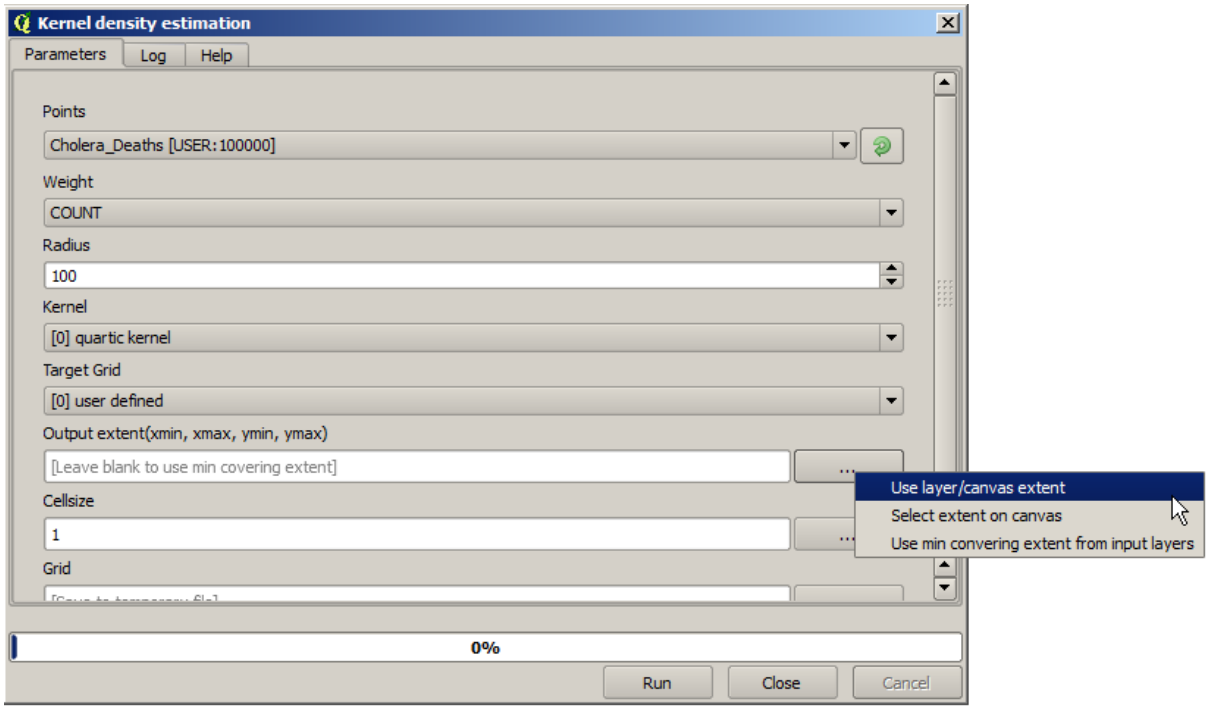


Obwohl die Anzahl der Linien zur zentralen Pumpe größer ist, dürfen wir nicht vergessen, dass die Anzahl der Linien nicht der Anzahl der Todesfälle sondern der Anzahl der Orte an denen Todesfälle auftraten, entspricht. Es ist zwar ein repräsentativer Parameter, allerdings ohne Berücksichtigung der mehrfachen Todesfälle an einigen Orten.

Ein Dichtelayer kann das ebenfalls sehr gut verdeutlichen. Wir können ihn mit Hilfe des Algorithmus *Heatmap (Kerndichtenschätzung)* erstellen. Wenn wir den Layer *Cholera_deaths*, das *COUNT* Feld als Wichtungsfeld mit einem Radius von 100 und die Ausdehnung des Rasterlayers der Straßen verwenden, erhalten wir in etwa die folgende Ausgabe.



Denken Sie daran, dass Sie die Ausdehnung der Ausgabe nicht eintippen müssen. Klicken Sie auf die Schaltfläche auf der rechten Seite und wählen *Use layer/canvas extent*.



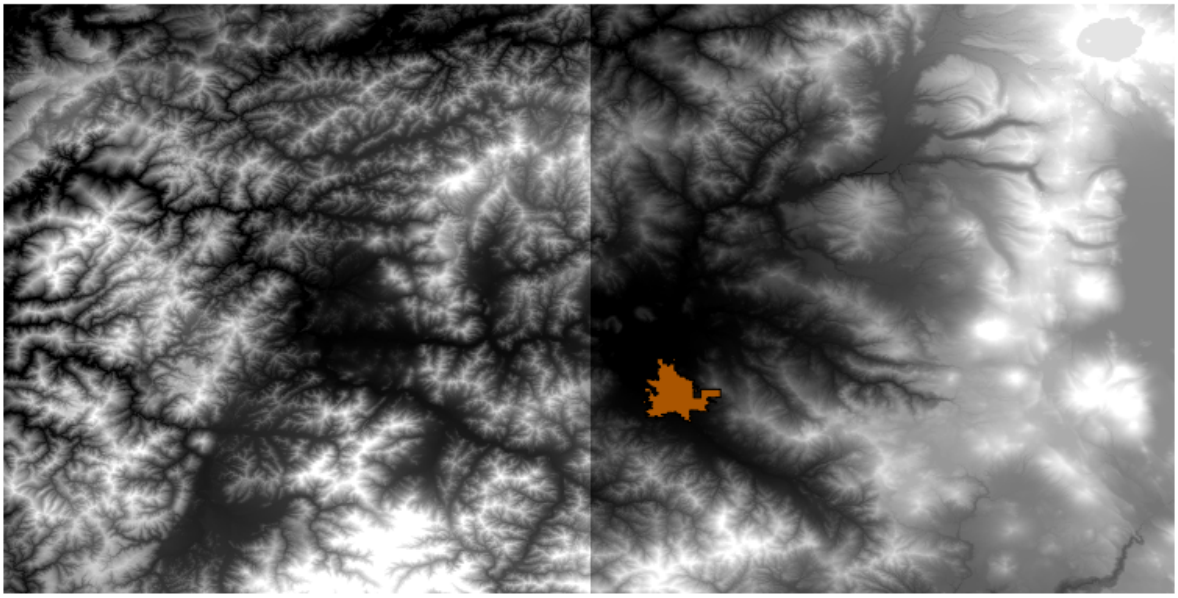
Wählen Sie den Rasterlayer der Straßen, seine Ausdehung wird dann automatisch in das Textfeld übernommen. Mit der Zellgröße können wir genauso vorgehen, d.h. wir wählen diesen Layer erneut.

Zusammen mit dem Layer der Pumpen erkennen wir, dass sich eindeutig eine Pumpe im Zentrum der höchsten Dichte der Todesfälle befindet.

17.15 Zuschneiden und Verschmelzen von Rasterlayern

Bemerkung: In dieser Lektion lernen wir ein weiteres Beispiel zur Vorbereitung realer räumlicher Daten mit Hilfe von Geoalgorithmen.

In dieser Lektion werden wir einen Gefällelayer für das Umland eines Stadtgebietes bestimmen. Das Stadtgebiet liegt als einzelnes Polygon in einem Vektorlayer vor. Das Basis DEM ist in zwei Rasterlayer unterteilt, die ein viel größeres Gebiet abdecken, als unser Arbeitsgebiet im Umland der Stadt. Nach dem Öffnen des zur Lektion gehörenden Projektes, sehen wir in etwa das Folgende.



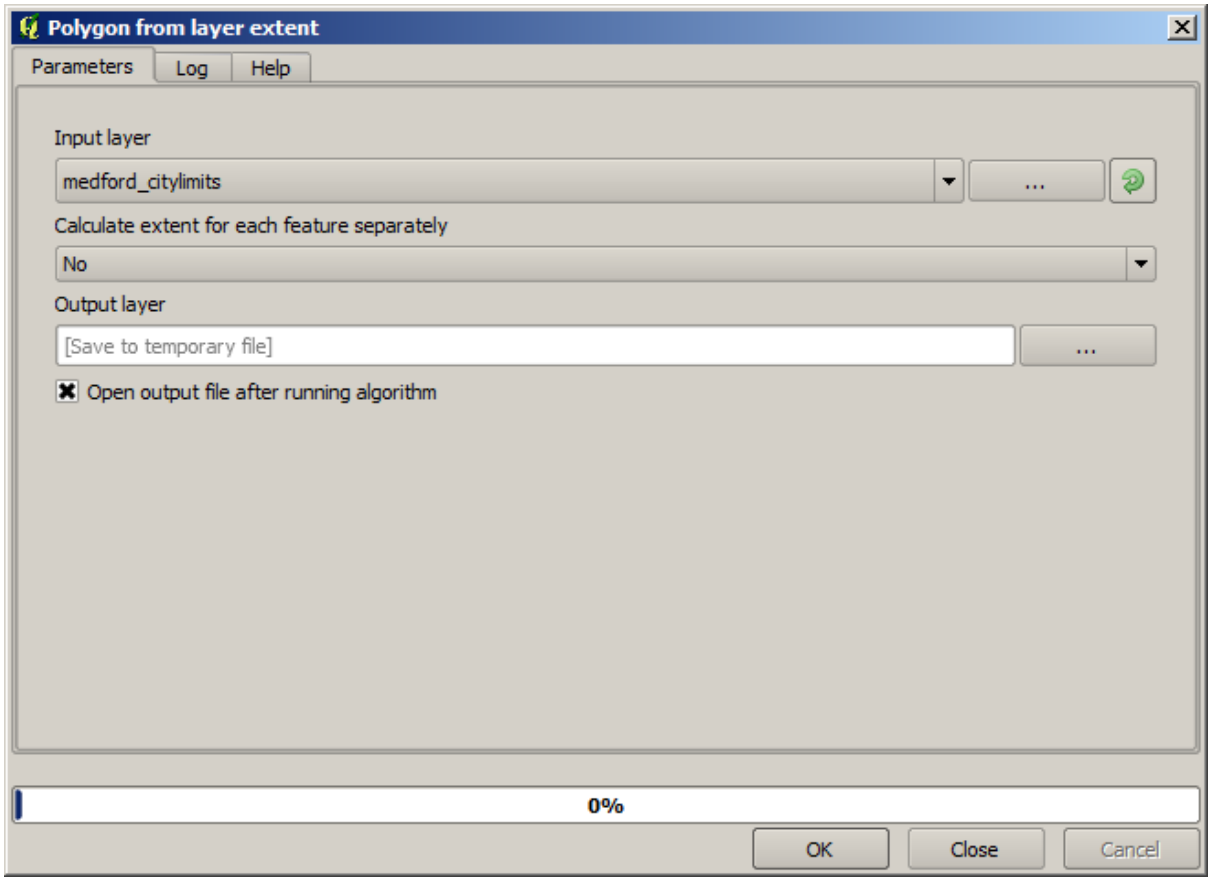
Bei diesen Layern gibt es zwei Probleme:

- Sie überdecken ein für unsere Zwecke zu großes Gebiet (uns interessiert nur eine kleine Region um das Stadtzentrum herum)
- Sie liegen in zwei verschiedenen Dateien vor (die Stadtgrenze liegt nur in einem Rasterlayer, wir benötigen aber eine etwas größere Fläche außerhalb der Stadt).

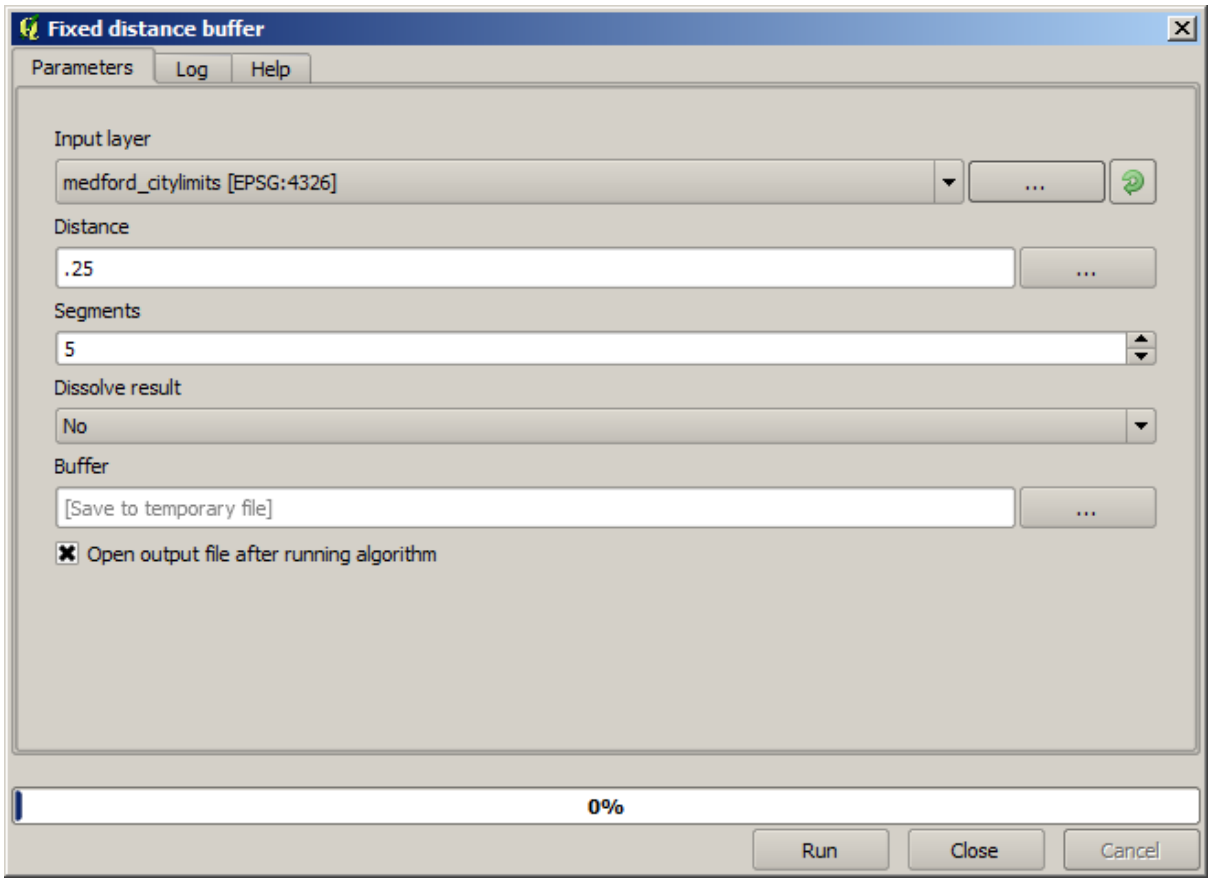
Beide Probleme lassen sich leicht mit dem passenden Geoalgorithmus beheben.

Als Erstes erstellen wir ein Rechteck, das unser Untersuchungsgebiet bestimmt. Wir erstellen dazu einen Layer, der den Begrenzungsrahmen der Stadtfläche darstellt. Um diesen Layer erstellen wir einen Puffer, damit wir einen Rasterlayer erhalten, der etwas mehr als die erforderliche Fläche abdeckt.

Um den Begrenzungsrahmen zu erstellen, können wir den Algorithmus *Layerausdehnung extrahieren* verwenden.

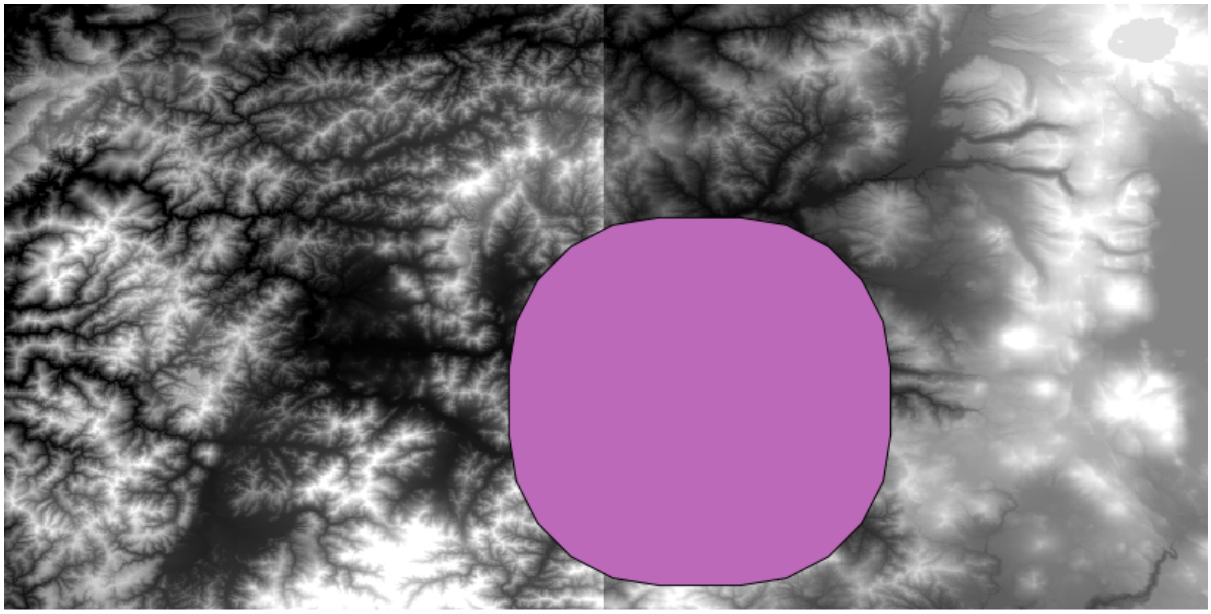


Zur Erstellung des Puffers verwenden wir den Algorithmus *Puffer* mit den folgenden Parametern.

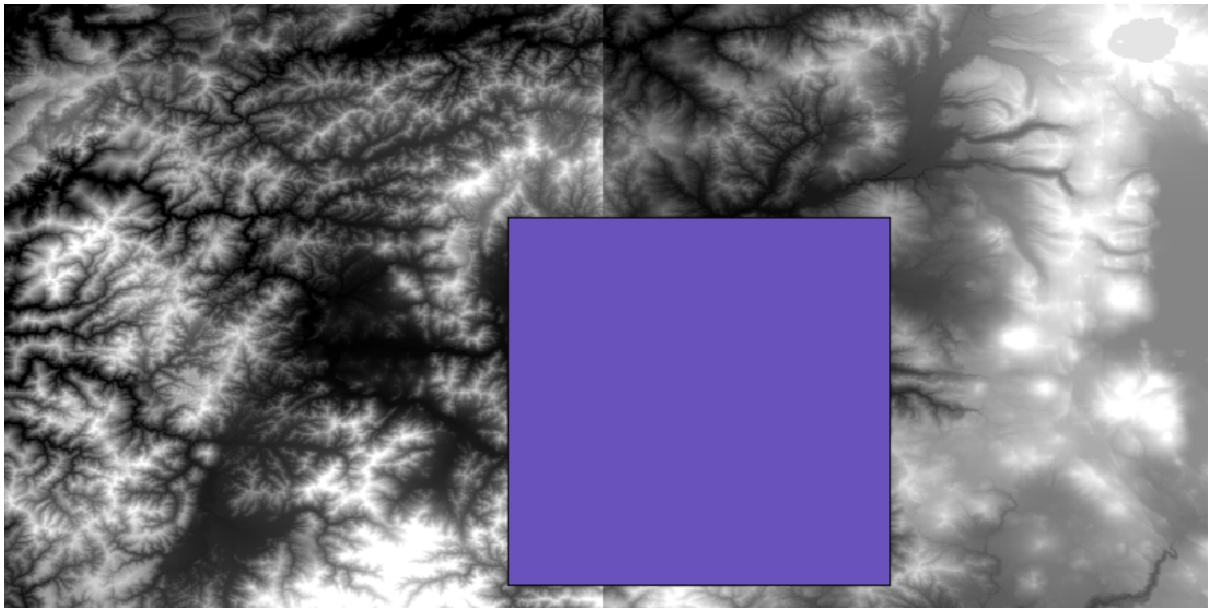


Warnung: Syntax in neueren Versionen geändert; sowohl Abstand als auch Bogen-Eckpunkt auf .25 setzen

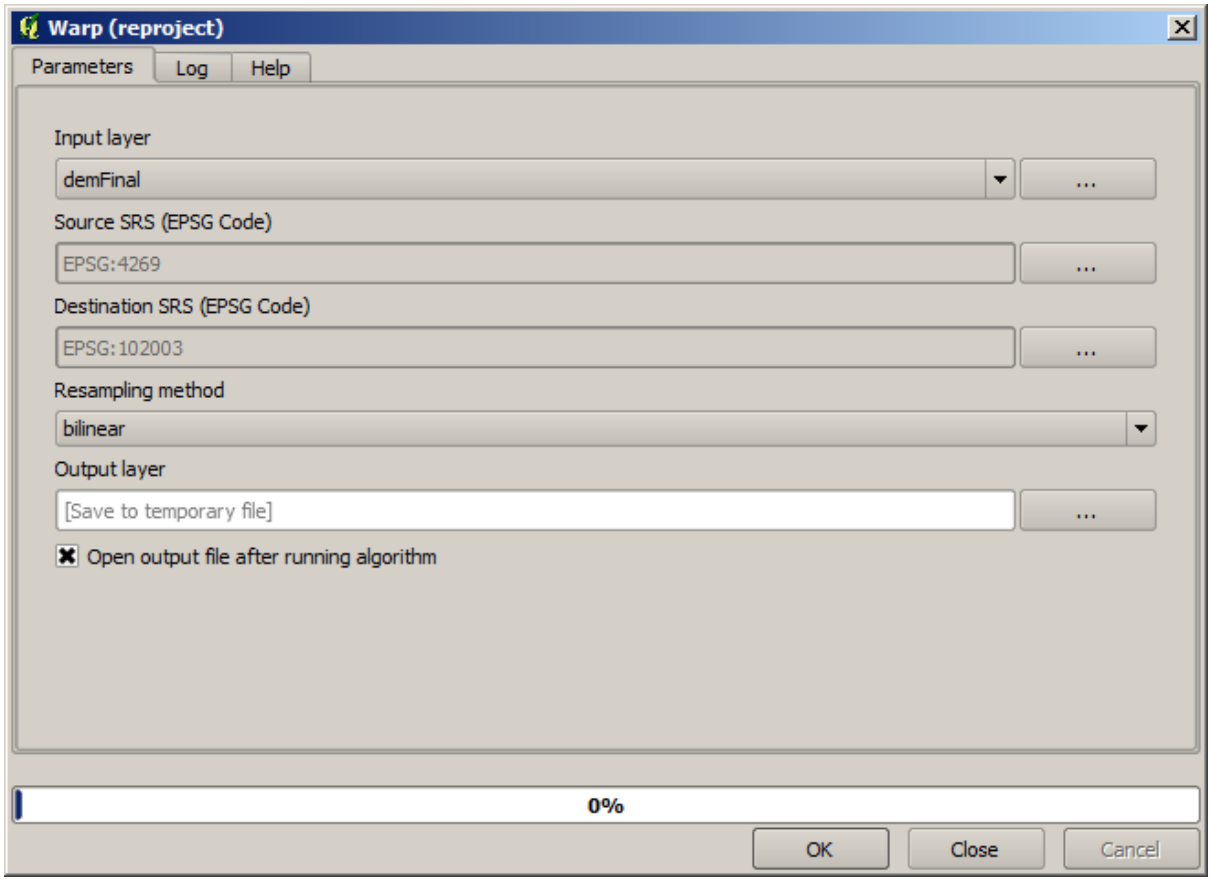
Hier sehen Sie den Begrenzungsrahmen, der mit den gegebenen Parametern erzeugt wird.



Es ist ein abgerundeter Rahmen. Wenn wir den Algorithmus *Layerausdehnung extrahieren* auf ihn anwenden, erzeugen wir daraus einen rechteckigen Rahmen. Wir hätten auch zuerst die Stadtgrenzen puffern und darauf basierend den Begrenzungsrahmen erstellen können. Dadurch hätten wir einen Schritt gespart.

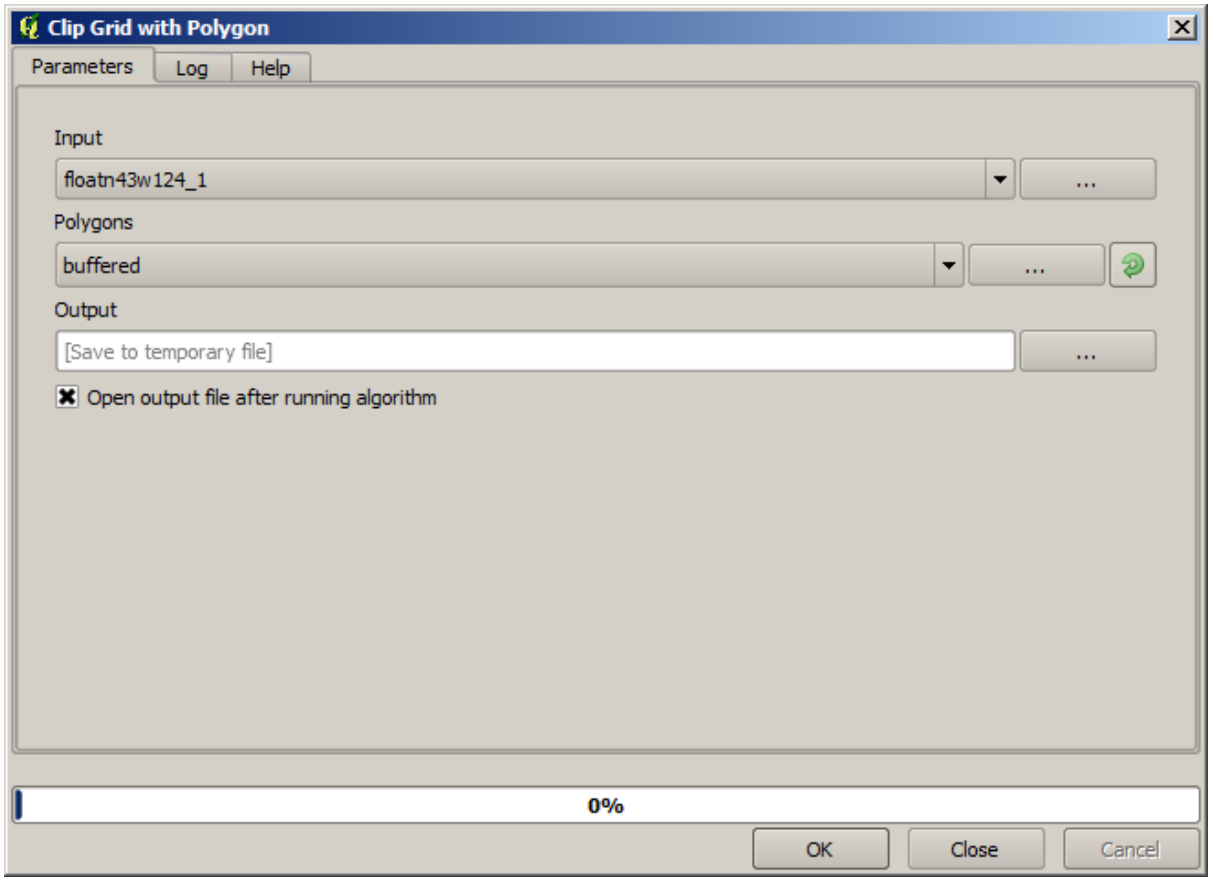


Die Rasterlayer verwenden eine andere Projektion als der Vektorlayer. Wir müssen sie bevor wir fortfahren mit Hilfe des Werkzeuges *Transformieren (reprojizieren)* transformieren.

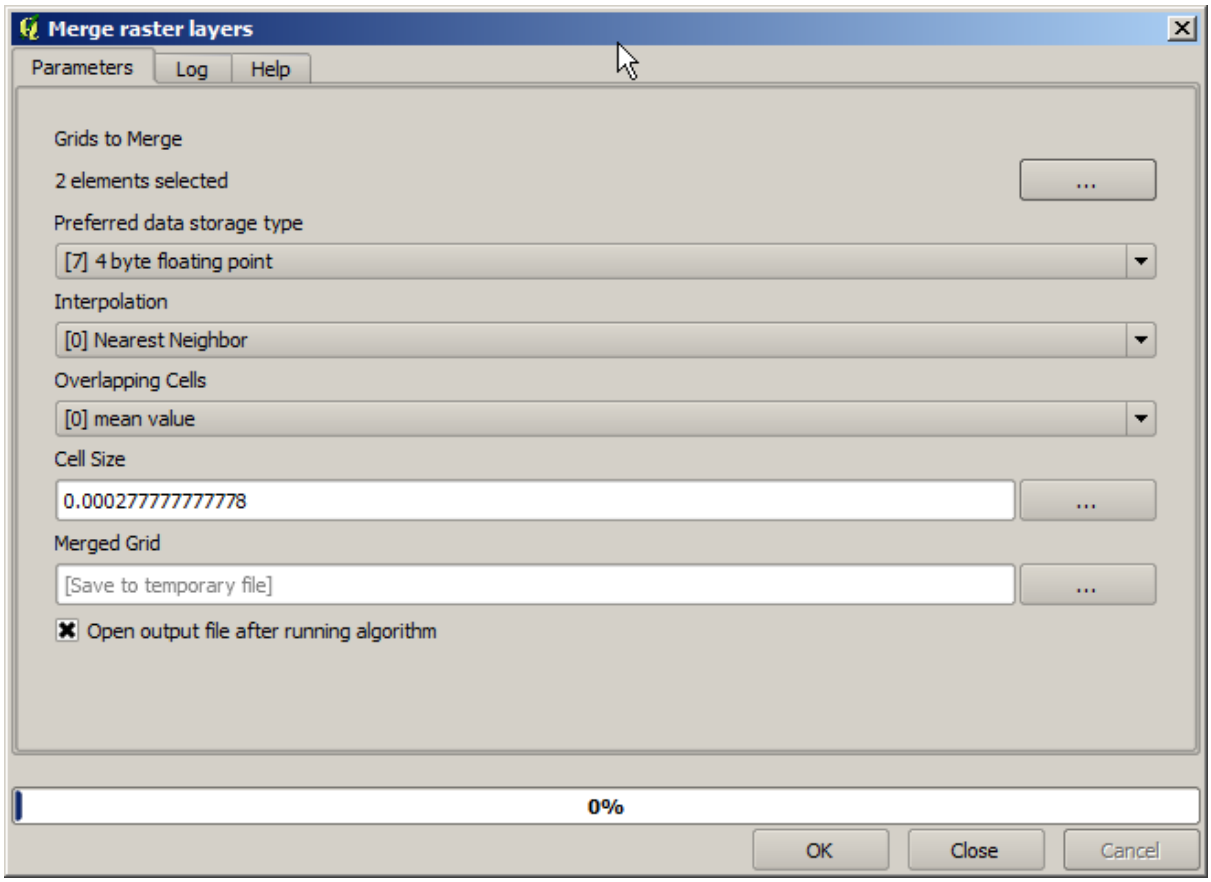


Bemerkung: Neuere Versionen verwenden eine komplexere Schnittstelle. Stellen Sie sicher, dass mindestens eine Komprimierungsmethode gewählt wurde.

Mit Hilfe des Layers des Begrenzungsrahmens schneiden wir beide Rasterlayer zu. Wir verwenden dazu das Werkzeug *Raster auf Layermaske zuschneiden*.

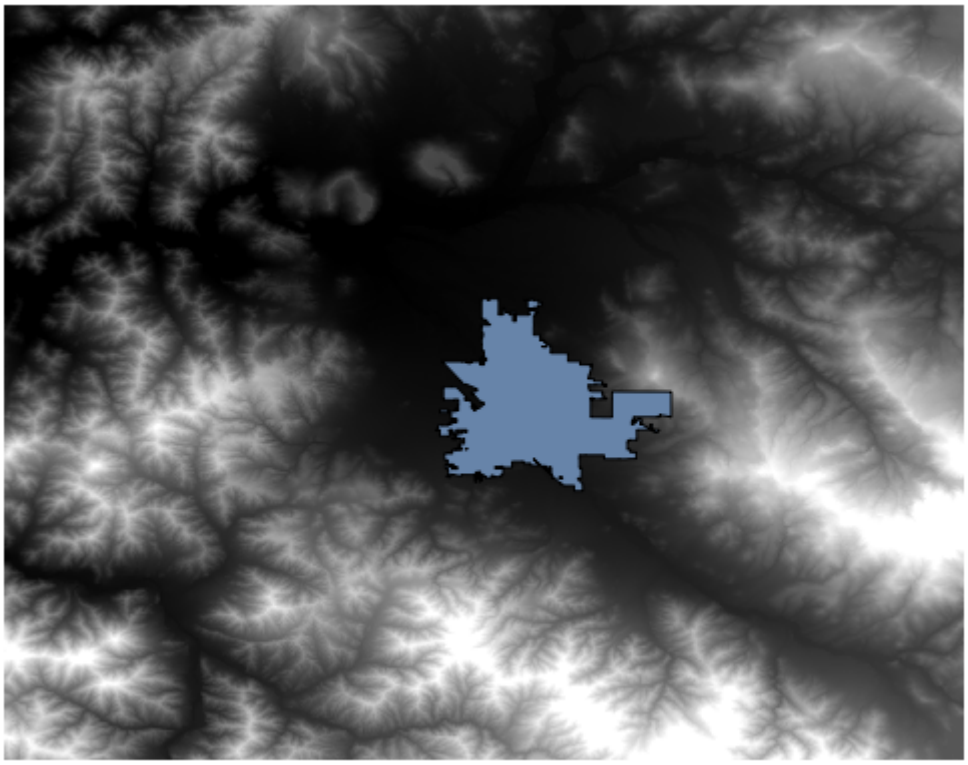


Once the layers have been cropped, they can be merged using the SAGA *Mosaic raster layers* algorithm.



Bemerkung: Man kann Zeit sparen, wenn man zuerst verschmelzt und danach zuschneidet. Der Zuschneidealgorithmus muss dann nicht zweimal aufgerufen werden. Wenn man mehrere große Layer zusammenfügen möchte, kann allerdings ein ziemlich großer Layer entstehen, der später schwer zu verarbeiten ist. In diesem Fall ist es günstiger, den Zuschneidealgorithmus mehrmals aufzurufen. Das ist zeitaufwendig, aber wir werden bald weitere Werkzeuge kennenlernen, die die Automatisierung solcher Operationen erlaubt. In diesem Beispiel haben wir nur zwei Layer, so dass das jetzt nicht so wichtig ist.

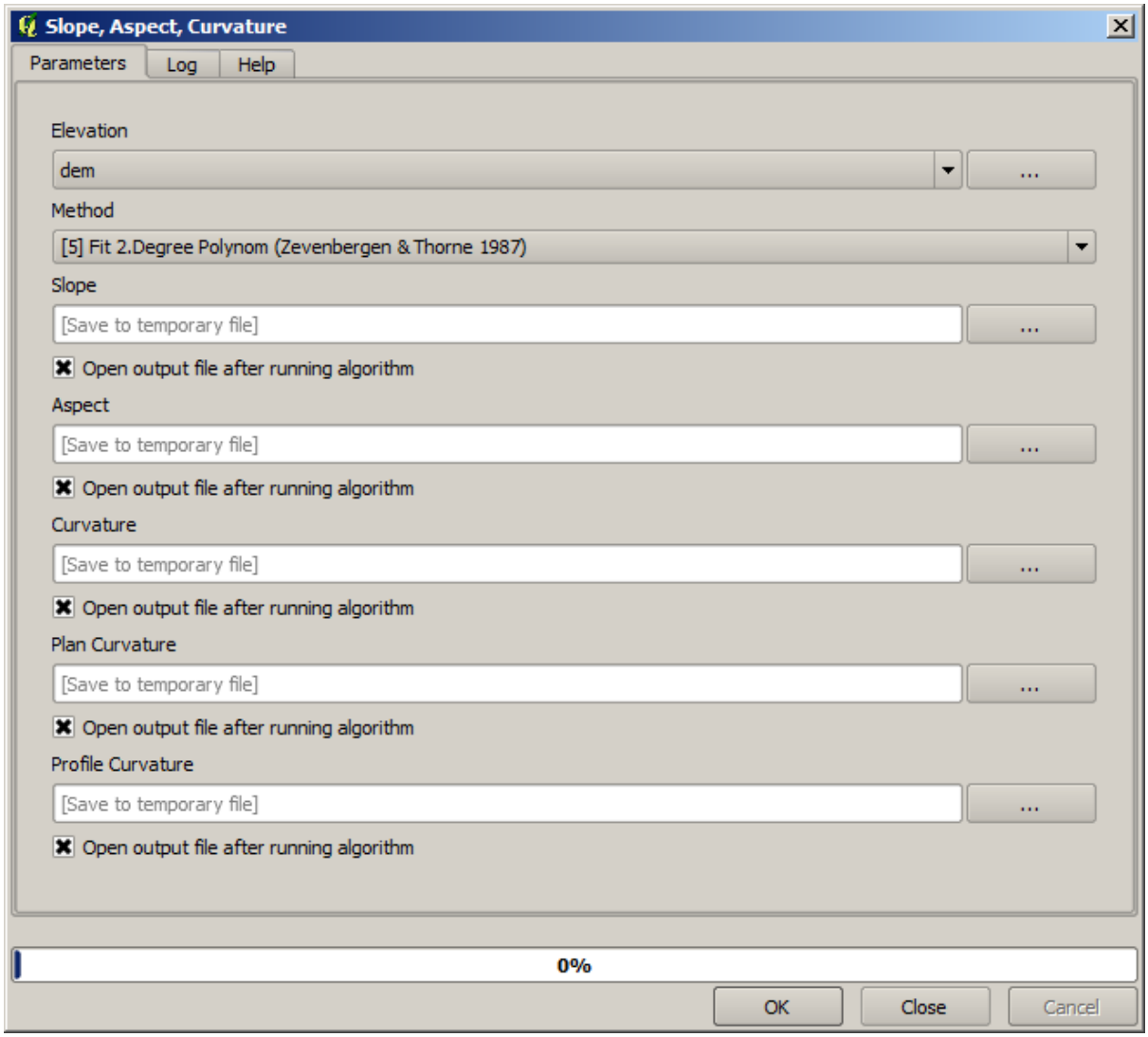
Wir haben jetzt unser finales DEM wie wir es wollten.



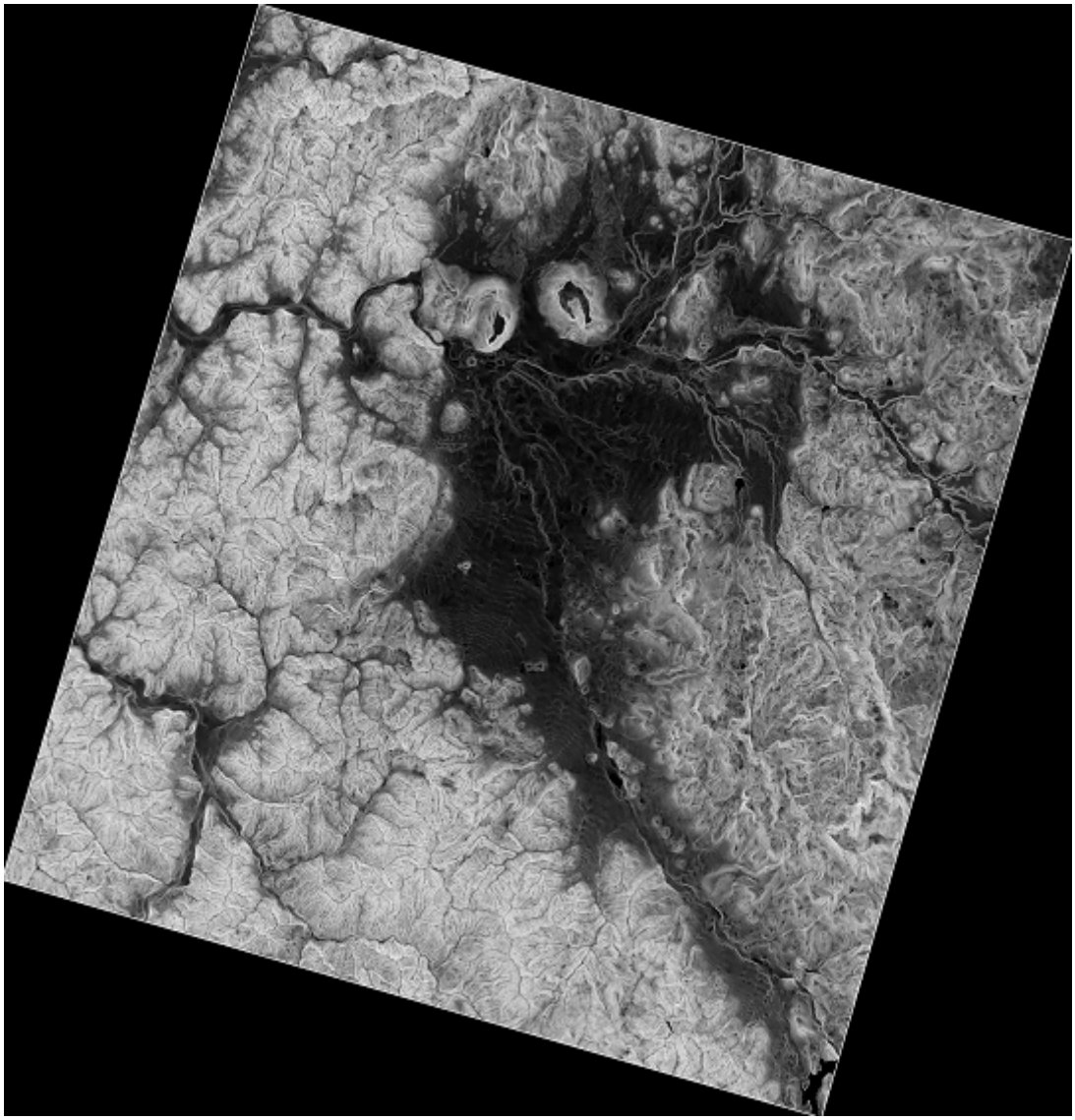
Jetzt ist es an der Zeit, den Layer mit dem Gefälle zu berechnen.

Ein Gefällelayer kann mit Hilfe des Algorithmus *Slope, Aspect, Curvature* berechnet werden. Allerdings ist unserer erstelltes DEM nicht als Eingabedatei geeignet, da nur die Höhenwerte in Meter hinterlegt sind. Die Zellengröße verwendet nicht Meter (der Layer liegt in einem KBS mit geographischen Koordinaten vor). Wir müssen ihn reprojizieren. Dazu können wir wieder den Algorithmus *Transformieren (reprojizieren)* verwenden. Wir reprojizieren in ein KBS mit der Einheit Meter (z.B. 3857), so dass wir dann das Gefälle korrekt mit SAGA oder GDAL bestimmen können.

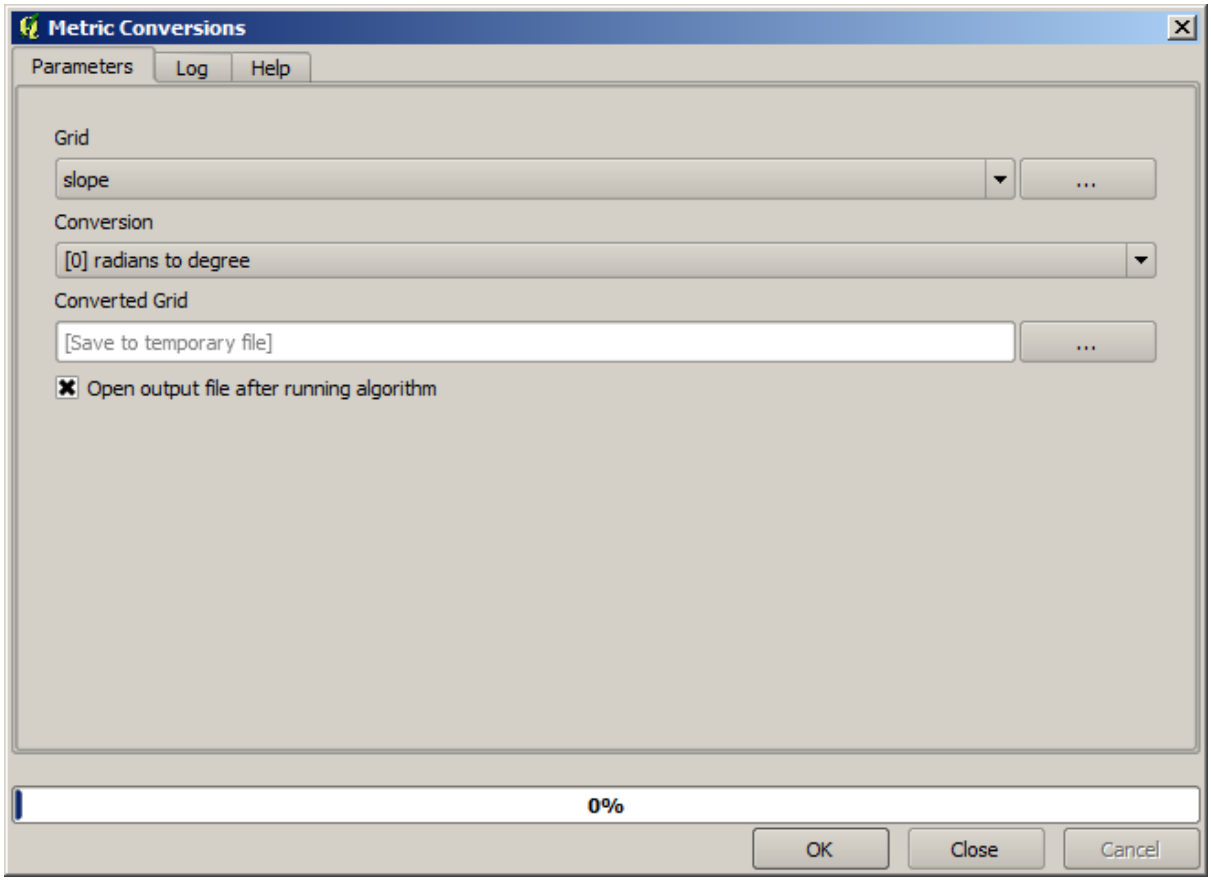
Mit dem neuen DEM kann nun die Hangneigung berechnet werden.



Hier ist der resultierende Layer mit der Hangneigung.



Die Hangneigung, die durch den Algorithmus *Slope, Aspect, Curvature* berechnet wird, kann in Grad oder Radiant ausgegeben werden. Die verbreitetere und auch praktikablere Einheit ist Grad. Falls Sie die Neigung in Radiant berechnet haben, kann man die Einheit mit Hilfe des Algorithmus *Metric conversions* umrechnen (für den Fall, dass man diesen Algortihmus nicht kennen würde, hätte man auch den Rasterrechner verwenden können, den wir schon benutzt haben).



Nach dem erneuten Reprojizieren des konvertierten Neigungslayers mit *Transformieren (reprojizieren)* zurück zum Ausgangs KBS erhalten wir den Ergebnislayer, den wir uns vorgestellt haben.

Warnung: todo: Bild einfügen

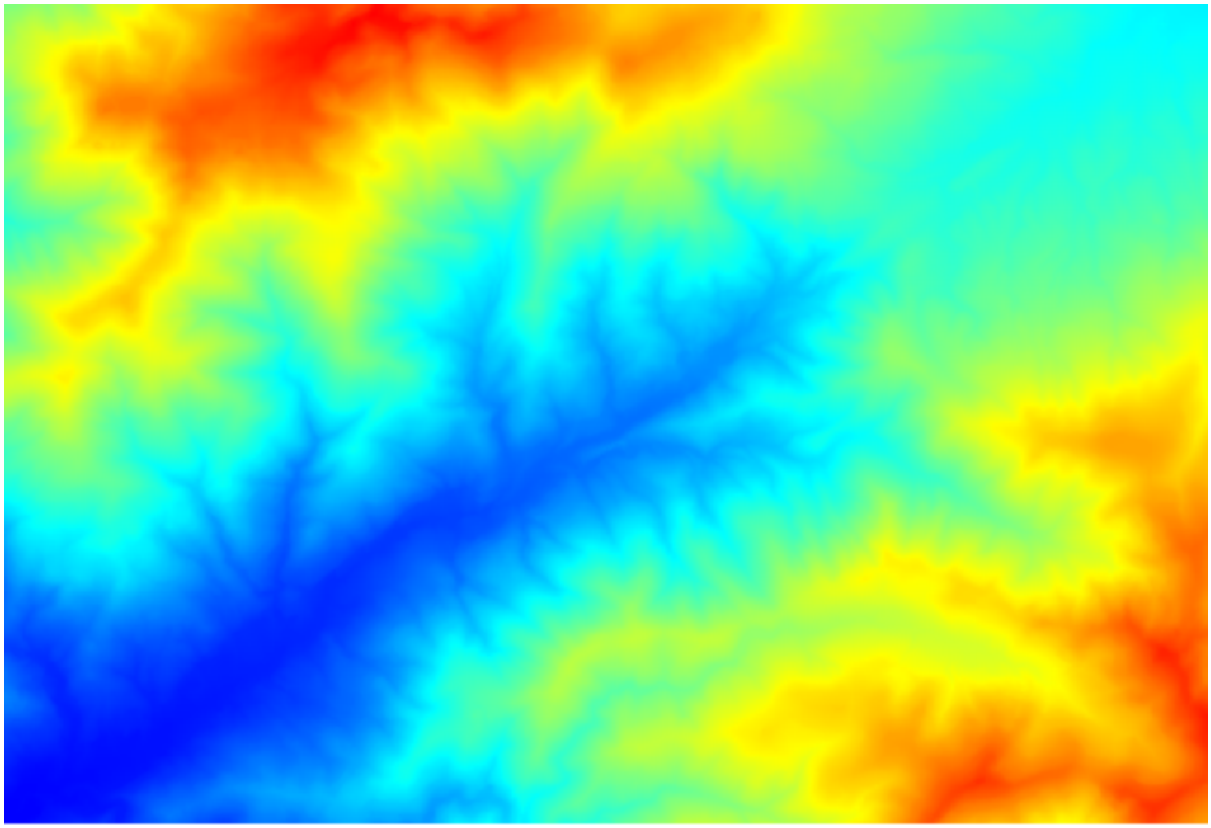
Durch den Prozess der Reprojektion, liegen unter Umständen Teile des Ergebnislayers außerhalb des Begrenzungsrahmens, den wir anfangs berechnet hatten. Das kann durch ein erneutes Zuschneiden des Rasters analog zum Vorgehen beim DEM gelöst werden.

17.16 Hydrologische Analyse

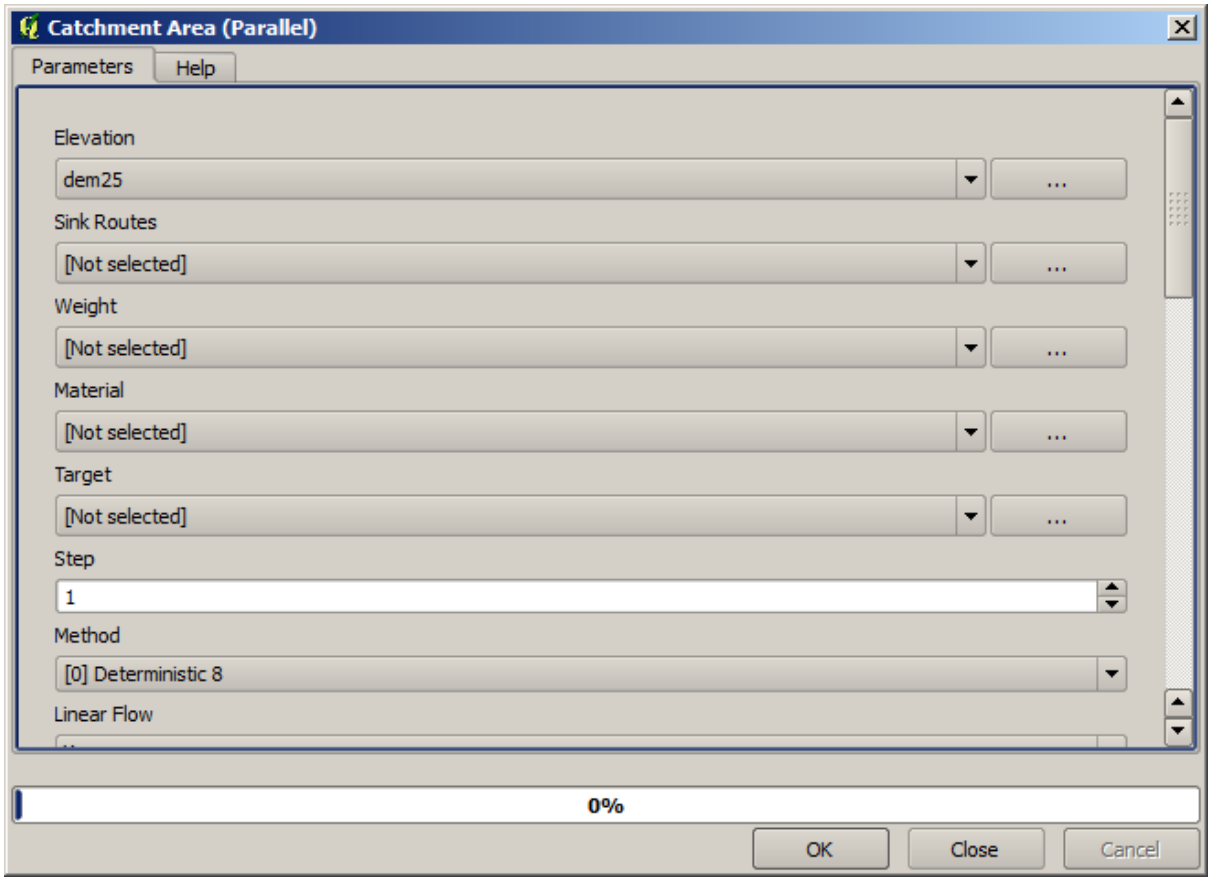
Bemerkung: In dieser Lektion werden wir einige hydrologische Untersuchungen durchführen. Diese werden in einigen der anderen Übungen gebraucht, da diese ein sehr gutes Beispiel für einen analytischen Workflow darstellt und diese für einige erweiterte Funktionen genutzt werden.

Objectives: Starting with a DEM, we are going to extract a channel network, delineate watersheds and calculate some statistics.

1. Zu Beginn müssen die Übungsdaten in das Projekt geladen werden, welche lediglich ein DGM bzw. DEM ist.

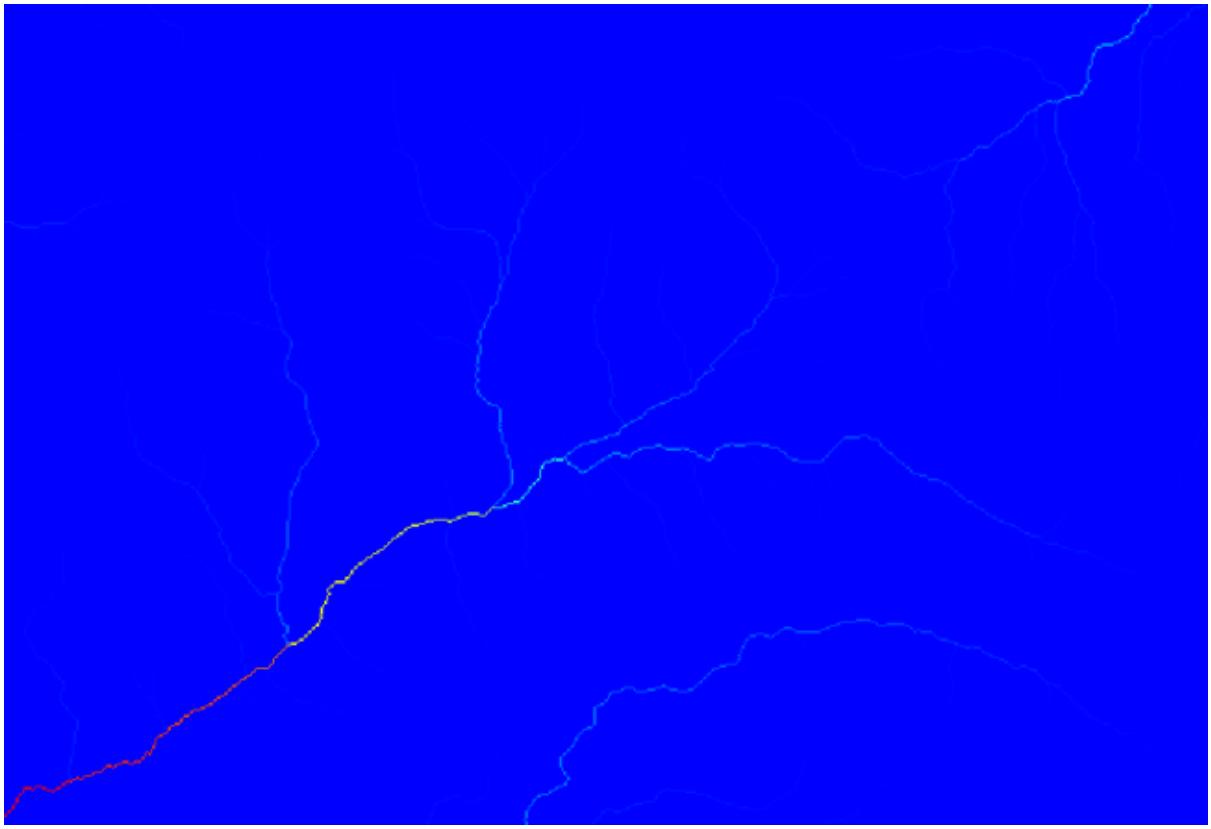


2. The first module to execute is *Catchment area* (in some SAGA versions it is called *Flow accumulation (Top Down)*). You can use any of the others named *Catchment area*. They have different algorithms underneath, but the results are basically the same.
3. Select the DEM in the *Elevation* field, and leave the default values for the rest of the parameters.

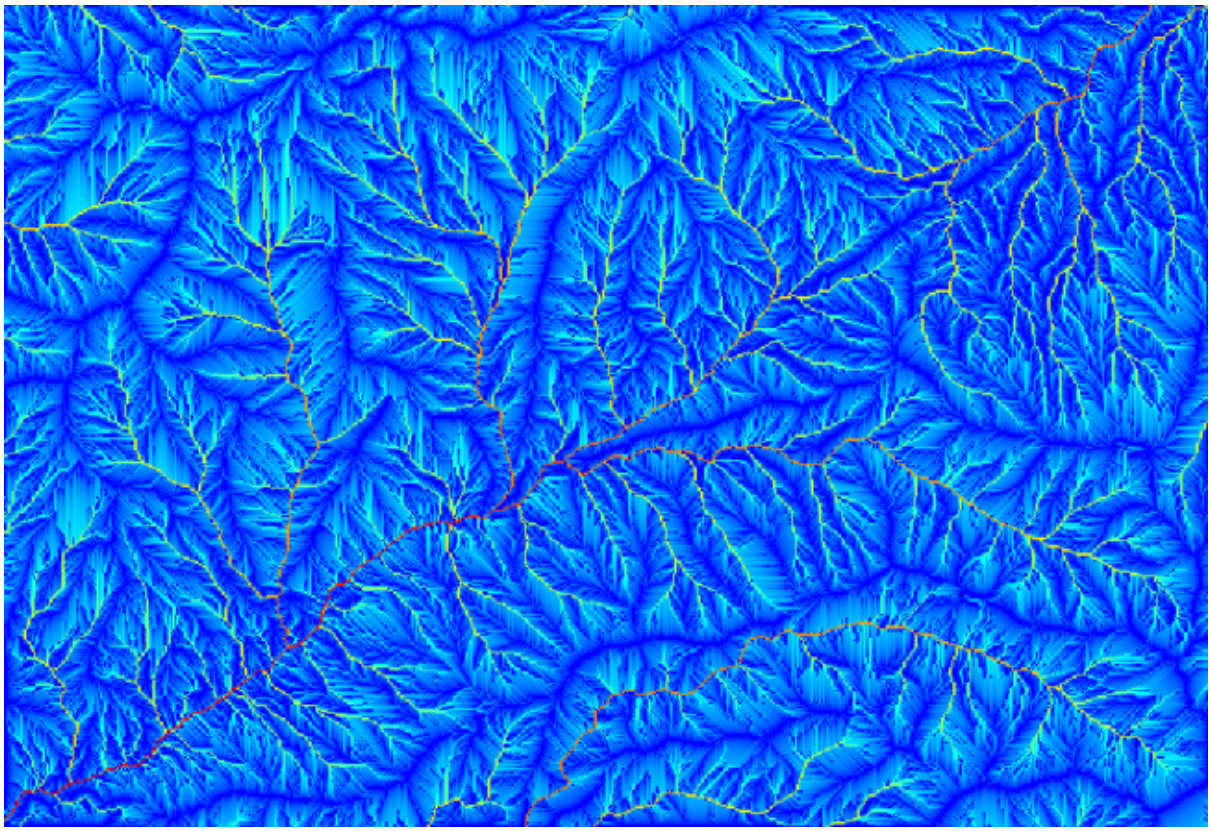


Some algorithms calculate many layers, but the *Catchment Area* layer is the only one we will be using. You can get rid of the other ones if you want.

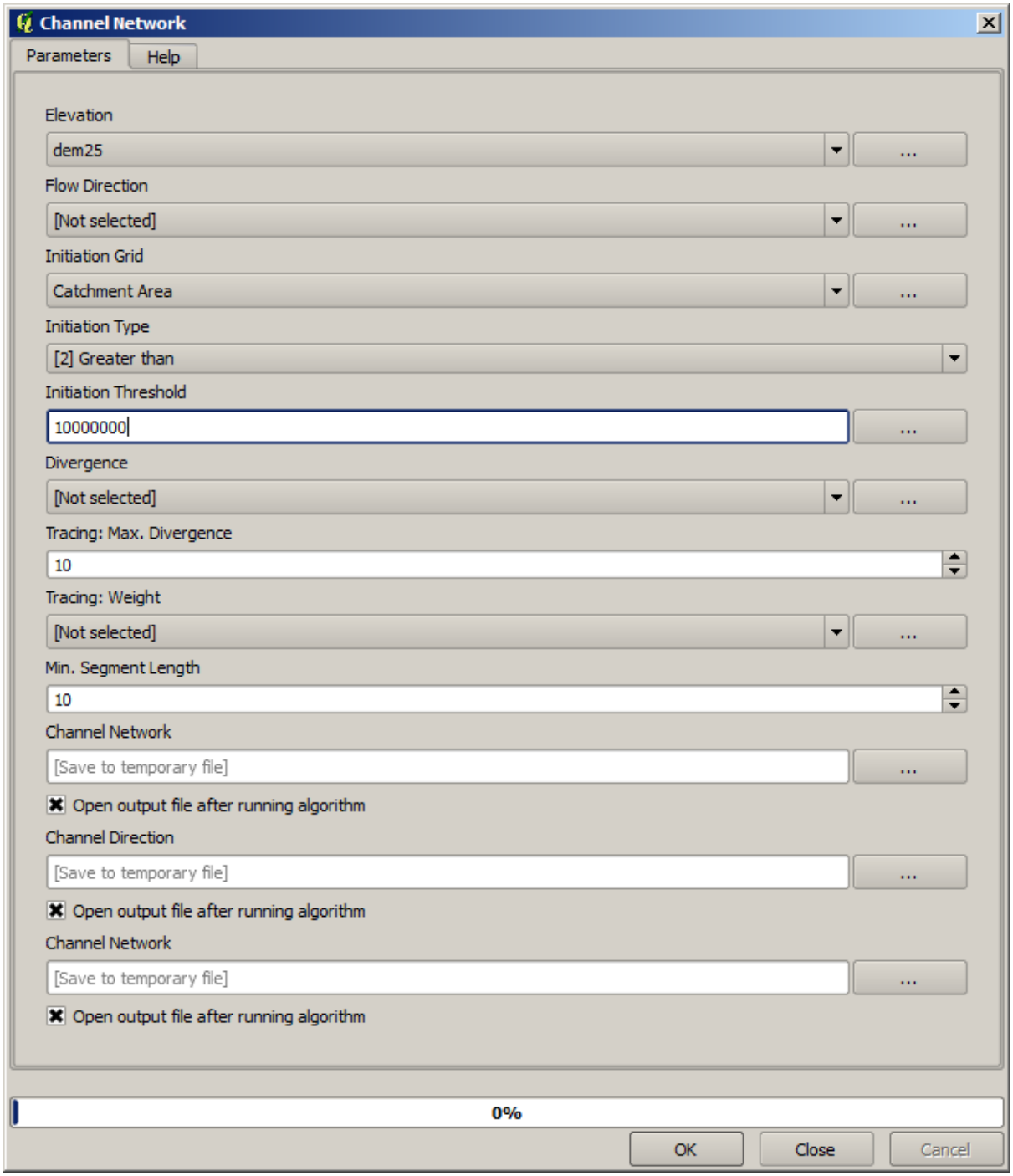
Der erstellte Layer ist nicht sonderlich Informativ.



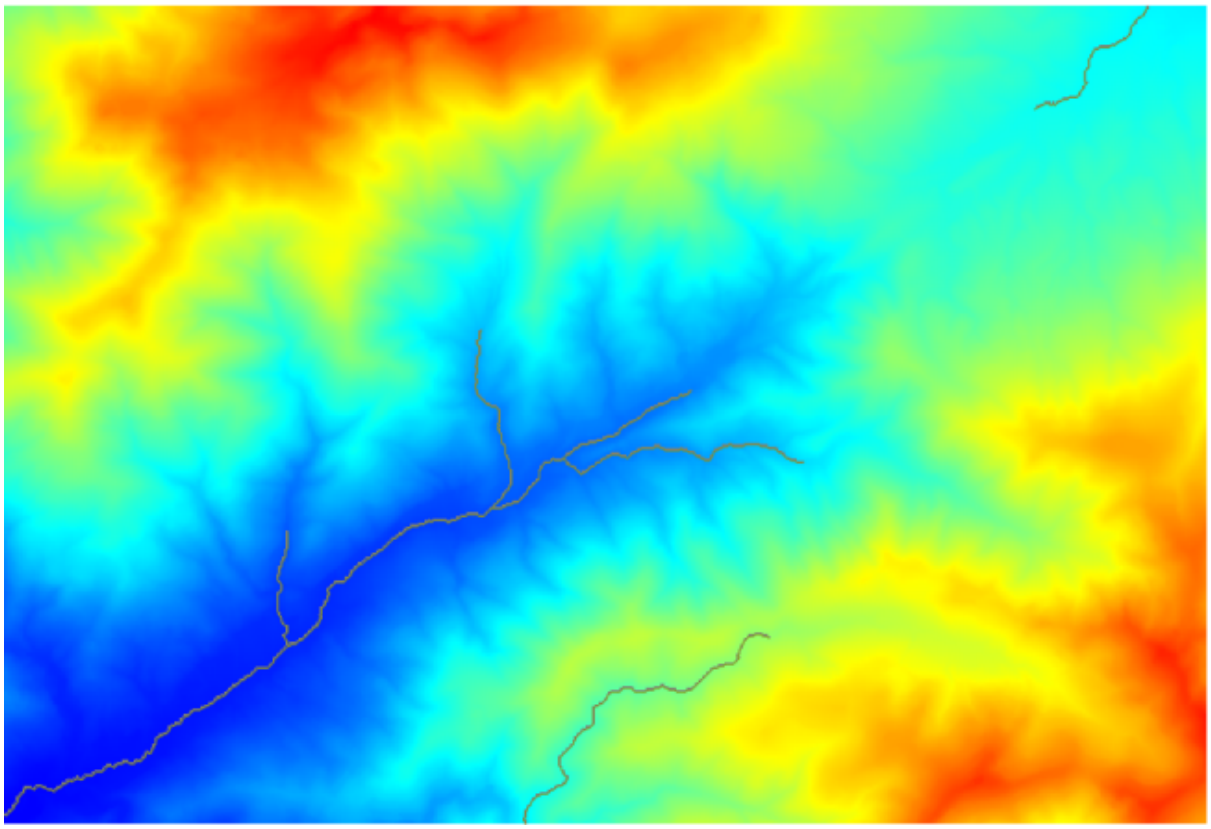
To know why, you can have a look at the histogram and you will see that values are not evenly distributed (there are a few cells with very high value, those corresponding to the channel network). Use the *Raster calculator* algorithm to calculate the logarithm of the catchment value area and you will 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*

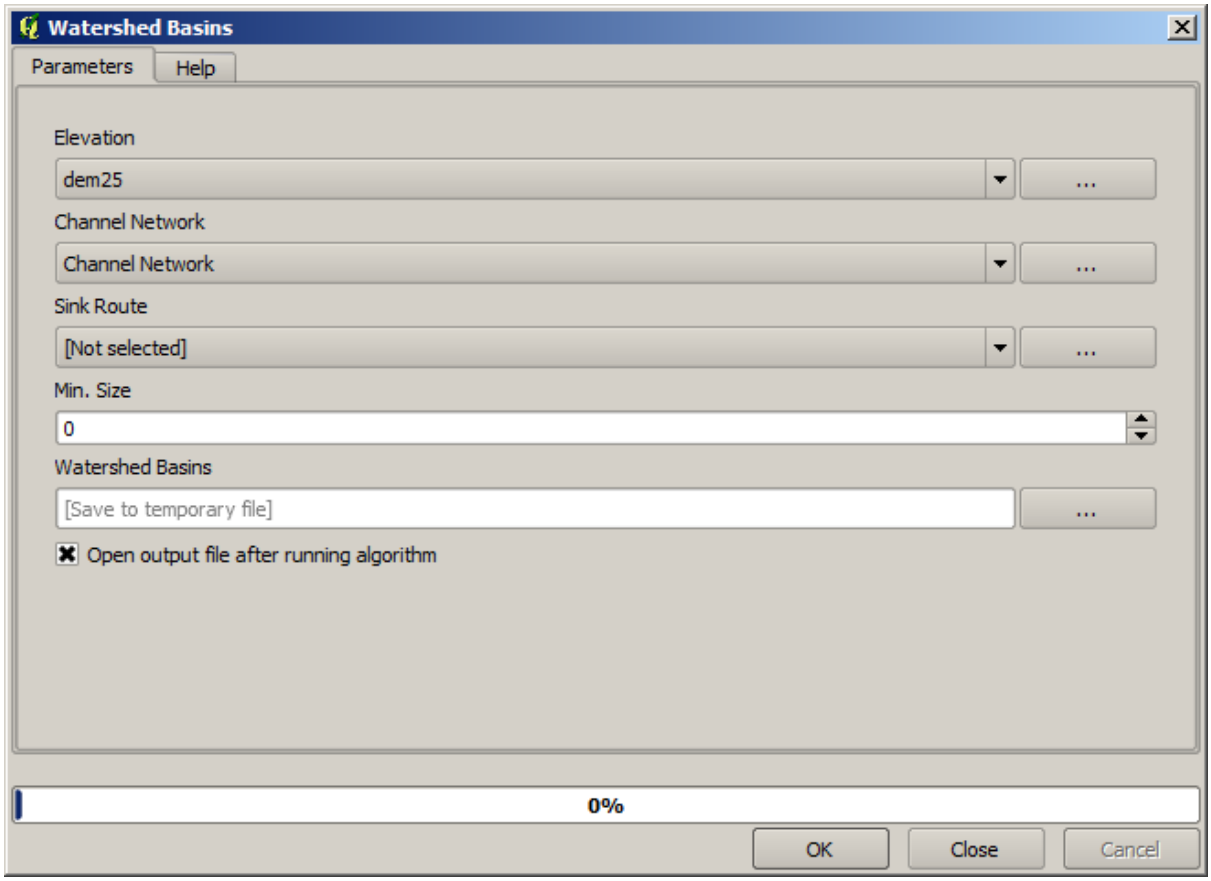


If you increase the *Initiation threshold* value, you will get a more sparse channel network. If you decrease it, you will get a denser one. With the proposed value, this is what you get.

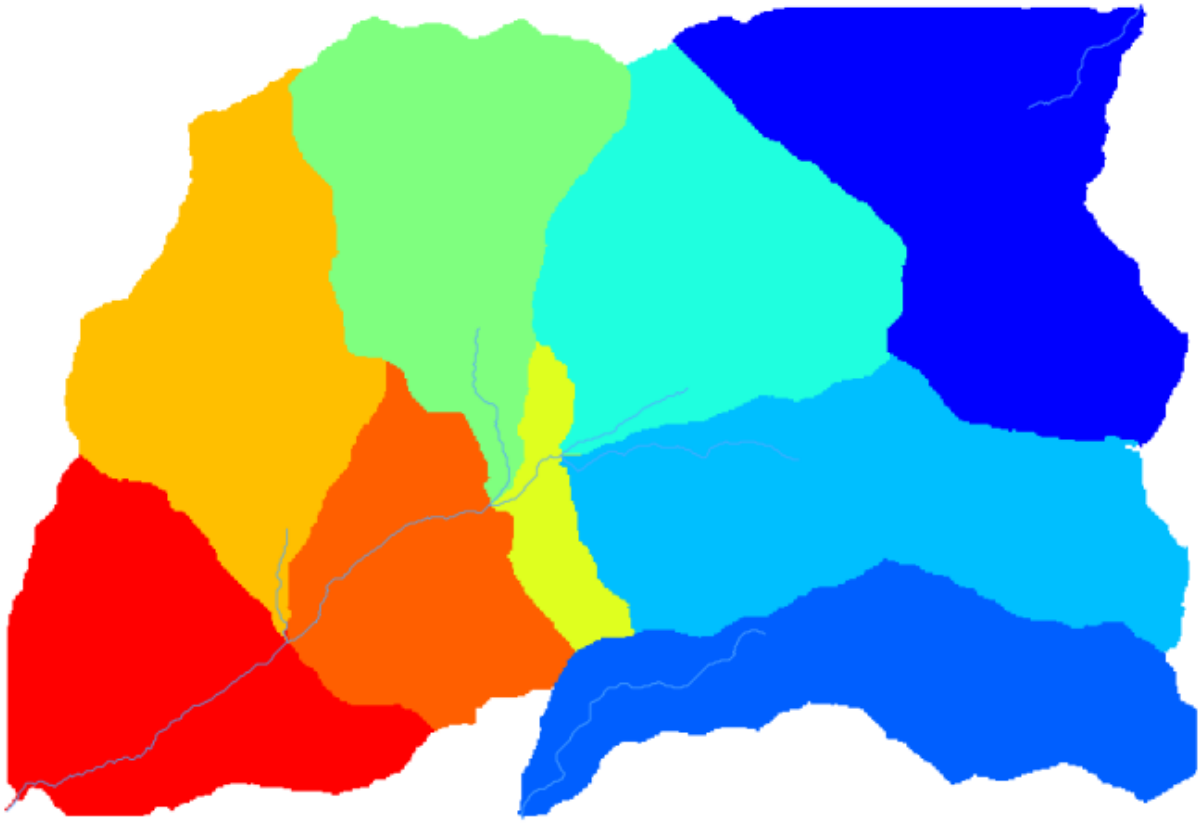


The image above shows just the resulting vector layer and the DEM, but there should be also a raster layer with the same channel network. That raster will be, in fact, the layer we will be using.

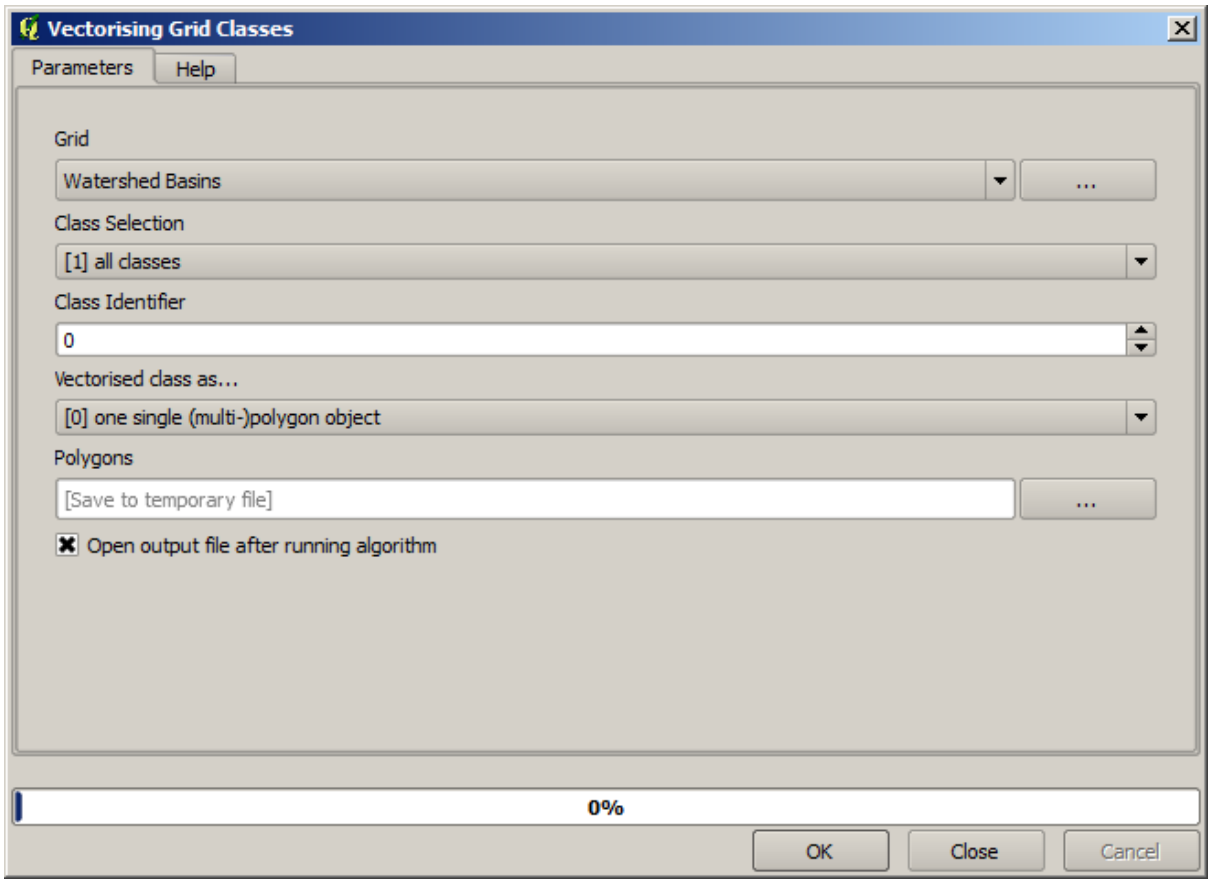
5. Now, we will use the *Watersheds basins* algorithm to delineate the subbasins corresponding to that channel network, using as outlet points all the junctions in it. Here is how you have to set the corresponding parameters dialog.

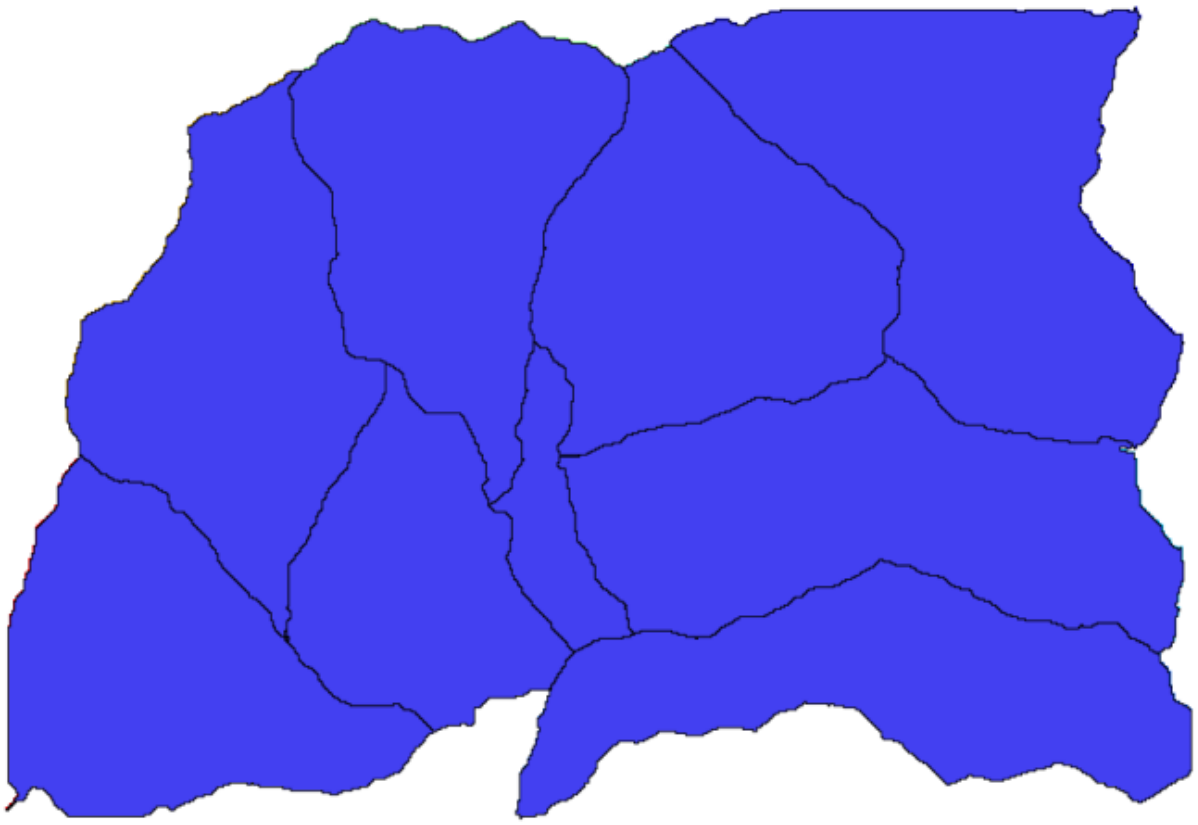


Somit entsteht folgendes.



6. This is a raster result. You can vectorise it using the *Vectorising grid classes* algorithm.

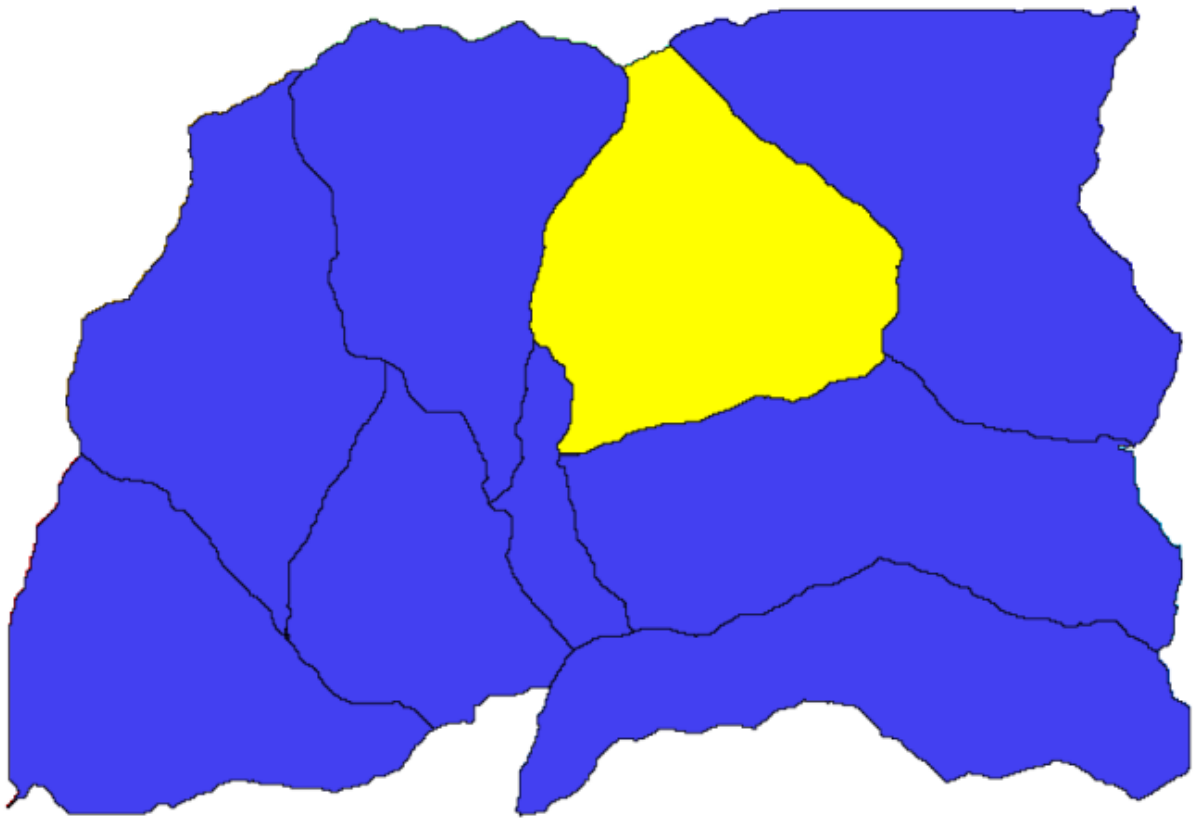




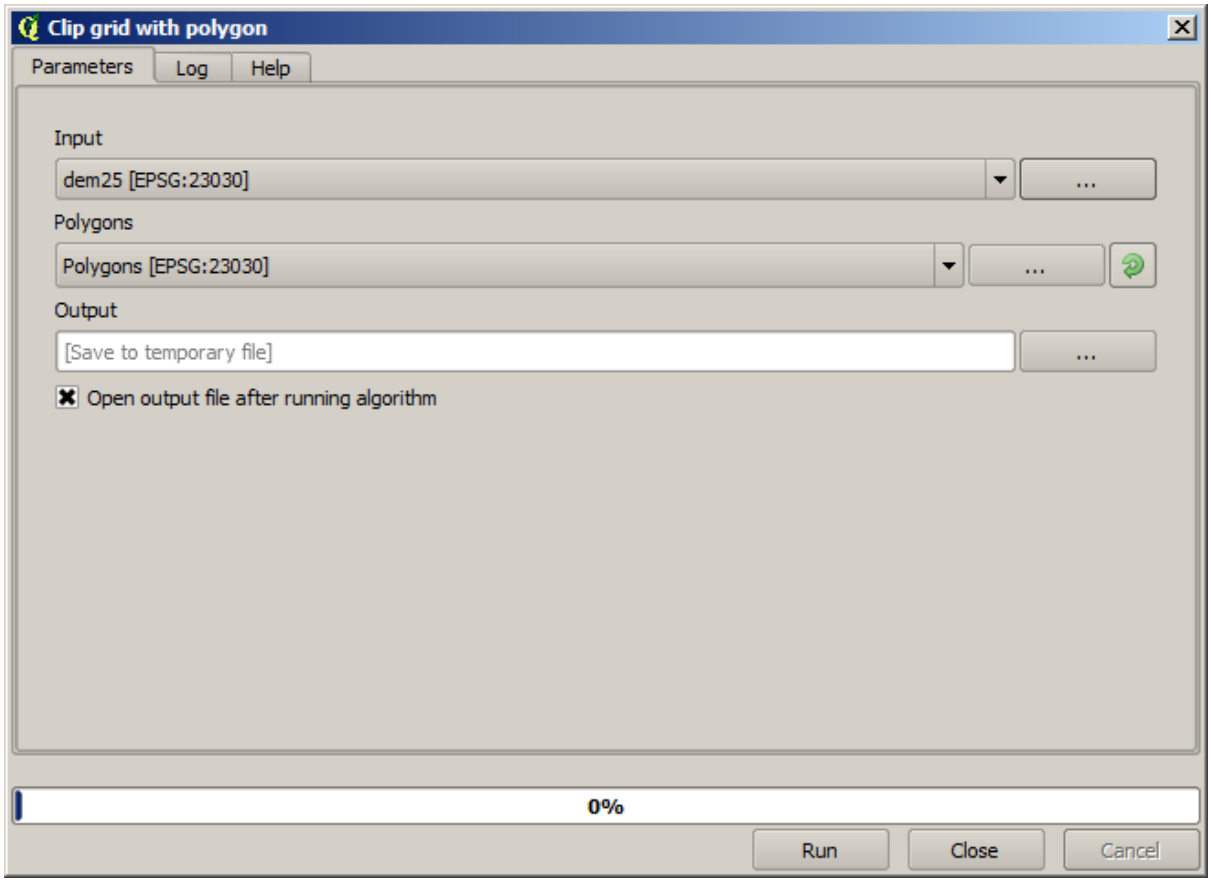
Nun versuchen wir statistische Höhenwerte in einem unsere Teileinzugsgebiete zu berechnen. Der Gedanke ist einen Layer zu haben, welche stellvertretend für die Höhen im Teileinzugsgebiet steht um dieses an das Modul zu übergeben, welches diese Statistiken berechnet.

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

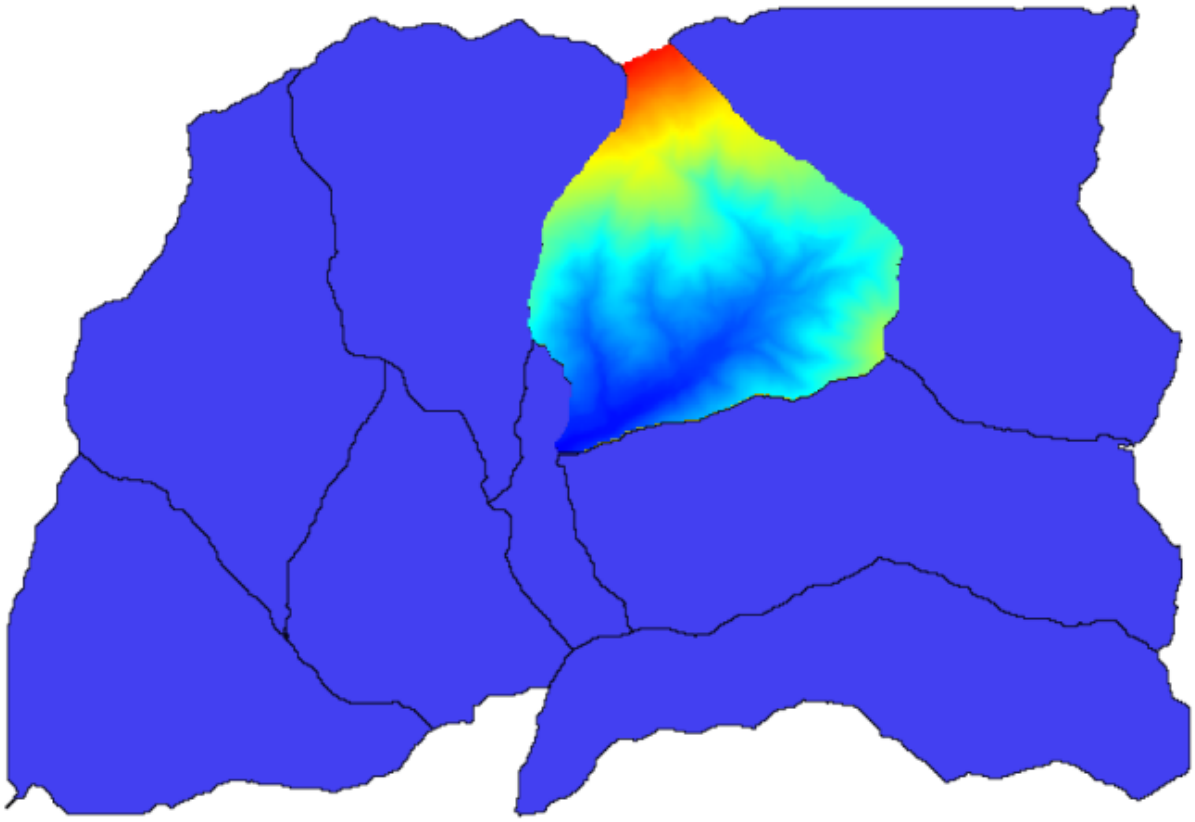


2. Call the clipping algorithm with the following parameters:

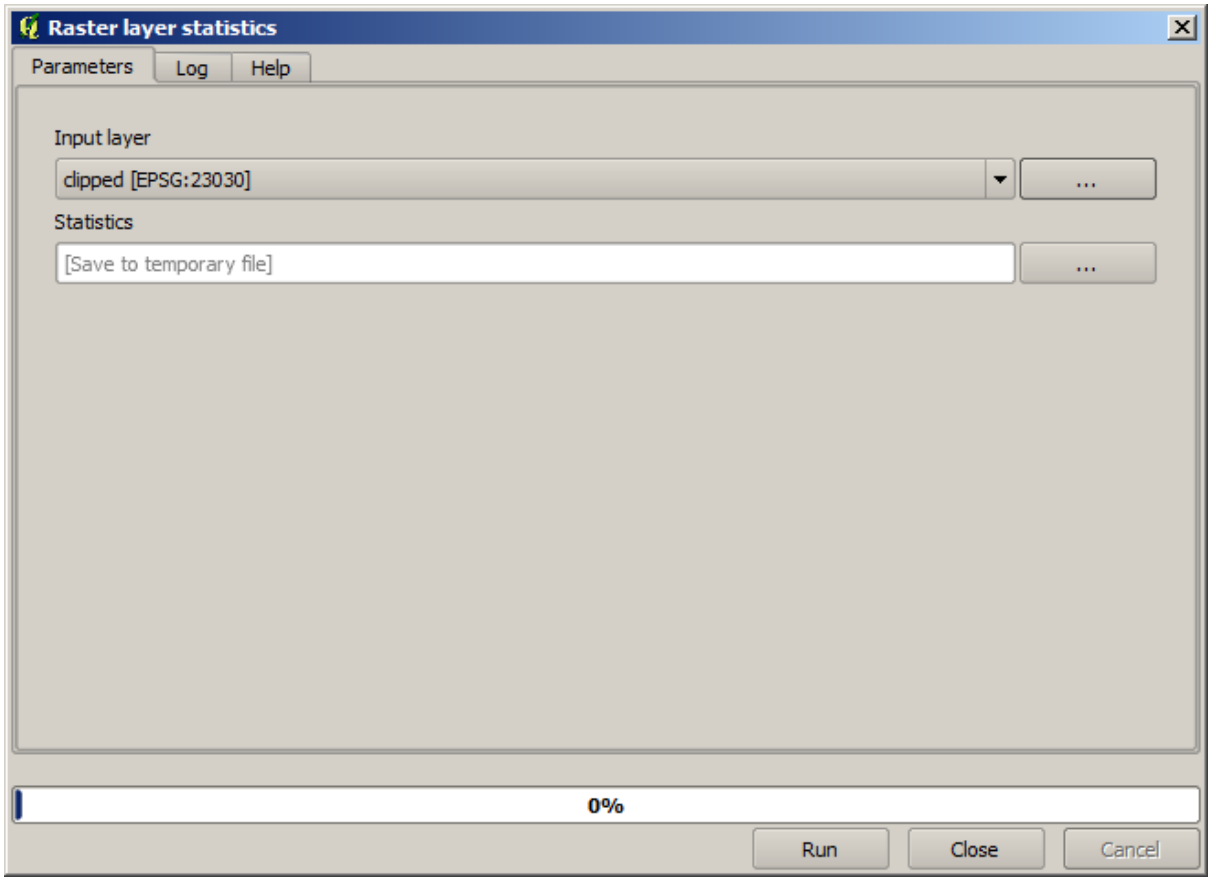


The element selected in the input field is, of course, the DEM we want to clip.

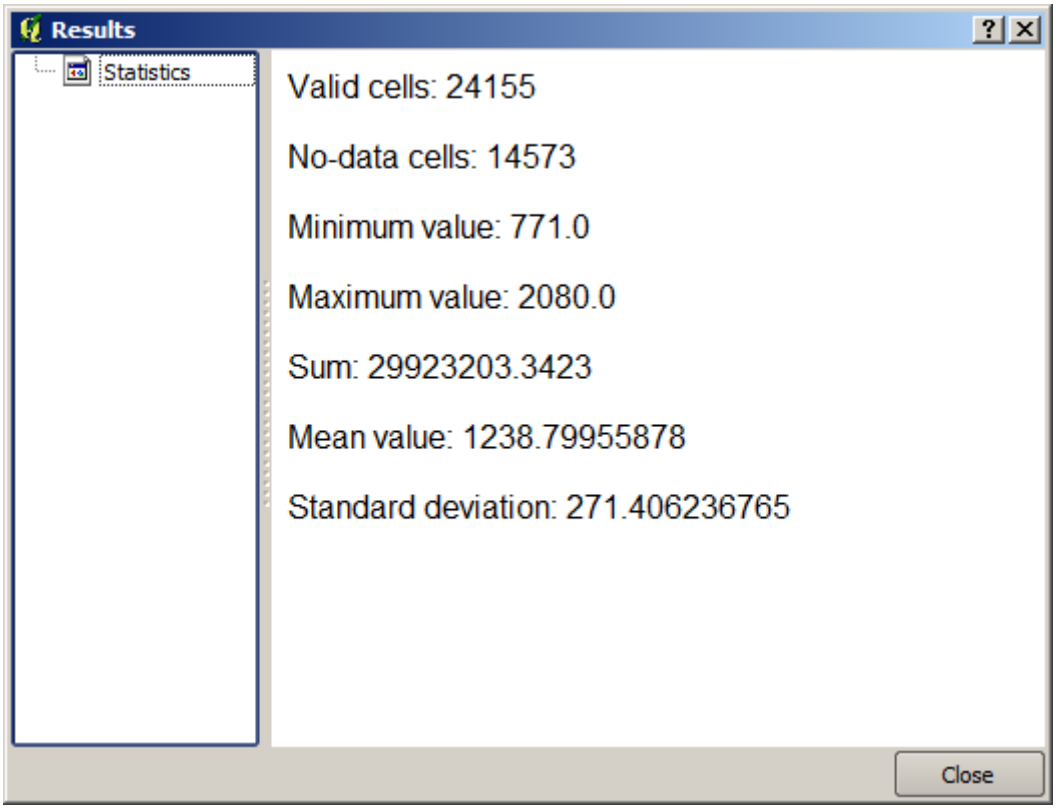
Dies sollte dann in etwa so aussehen.



2. This layer is ready to be used in the *Raster layer statistics* algorithm.



Die resultierende Statistik ist folgende.



Wir werden die Einzugsgebietsberechnung und die statistischen Werte in einer anderen Übung nochmals verwenden,

um herauszufinden wie andere Elemente uns helfen, automatisiert dies durchzuführen und effektiver zu arbeiten.

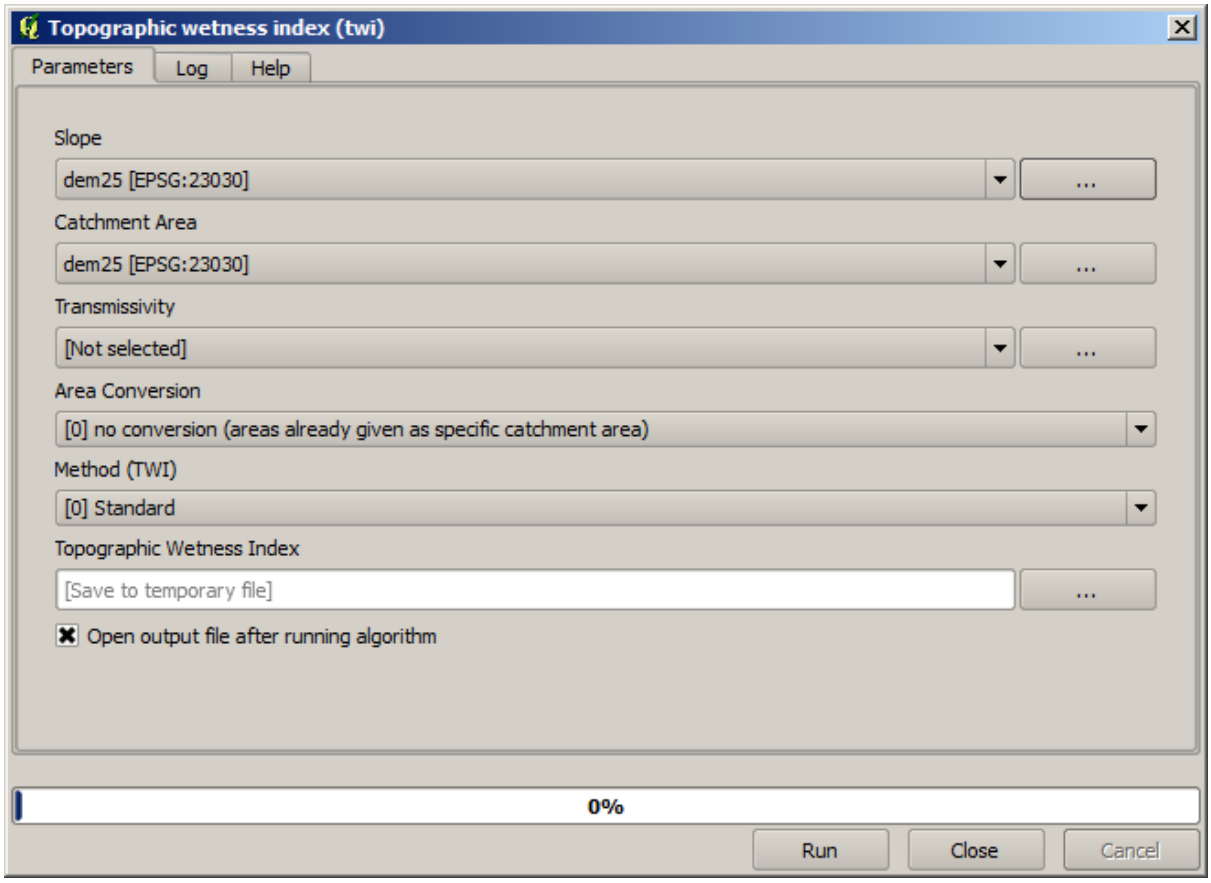
17.17 Starting with the model designer

Bemerkung: In this lesson we will use the model designer, a powerful component that we can use to define a workflow and run a chain of algorithms.

Bei der Arbeit mit der Verarbeitungsumgebung führt man in der Regel mehr als einen Algorithmus aus. Das Erreichen des Ergebnisses erfordert es meistens, mehrere Algorithmen zu starten, wobei die Ausgabe eines Algorithmus als Eingabe für einen anderen Algorithmus verwendet wird.

Using the model designer, 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)*.

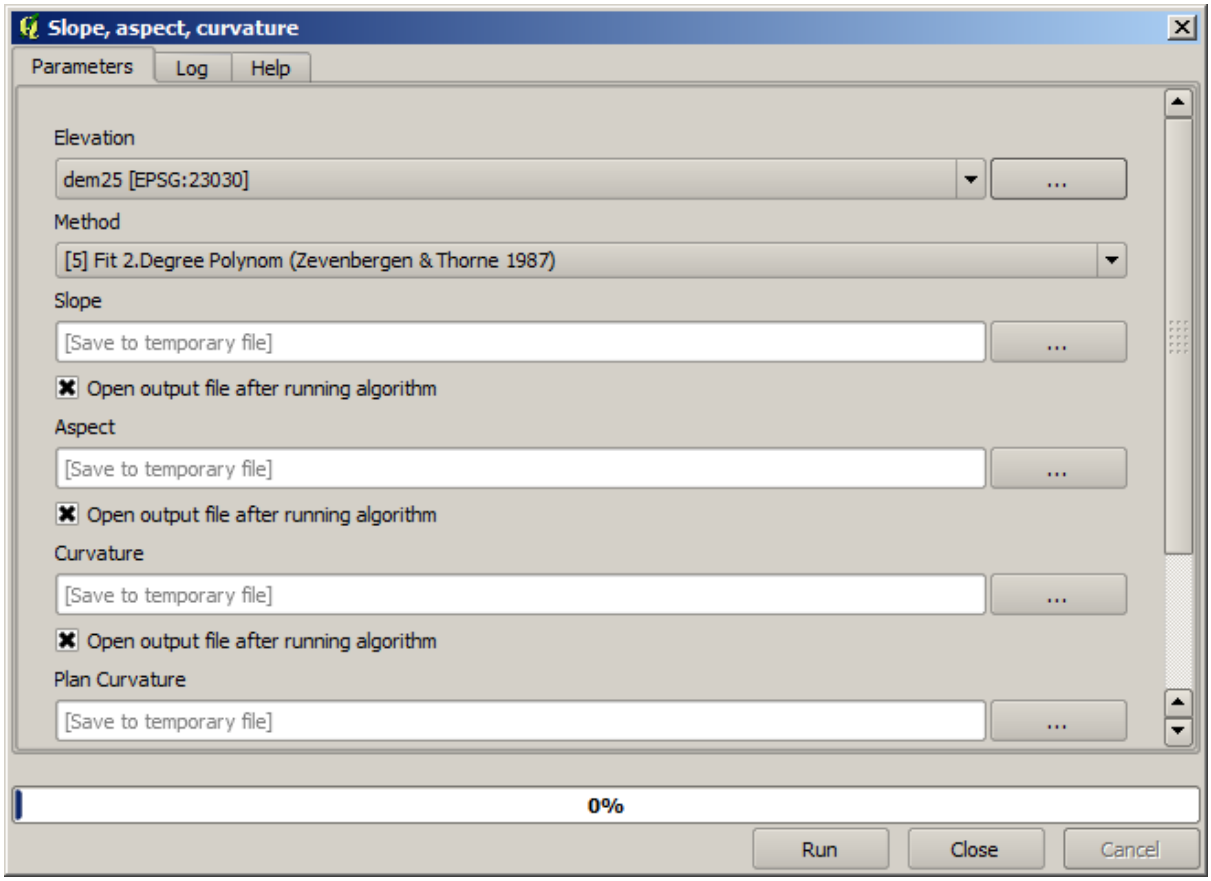


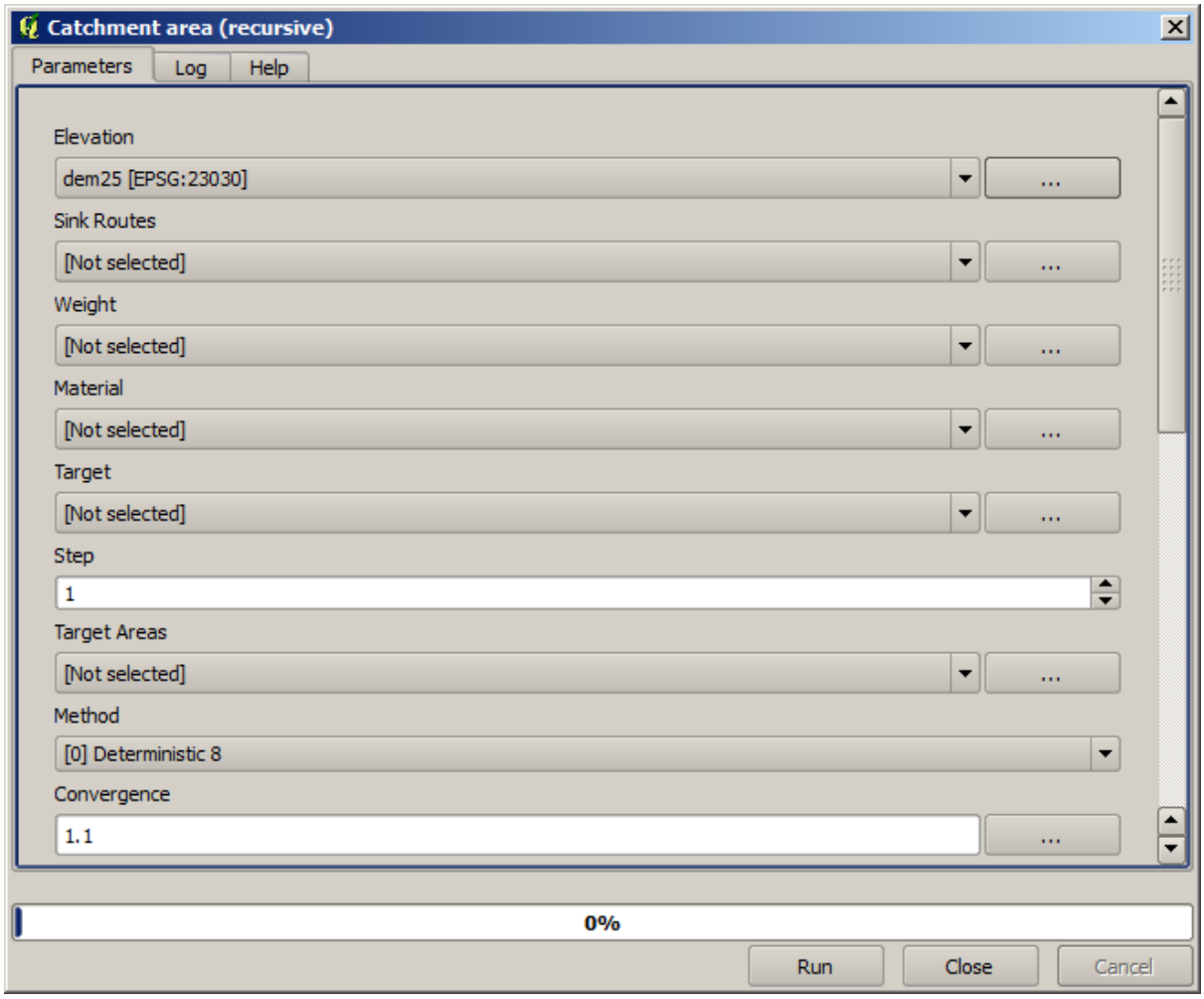
Wir sehen zwei Eingaben, die ausgefüllt werden müssen: *Slope* und *Catchment area*. Es gibt noch weitere optionale Eingabefelder, die wir außer Acht lassen können.

Die Daten für diese Lektion bestehen nur aus dem DEM, wir haben noch keine der benötigten Eingangsdaten. Wir wissen aber bereits, mit welchen Algorithmen man Hangneigung und Einzugsgebiete berechnen kann. Wir können diese Layer also zuerst berechnen und sie dann als Eingangsdaten für den TWI Algorithmus verwenden.

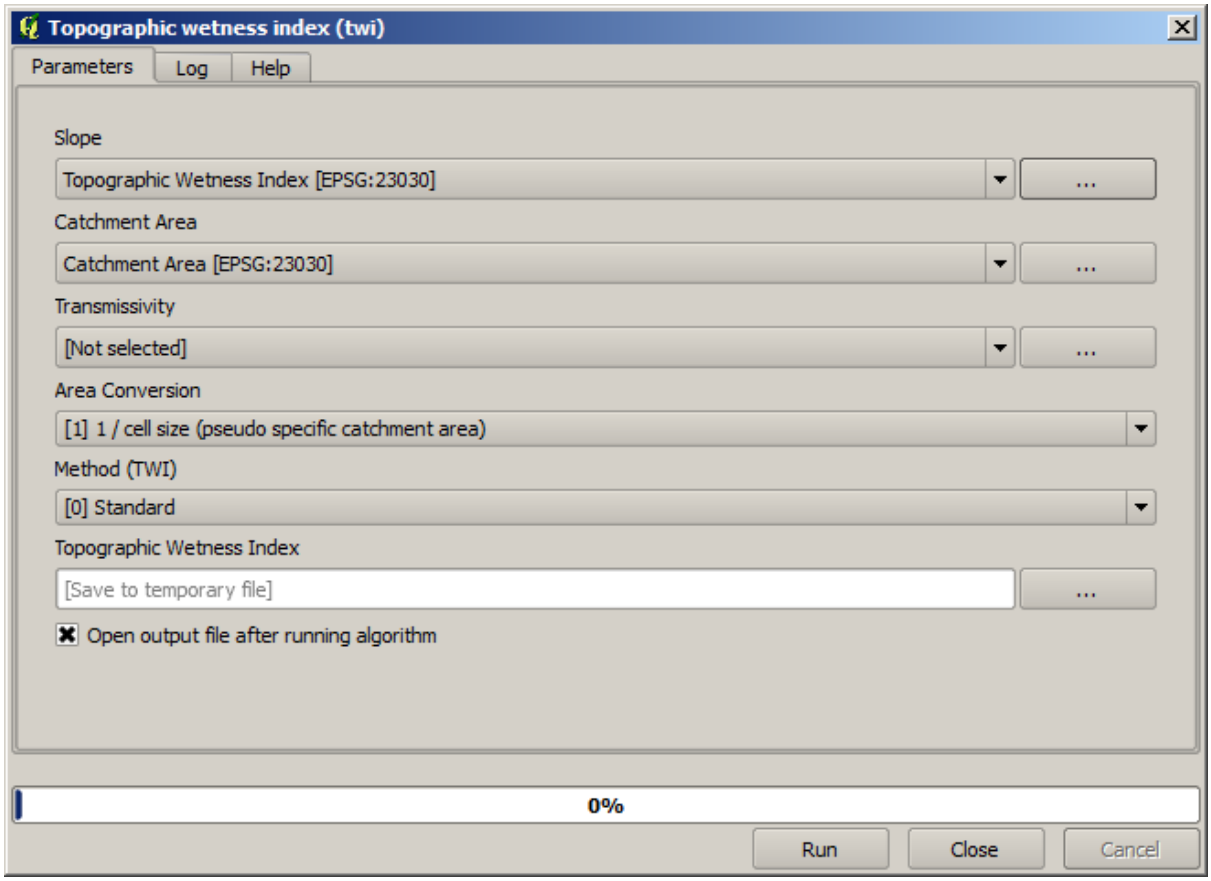
Es folgen die Paramter, die wir für die Berechnung der 2 Layer des Zwischenstandes benutzen.

Bemerkung: Die Hangneigung muss in Radiant und nicht in Grad berechnet werden.

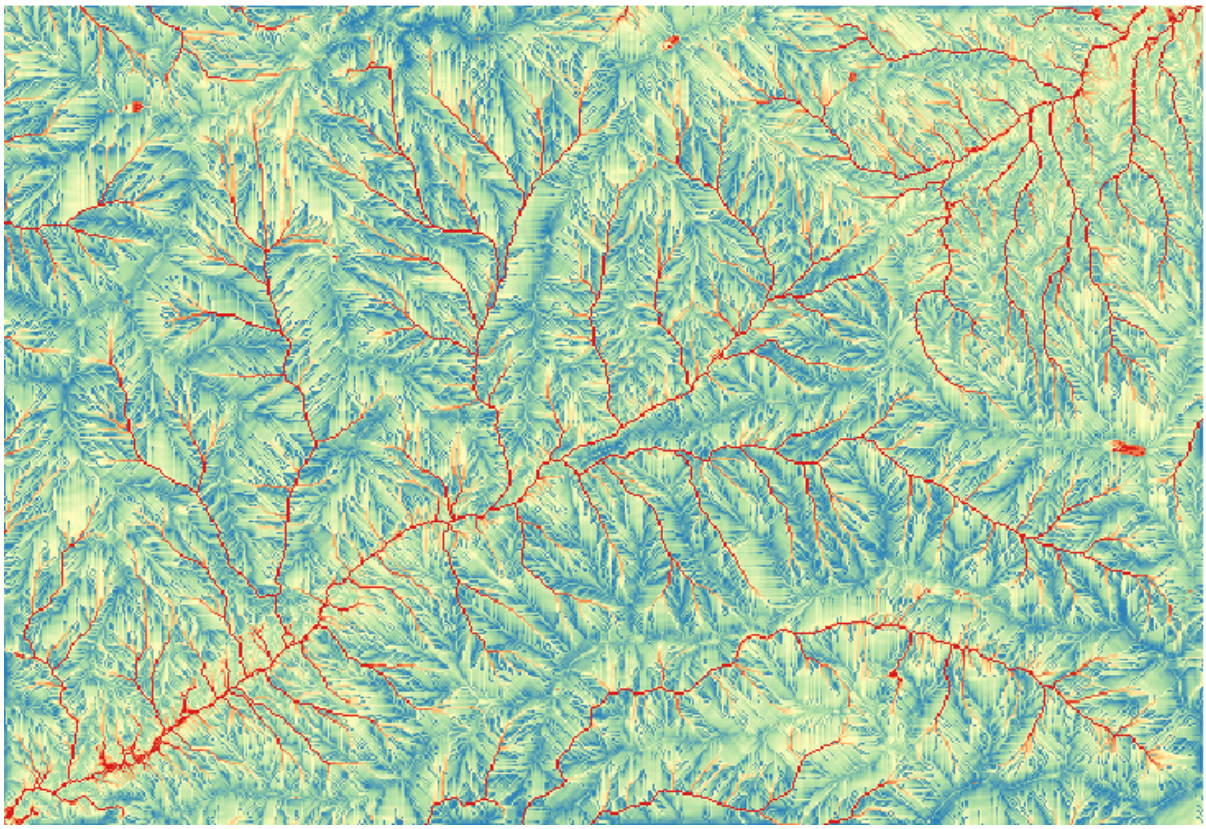

Slope, aspect, curvature
Parameters
Log
Help
Elevation
dem25 [EPSG:23030]
...
Method
[5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
Slope
[Save to temporary file]
...
Open output file after running algorithm
Aspect
[Save to temporary file]
...
Open output file after running algorithm
Curvature
[Save to temporary file]
...
Open output file after running algorithm
Plan Curvature
[Save to temporary file]
...
0%
Run
Close
Cancel



Und so setzen wir die Paramter im Dialog des TWI Algorithmus.

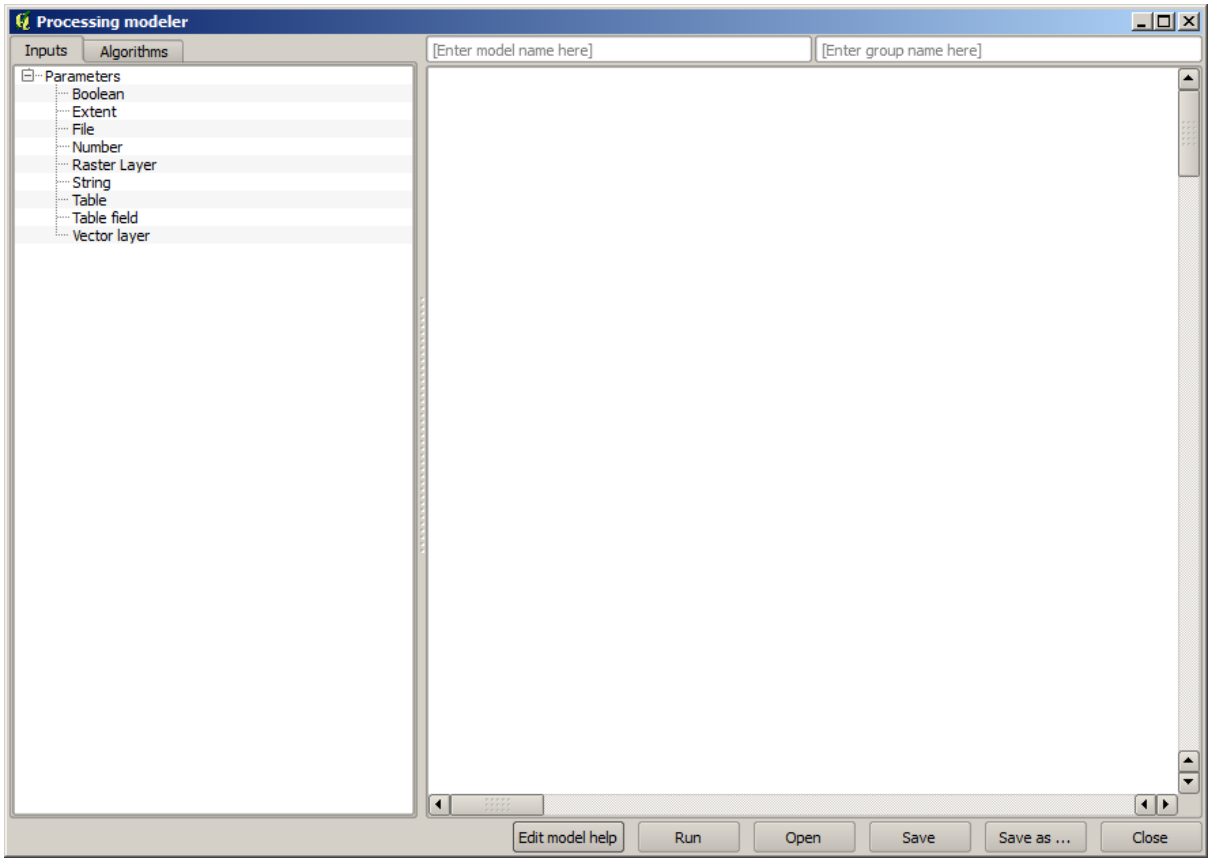


This is the result that you will obtain (the default singleband pseudocolor inverted palette has been used for rendering). You can use the `twi.qml` style provided.



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 model designer comes in.

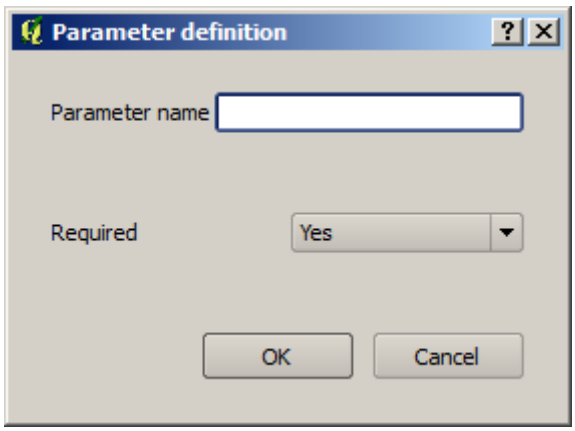
1. Öffnen Sie die Modellierung über den Eintrag im Verarbeitungsmenü.



   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. Wir beginnen mit den Eingabedaten. In diesem Fall muss nicht viel hinzugefügt werden. Wir brauchen als Eingabe nur den Rasterlayer mit dem DEM.

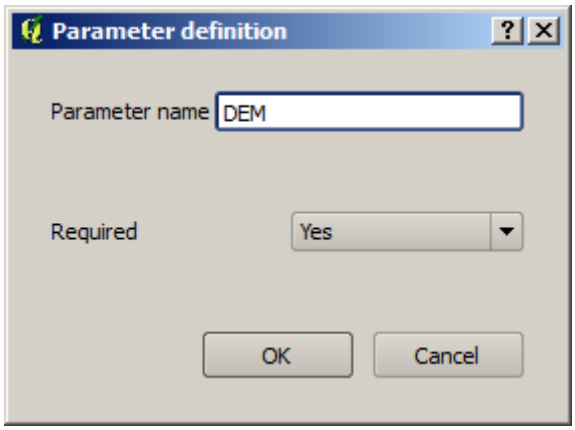
3. Double click on the *Raster layer* input and you will see the following dialog.



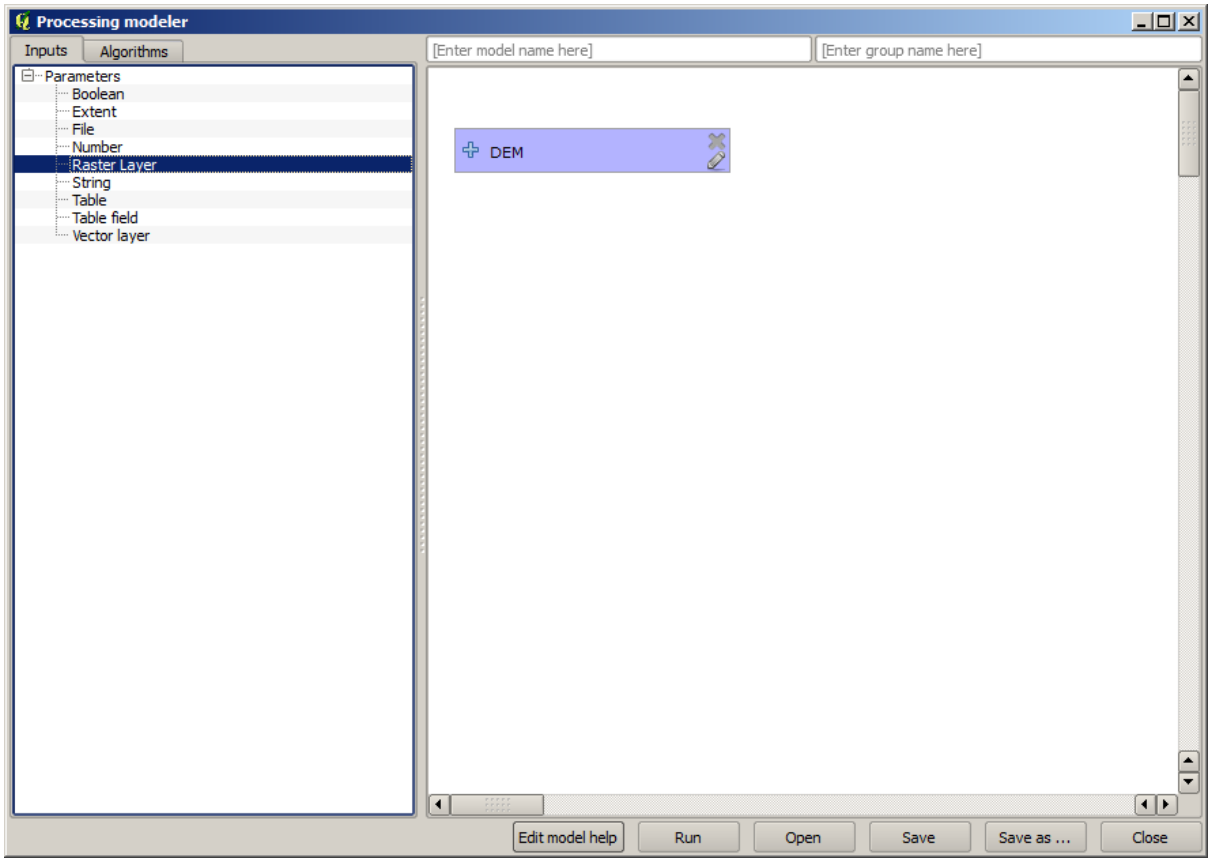
4. Here we will have to define the input we want:

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.

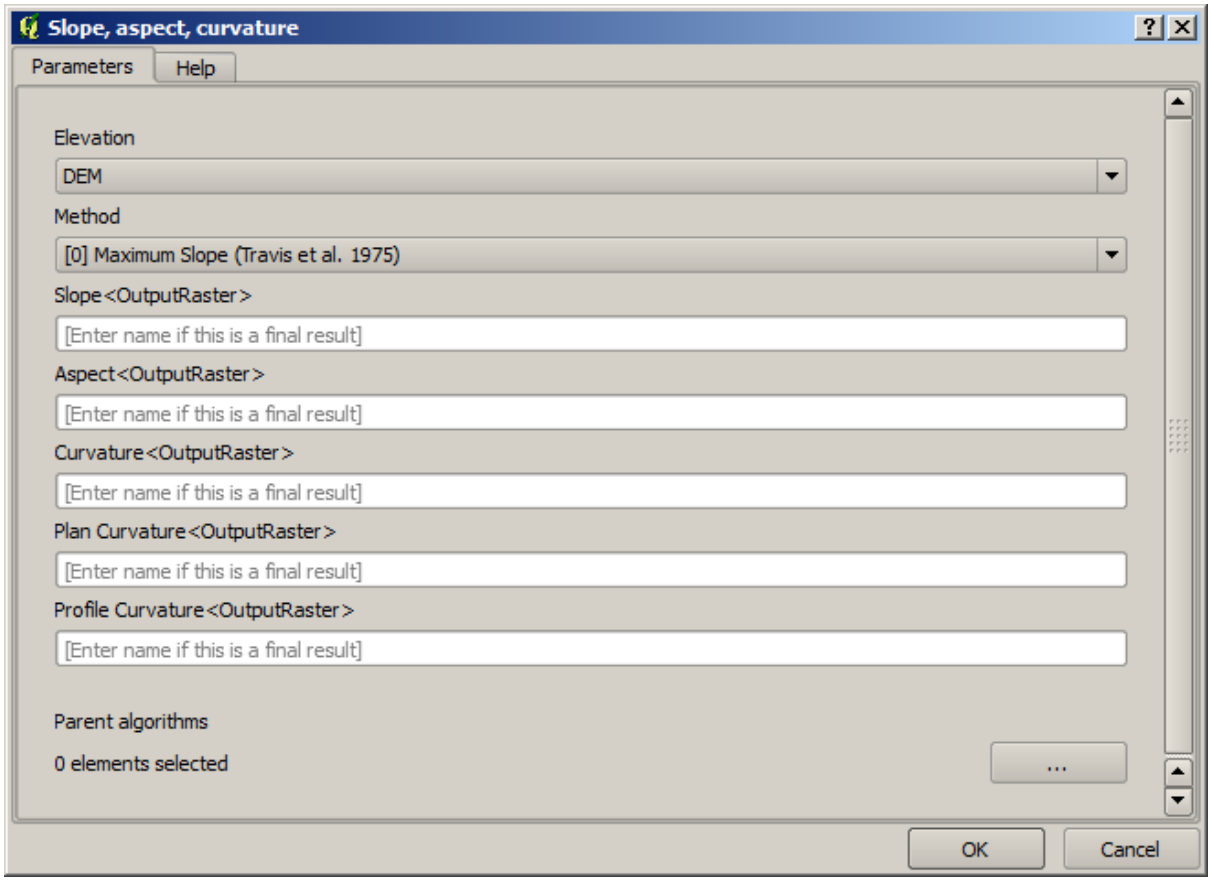
Der Dialog sollte wie folgt aussehen.



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.

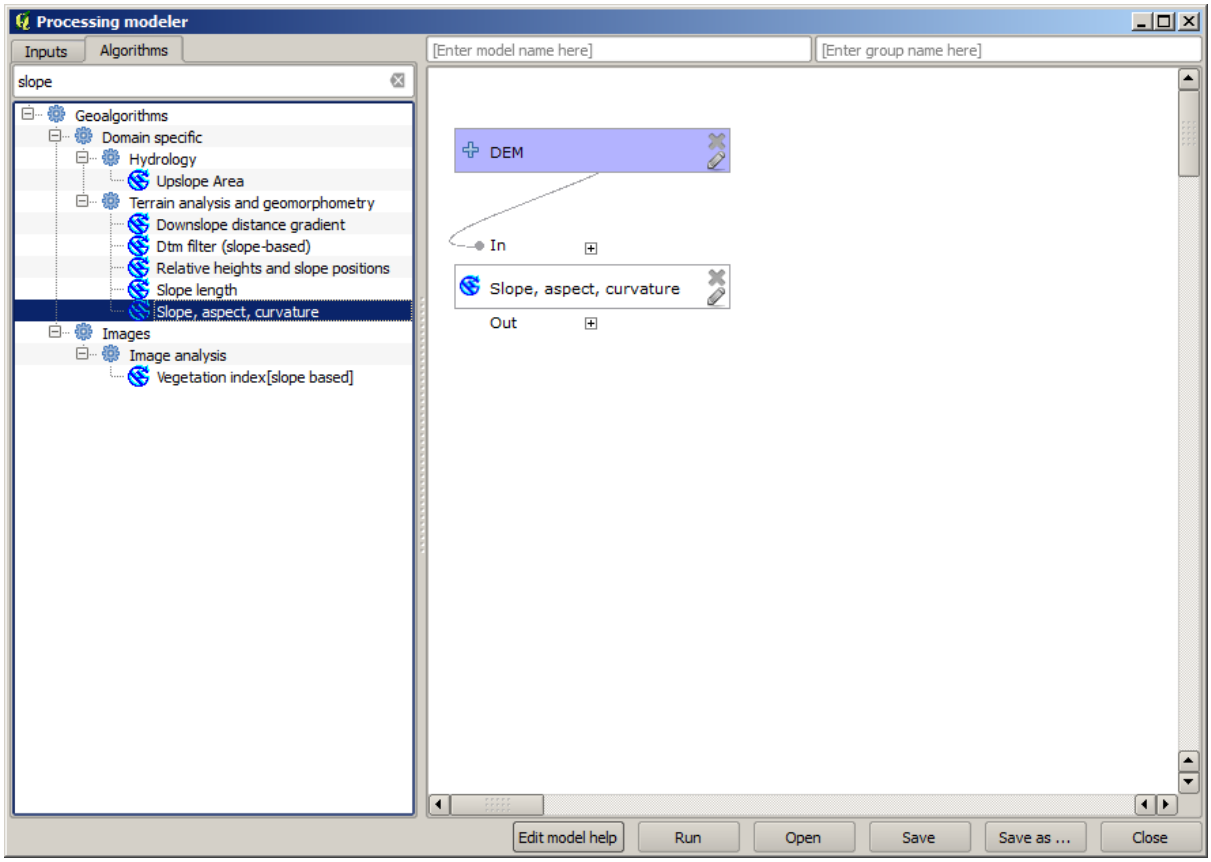


This dialog is very similar to the one that you can find when running the algorithm from the toolbox, but the element that you can use as parameter values are not taken from the current QGIS project, but from the model itself. That means that, in this case, we will not have all the raster layers of our 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.

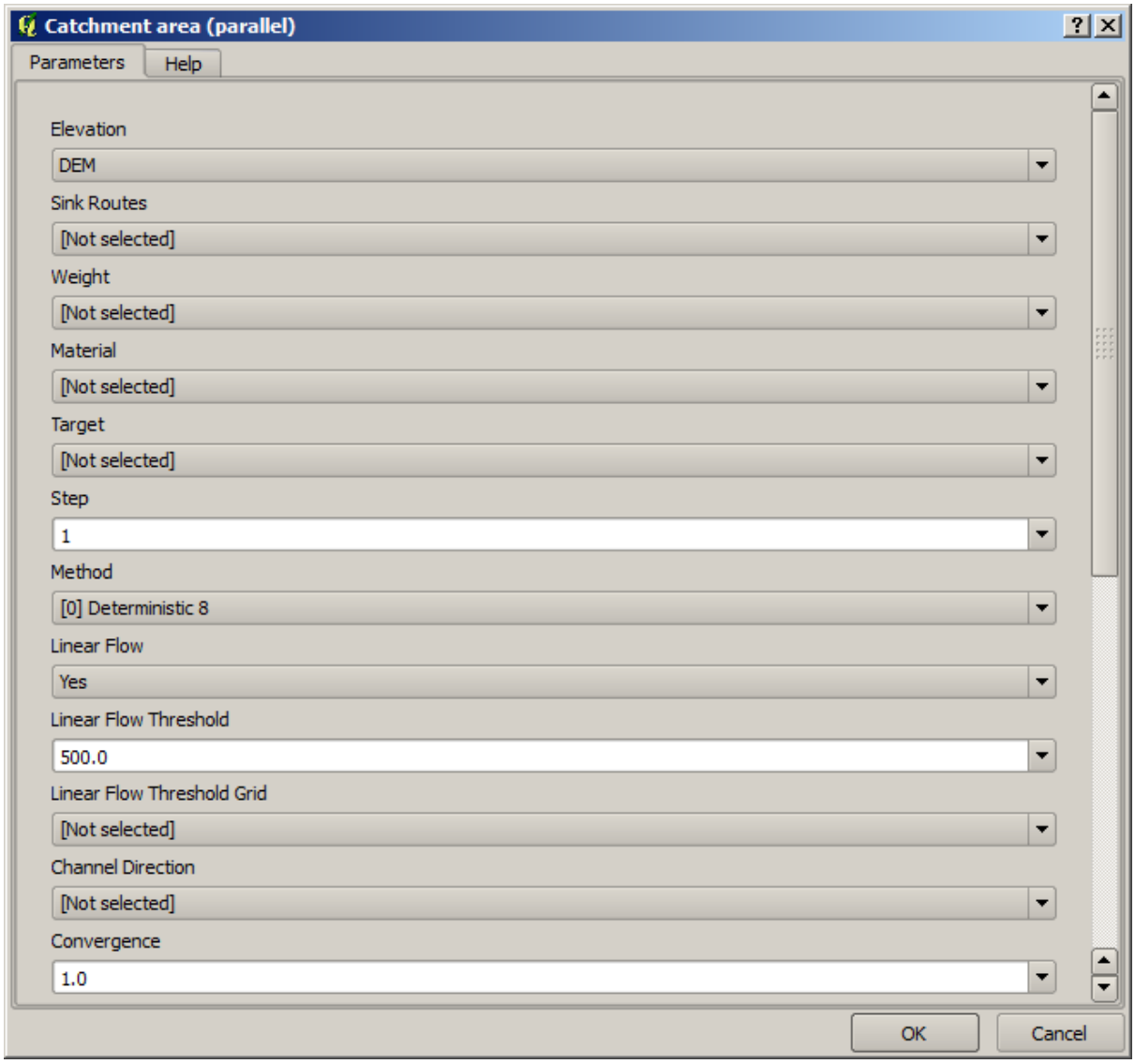
Bei Algorithmen im Modell werden die erzeugten Ausgaben etwas anders behandelt. Anstelle der Angabe eines Dateipfades wird nur angegeben, ob die Ausgabedatei temporär (sie wird nach Ausführung des Modells verworfen) oder dauerhaft sein soll. Bei unserem Beispiel sollen alle Ausgabedateien temporär sein. Den einzigen Layer den wir verwenden (der Layer mit der Hangneigung), wird nur zur Berechnung des TWI Layers benötigt. Der TWI Layer ist unser Endergebnis, das wir behalten wollen.

Wenn ein Layer kein dauerhaftes Ergebnis darstellt, sollte man das zugehörige Feld leer lassen. Ansonsten muss ein Name vergeben werden, der dann zur Identifikation im Parameter Dialog verwendet und später beim Aufruf des Modells angezeigt wird.

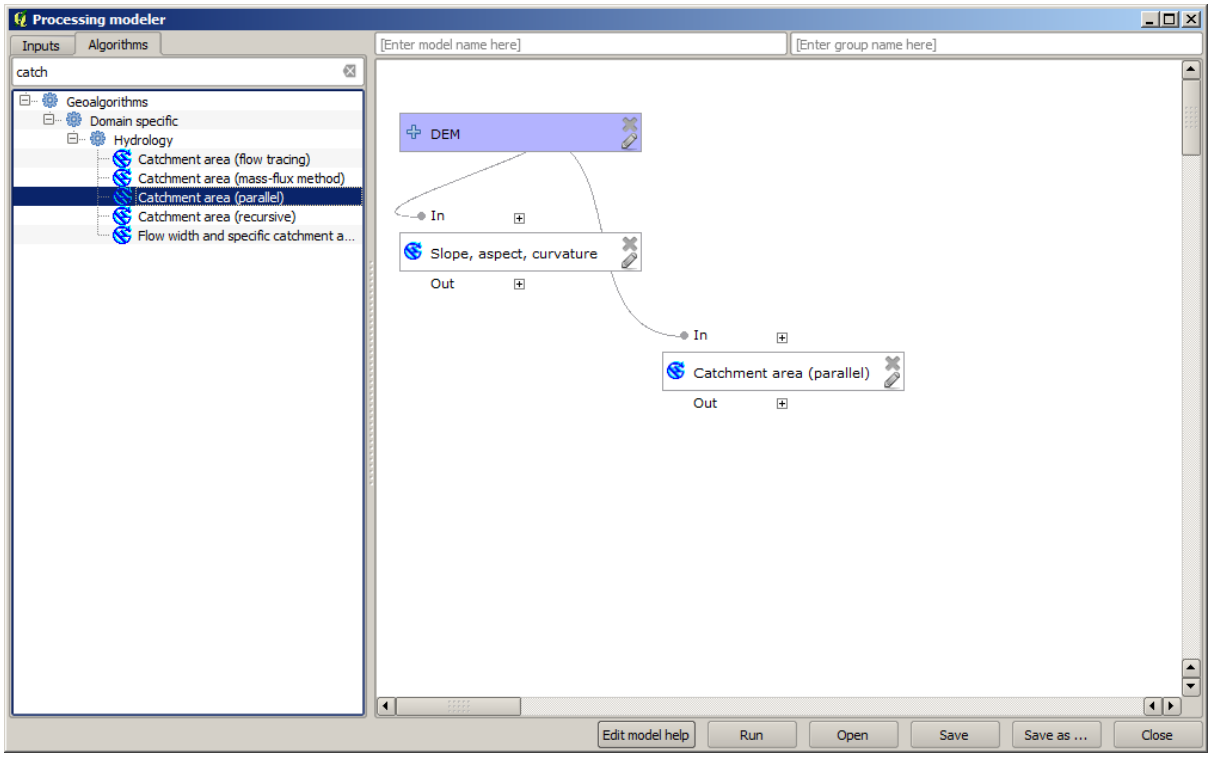
8. There is not much to select in this first dialog, since we do not have but just one layer in our model (The DEM input that we created). Actually, the default configuration of the dialog is the correct one in this case, so you just have to press *OK*. This is what you will now have in the modeler canvas.



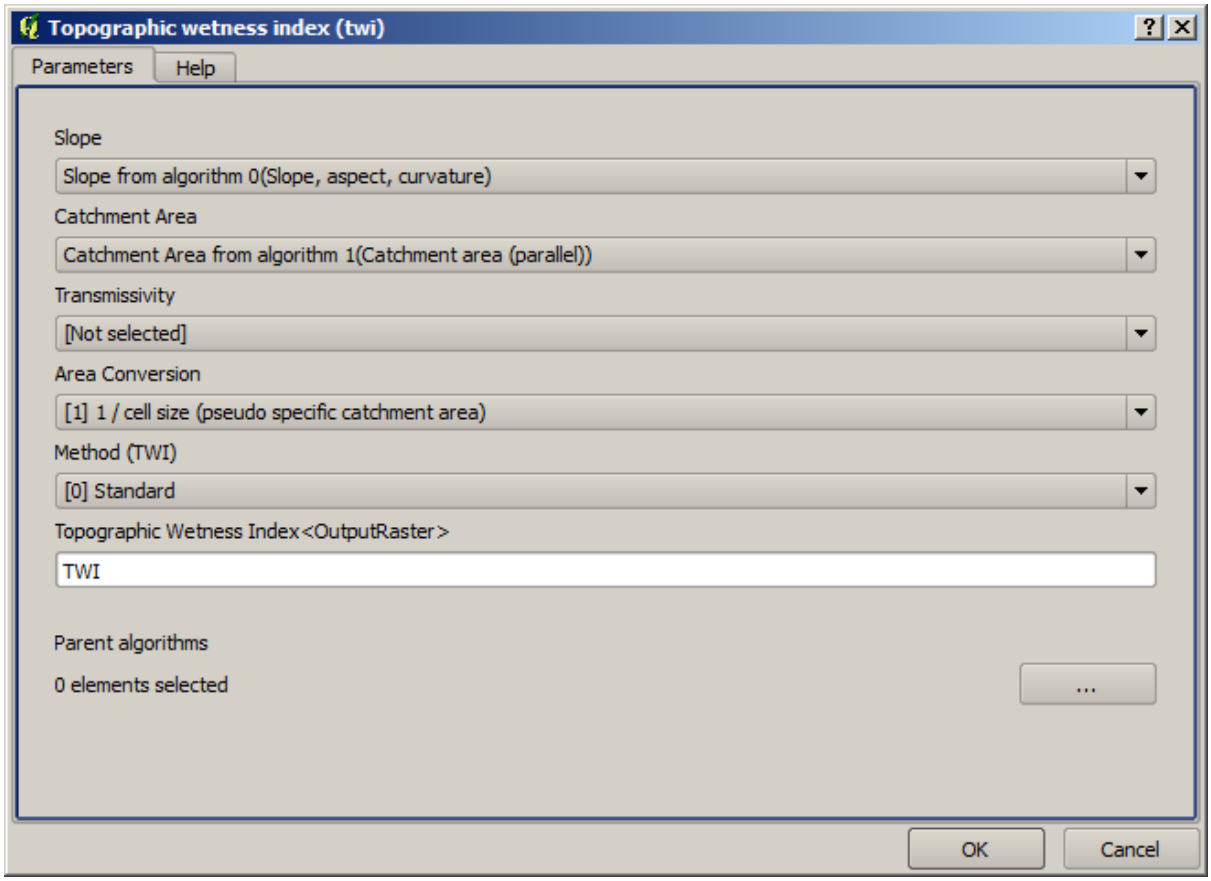
9. Als nächstes müssen wir einen Algorithmus zur Erzeugung des Einzugsgebietes in unser Modell einfügen. Wir verwenden den Algortihmus *Catchment area (Paralell)*. Als Eingabe verwenden wir erneut den DEM Layer. Die Ausgaben sind alle temporär. Der ausgefüllte Dialog sieht wie folgt aus.



Ihr Modell sollte nun wie folgt aussehen.



10. The last step is to add the *Topographic wetness index* algorithm, with the following configuration.

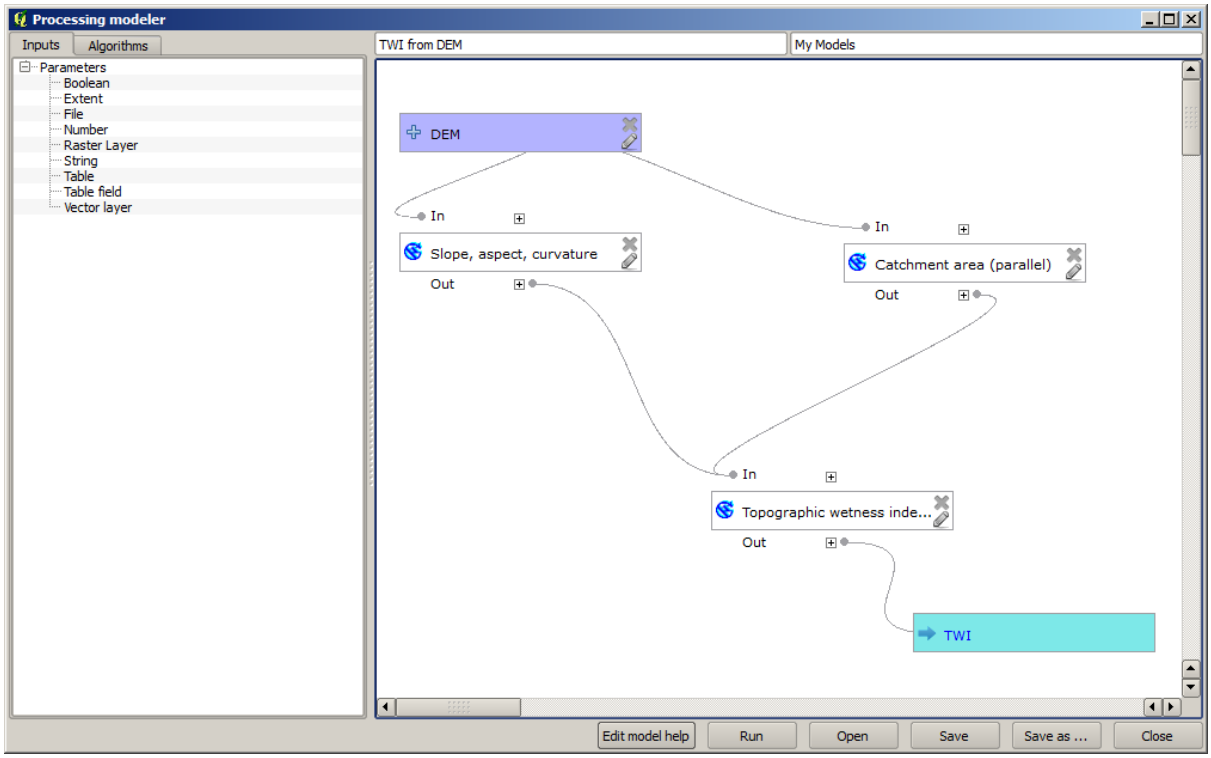


In diesem Fall nutzen wir nicht das DEM als Eingabe. Wir verwenden statt dessen die Layer Hangneigung und Einzugsgebiet, die von den anderen Algorithmen erstellt werden. Beim Einfügen von neuen Algorithmen wird

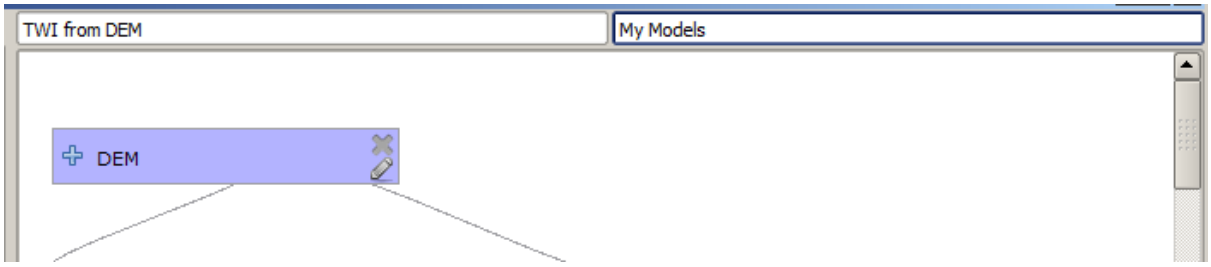
deren Ausgabe für andere Algorithmen verfügbar. Durch die Nutzung der Ausgabe für andere Algorithmen erstellt man den Arbeitsablauf.

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.

    Unser Modell ist nun fertig und sollte in etwa so aussehen.

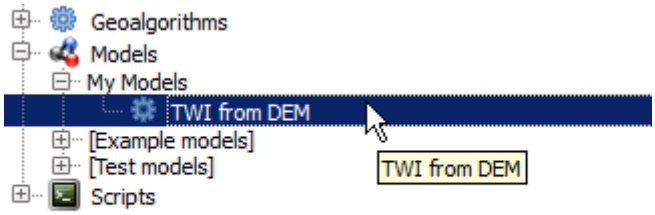


12. Enter a name and a group name in the upper part of the model window.

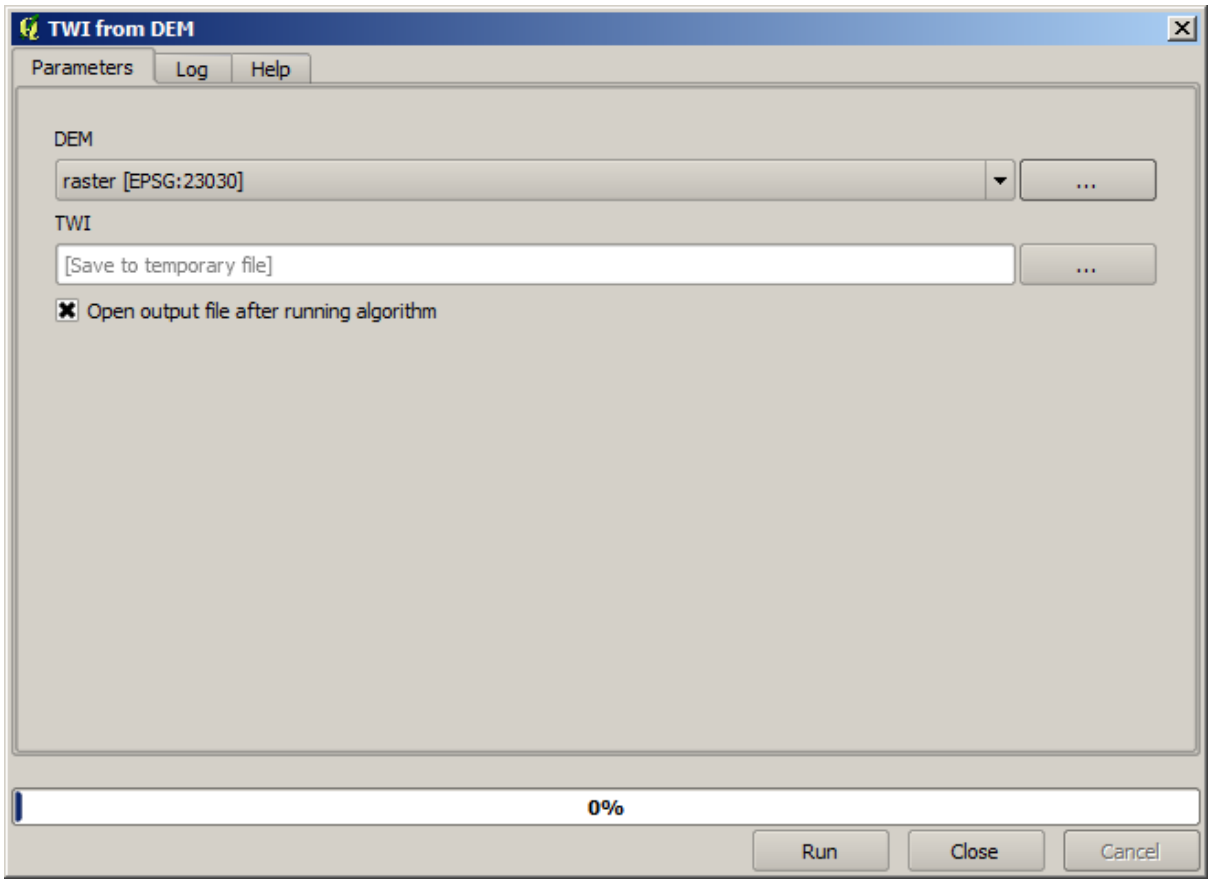


13. Save it clicking on the *Save* button. You can save it anywhere you want and open it later, but if you save it in the models folder (which is the folder that you will see when the save file dialog appears), your model will also be available in the toolbox as well. So stay on that folder and save 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.



Wie man sieht enthält der Parameterdialog die vorgebenen Eingabe- und Ausgabewerte.

16. Beim Starten des Modells wird das DEM als Eingabe genutzt und der TWI Layer in einem einzigen Schritt erstellt.

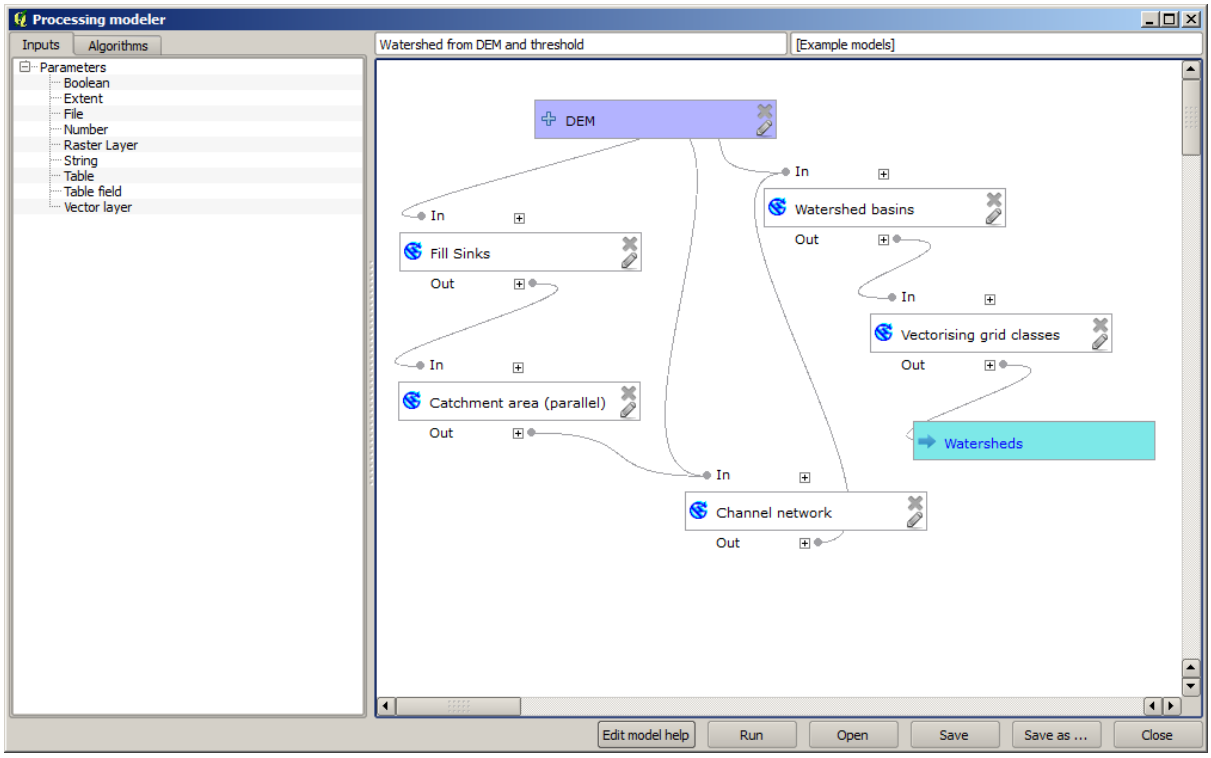
17.18 Mehr komplexe Modelle

Bemerkung: In this lesson we will work with a more complex model in the model designer.

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.

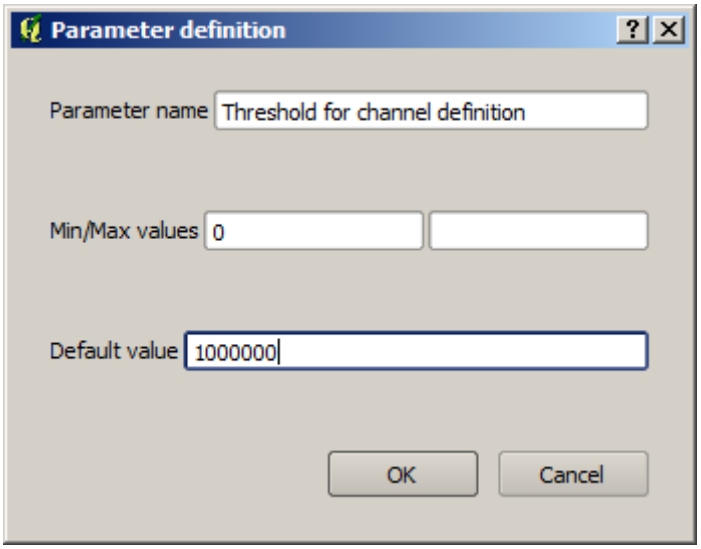
This lesson does not contain instructions about how to create your model. You already know the necessary steps (we saw them in a *previous lesson*) and you have already seen the basic ideas about the modeler, so you should try it yourself. Spend a few minutes trying to create your model, and don't worry about making mistakes. Remember: first add the inputs and then add the algorithms that use them to create the workflow.

Bemerkung: Wenn Sie etwas Hilfe benötigen und es nicht schaffen, dass vollständige Modell zu erstellen, finden sie im Ordner zu dieser Lektion eine fast fertige Version des Modells. Öffnen Sie die Verarbeitungsmodellierung und anschließend die Modelldatei im Datenordner. Sie sollten in etwa das Folgende sehen.

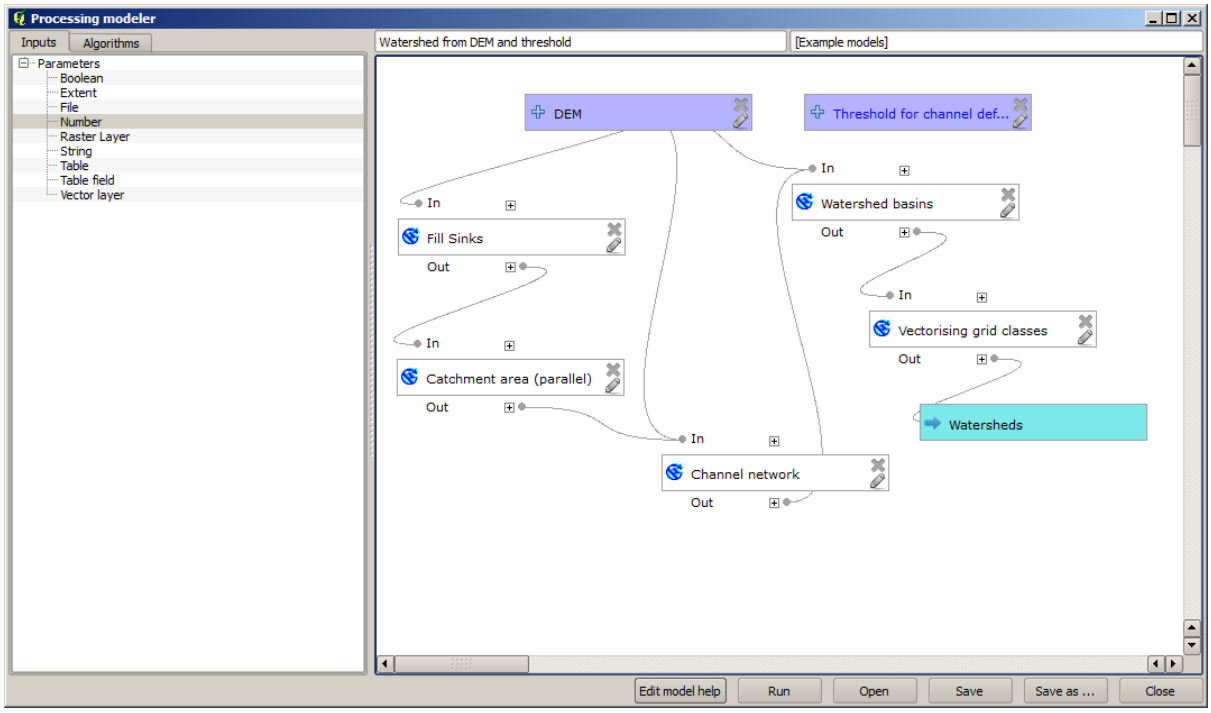


This model contains all the steps needed to complete the calculation, but it just has one input: the DEM. That means that the threshold for channel definition uses a fixed value, which makes the model not as useful as it could be. That is not a problem, since we can edit the model, and that is exactly 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`

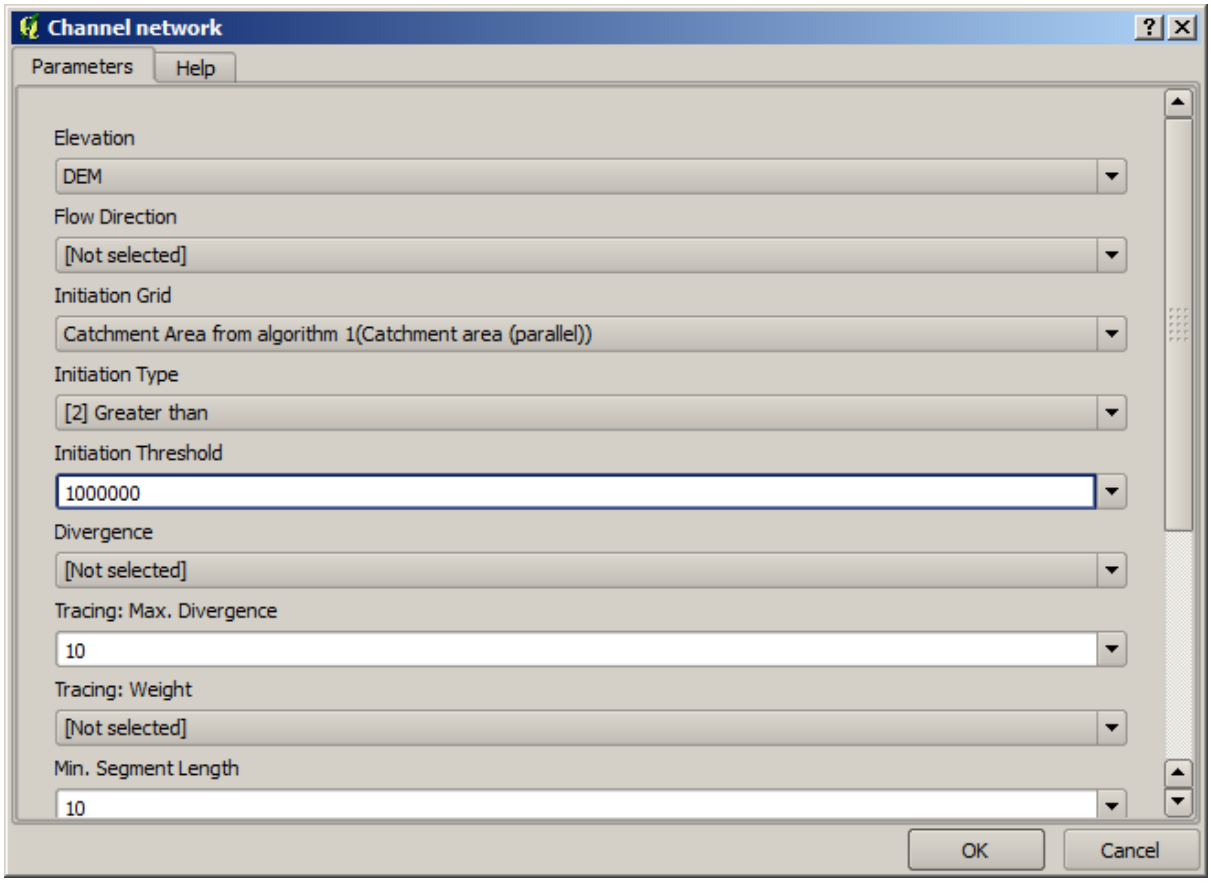


Ihr Modell sollte jetzt in etwa wie folgt aussehen.



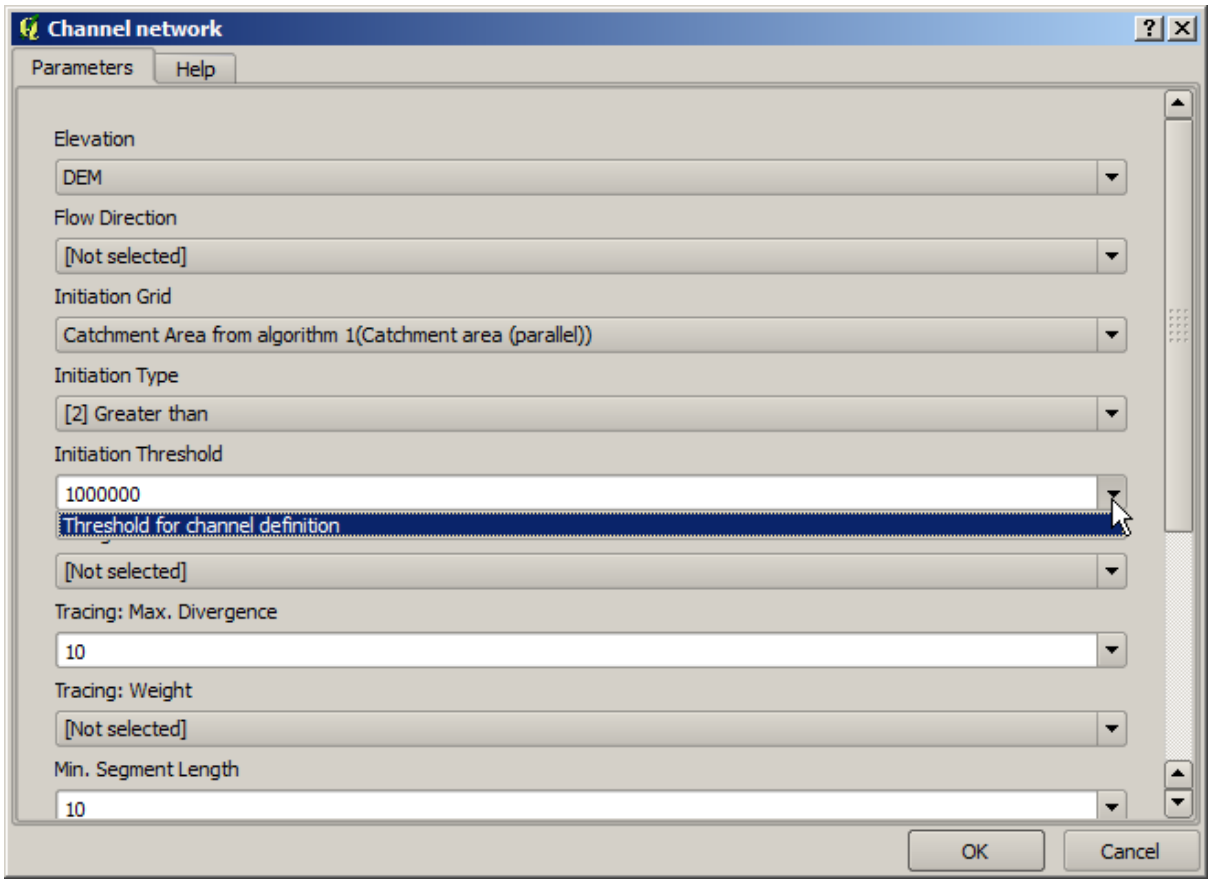
The input that we have just added is not used, so the model hasn't actually changed. We have to link that input to the algorithm that uses it, in this case the *Channel network* one. To edit an algorithm that already exists in the modeler, just click on the pen icon on the corresponding box in the canvas.

4. Click on the *Channel network* algorithm and you will see something like this.



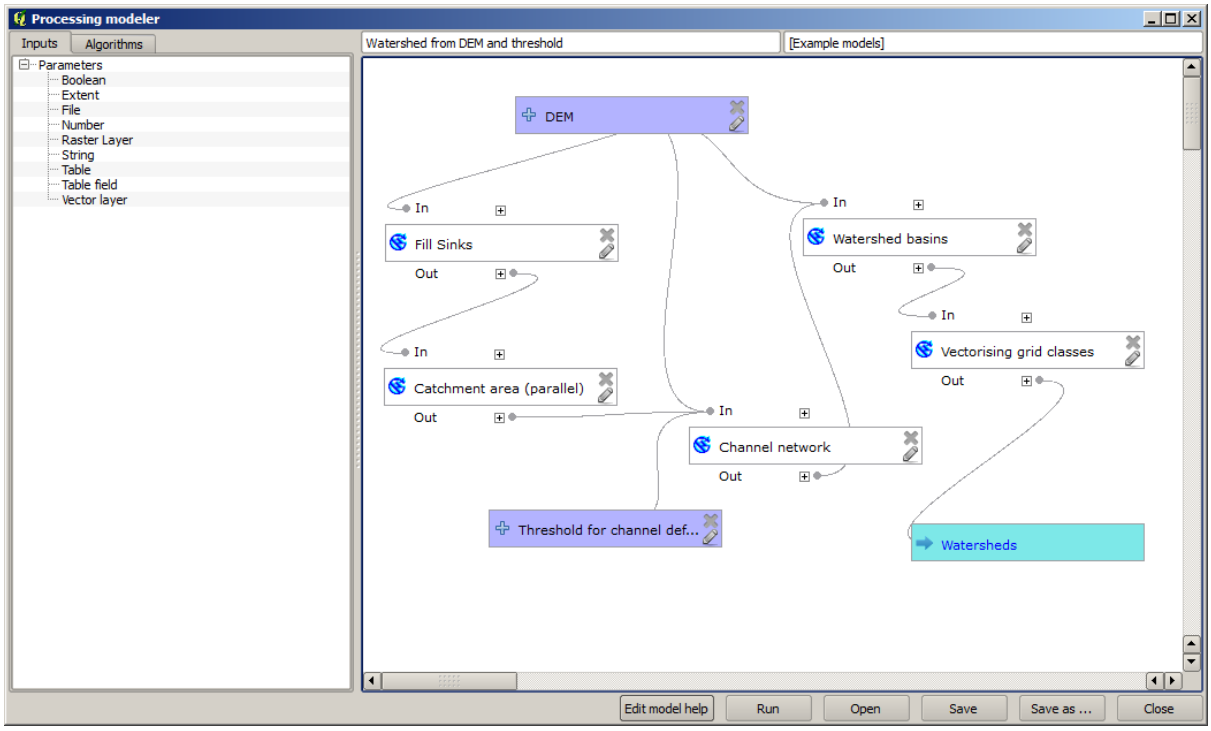
The dialog is filled with the current values used by the algorithm. You can see that the *Initiation threshold* parameter has a fixed value of 1,000,000 (this is also the default value of the algorithm, but any other value could be put in there). However, you might notice that the parameter is not entered in a common text box, but in a drop-down menu.

5. Unfold the threshold parameter menu and you will see something like this.



Den ins Modell eingefügten numerischen Parameter können wir hier auswählen. Immer wenn ein Modell eine numerische Eingabe erwartet, kann man einen festen Wert verwenden oder man nutzt eine der vorhandenen Eingabemöglichkeiten (denken sie daran, dass einige Algorithmen einzelne numerische Werte generieren. Wir werden bald mehr darüber lernen.) Im Fall von Parametern im Format Zeichenkette können ebenfalls Eingabevariablen im Format Zeichenkette oder feste Werte verwendet werden.

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*.

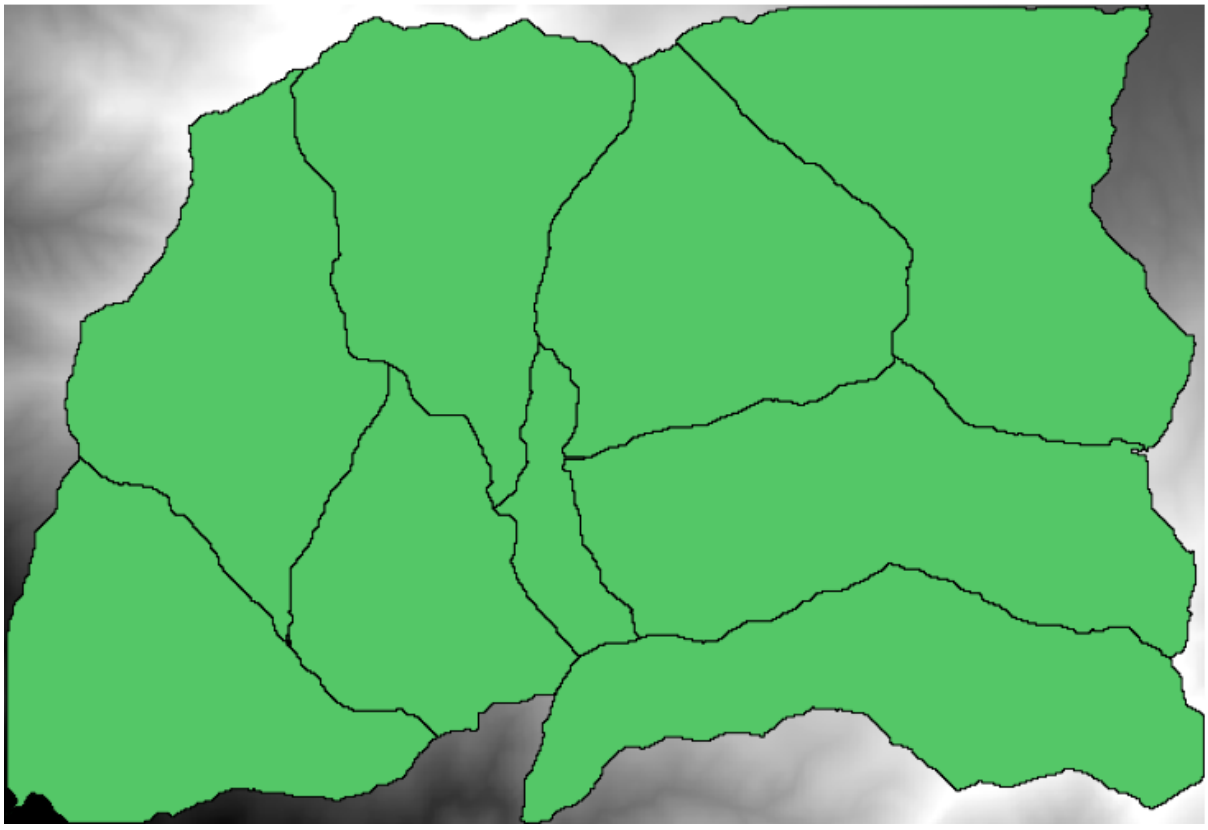


Abb. 17.1: Threshold = 100.000

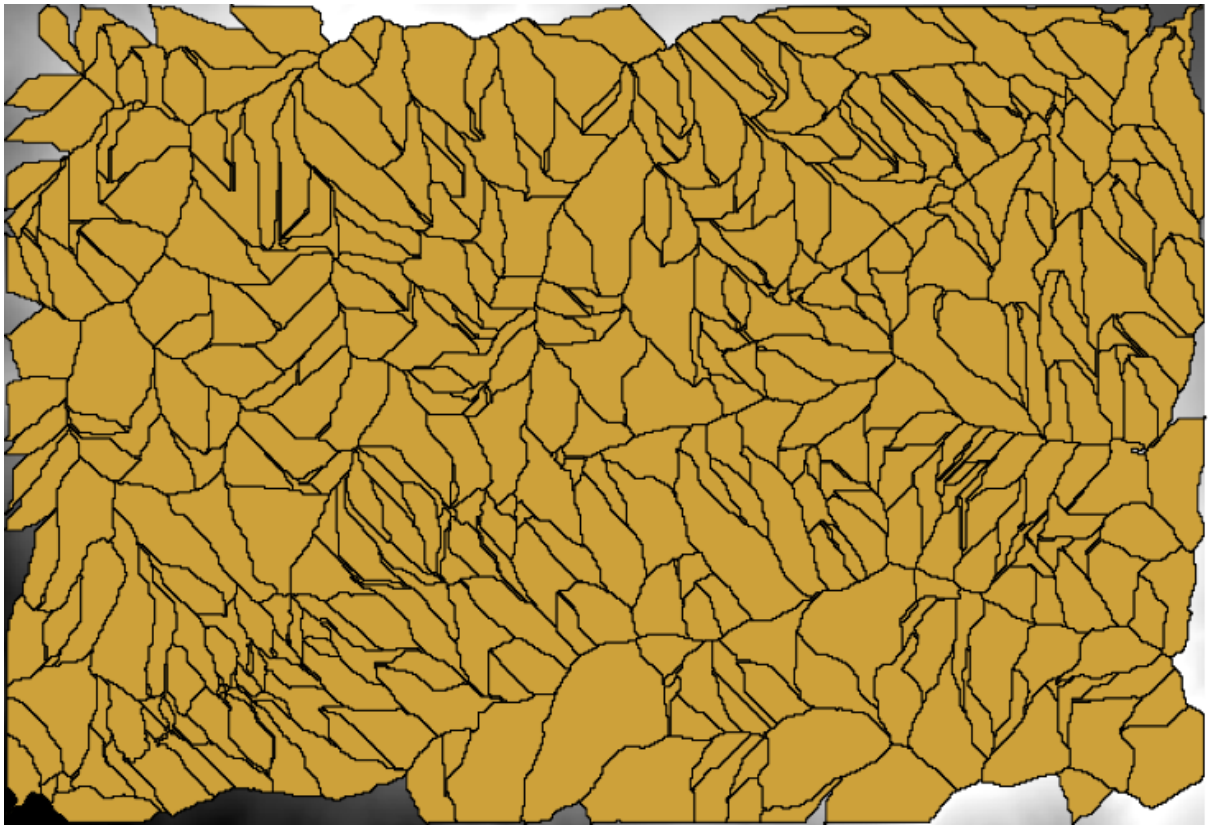


Abb. 17.2: Threshold = 1.0000.000

17.19 Nummerische Berechnungen im Modellierer

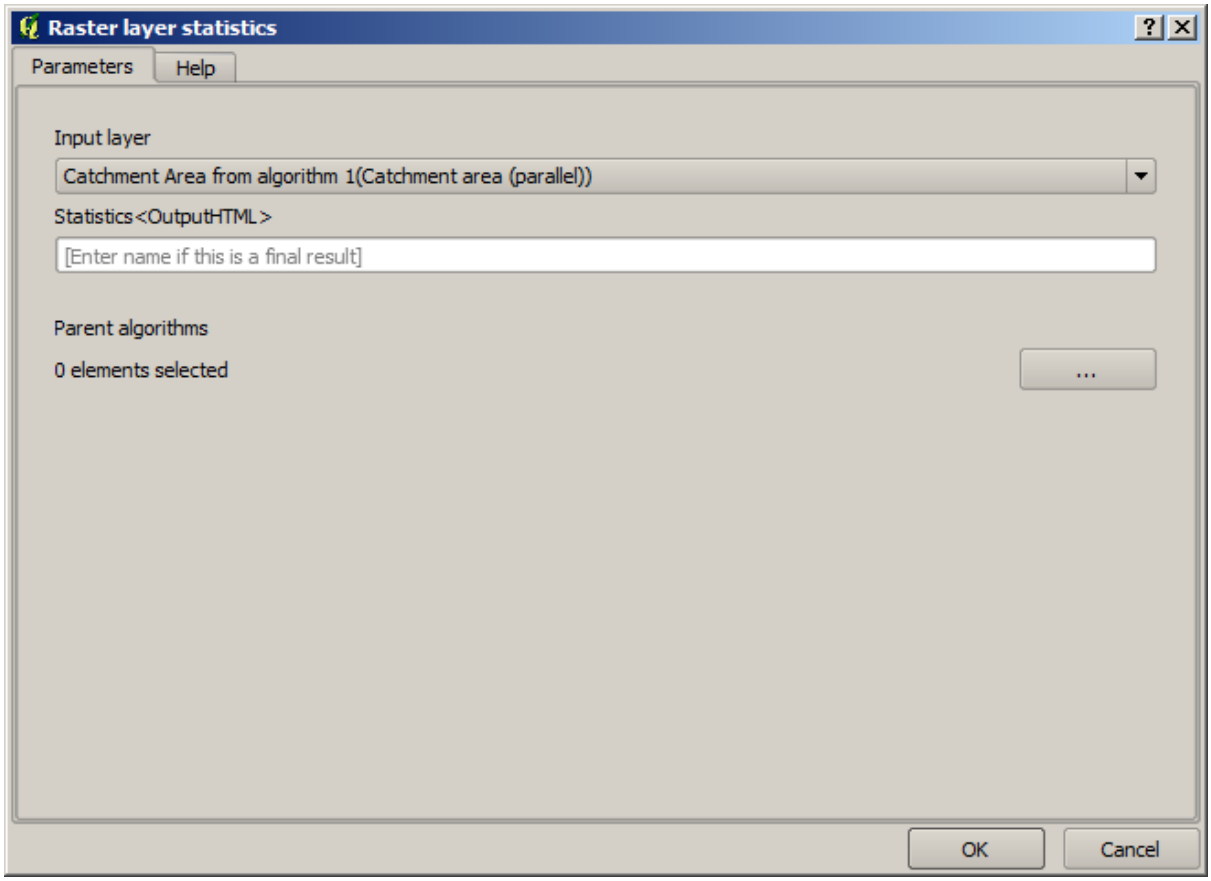
Warnung: Achtung, dieses Kapitel wurde nicht weiter getestet. Bitte melden Sie auftrende Probleme. Es sind keine Bilder enthalten.

Bemerkung: In dieser Lektion lernen wir, wie man numerische Ausgaben in der Verarbeitungsmodellierung verwendet.

In dieser Lektion werden wir das hydologische Modell aus der letzen Lektion (öffenen Sie es bevor Sie beginnen) so verändern, dass der Schwellenwert automatisch gefüllt wird und nicht mehr vom Nutzer vorgegeben werden muss. Da sich der Wert auf die Variable im Rasterlayer Schwellenwert bezieht, werden wir ihn aus diesem Layer mit Hilfe einiger einfacher statistischer Analysen extrahieren.

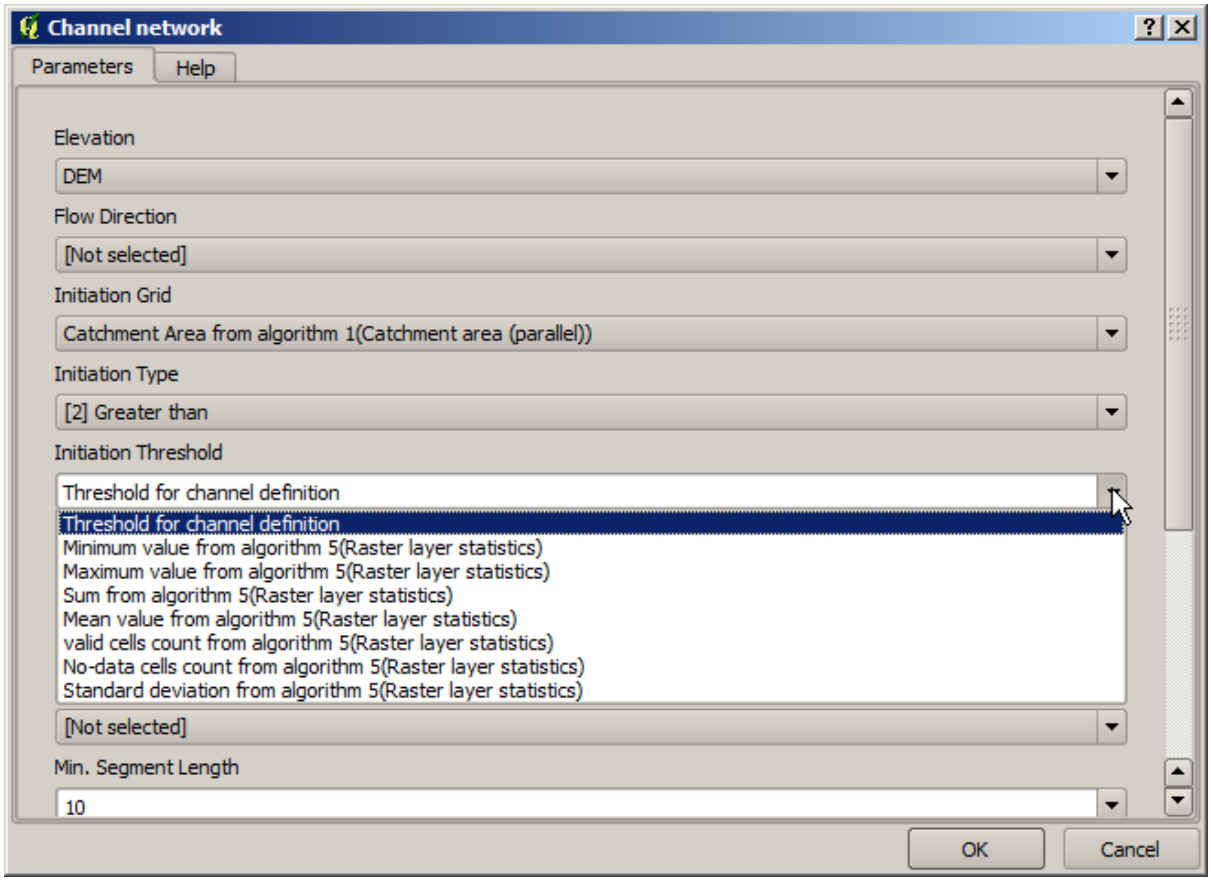
Wir nehmen am oben genannten Modell die folgenden Änderungen vor:

Berechnen Sie als Erstes Statistiken des Layers flow accumulation mit Hilfe des Algorithmus *Raster layer statistics*.



Damit wird eine Menge von statistischen Werten erstellt, die dann für die numerischen Felder in allen Algorithmen verfügbar sind.

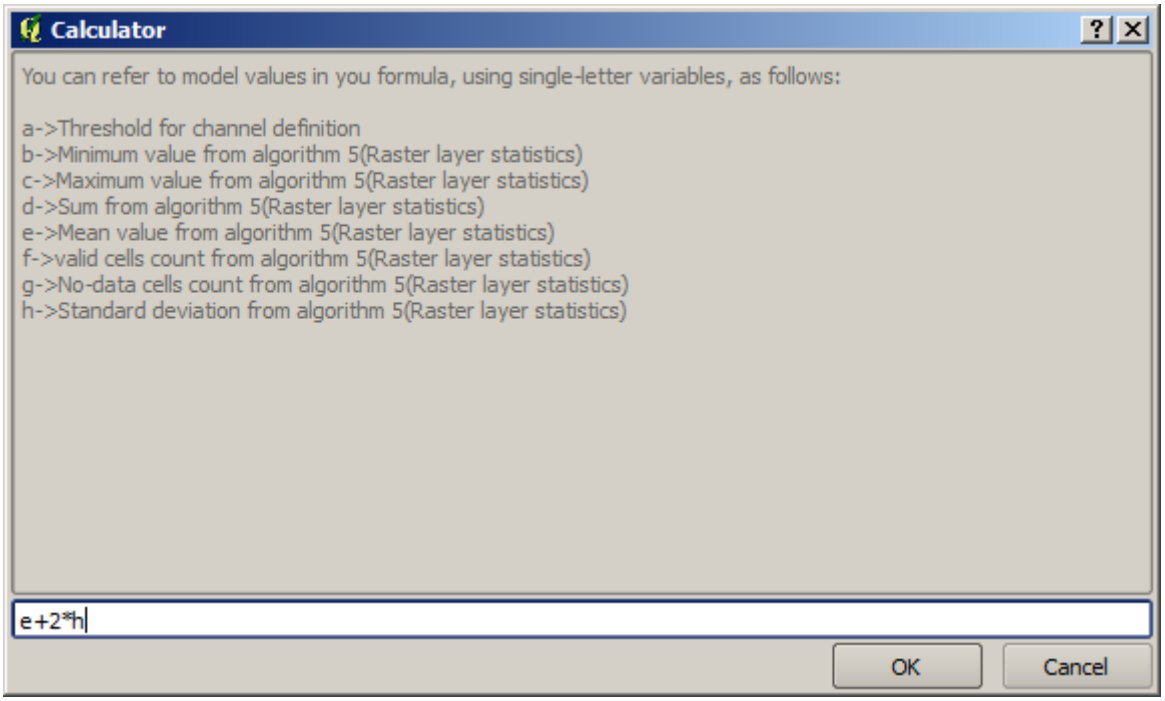
Wenn wir den Algorithmus *Channel network* wie in der letzen Lektion bearbeiten, sehen wir nun weitere Optionen außer der numerischen Eingabe, die wir hinzugefügt hatten.



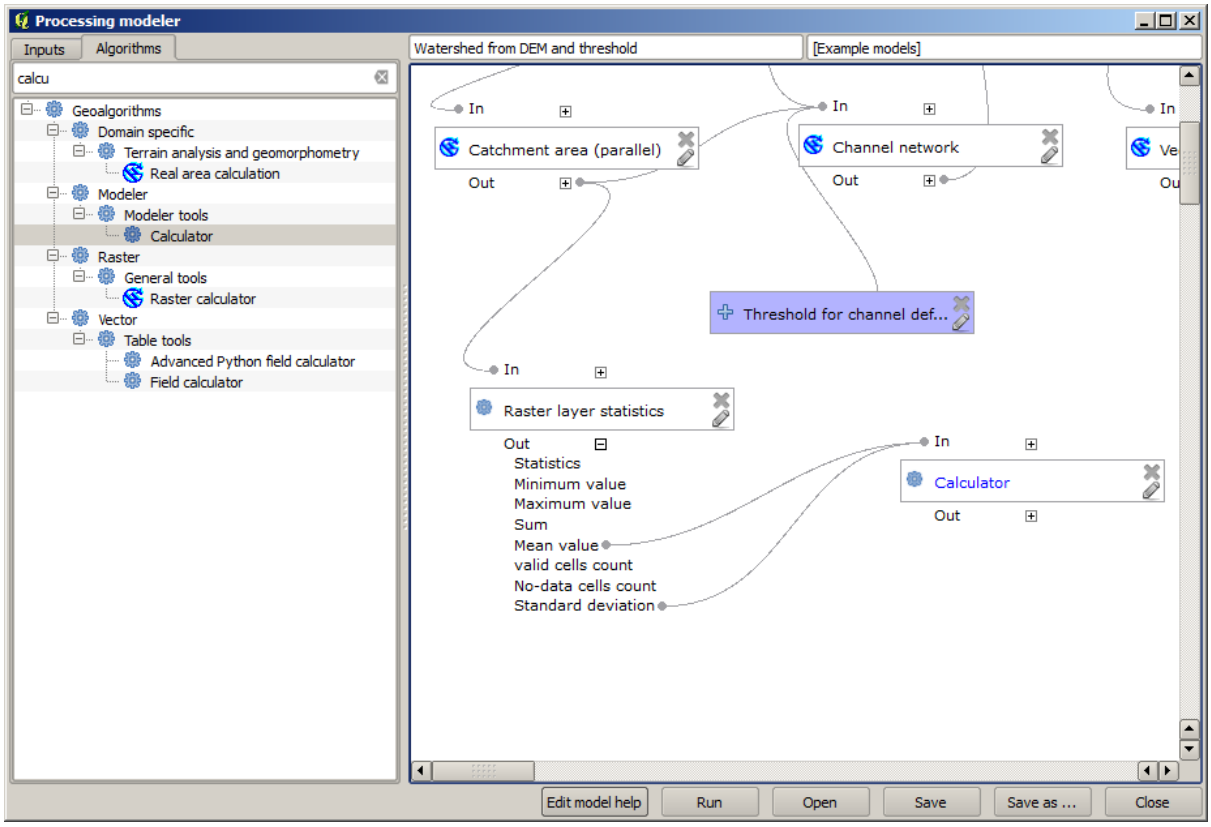
Allerdings ist keiner der Werte wirklich als Schwellenwert nutzbar. Die entstehenden Abflussnetzwerke wären nicht sehr realistisch. Wir können aber basierend auf diesen Werten einige neue Parameter ableiten. Wir können z.B. den Mittelwert plus 2 mal der Standardabweichung verwenden.

Um diese Rechenoperation zu ergänzen, können wir den Rechner aus der Gruppe *Geoalgorithms/modeler/modeler-tools* verwenden. Diese Gruppe enthält Algorithmen, die außerhalb der Verarbeitungsmodellierung nicht sehr nützlich sind. Sie stellen aber nützliche Funktionalitäten für die Erstellung eines Modells bereit.

Der Parameter Dialog des Rechner Algorithmus sieht wie folgt aus:

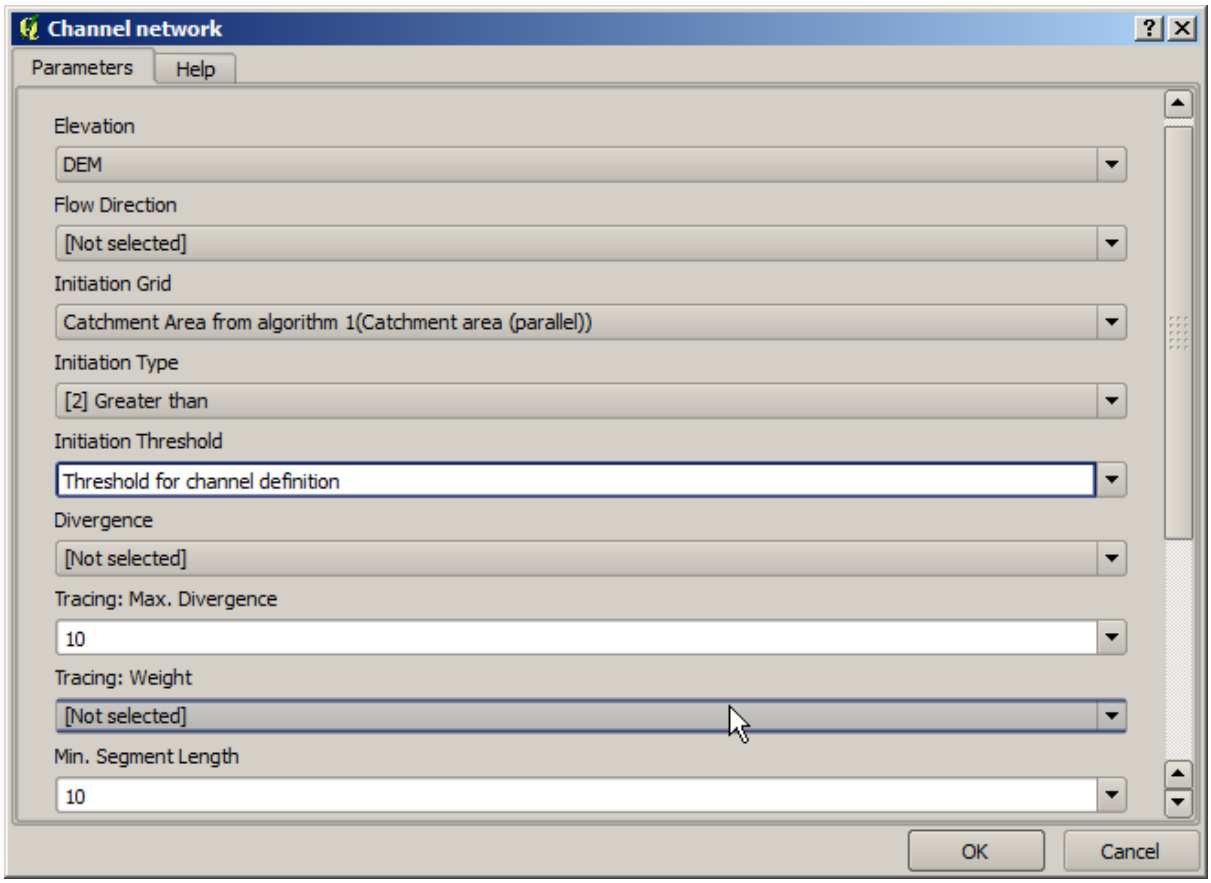


Der Dialog weicht von den anderen vorher verwendeten Dialogen ab. Es sind aber dieselben Variablen verfügbar wie im Feld *Threshold* im Algorithmus *Channel network*. Geben Sie die obige Formel ein und klicken auf *OK*, um den Algorithmus hinzuzufügen.

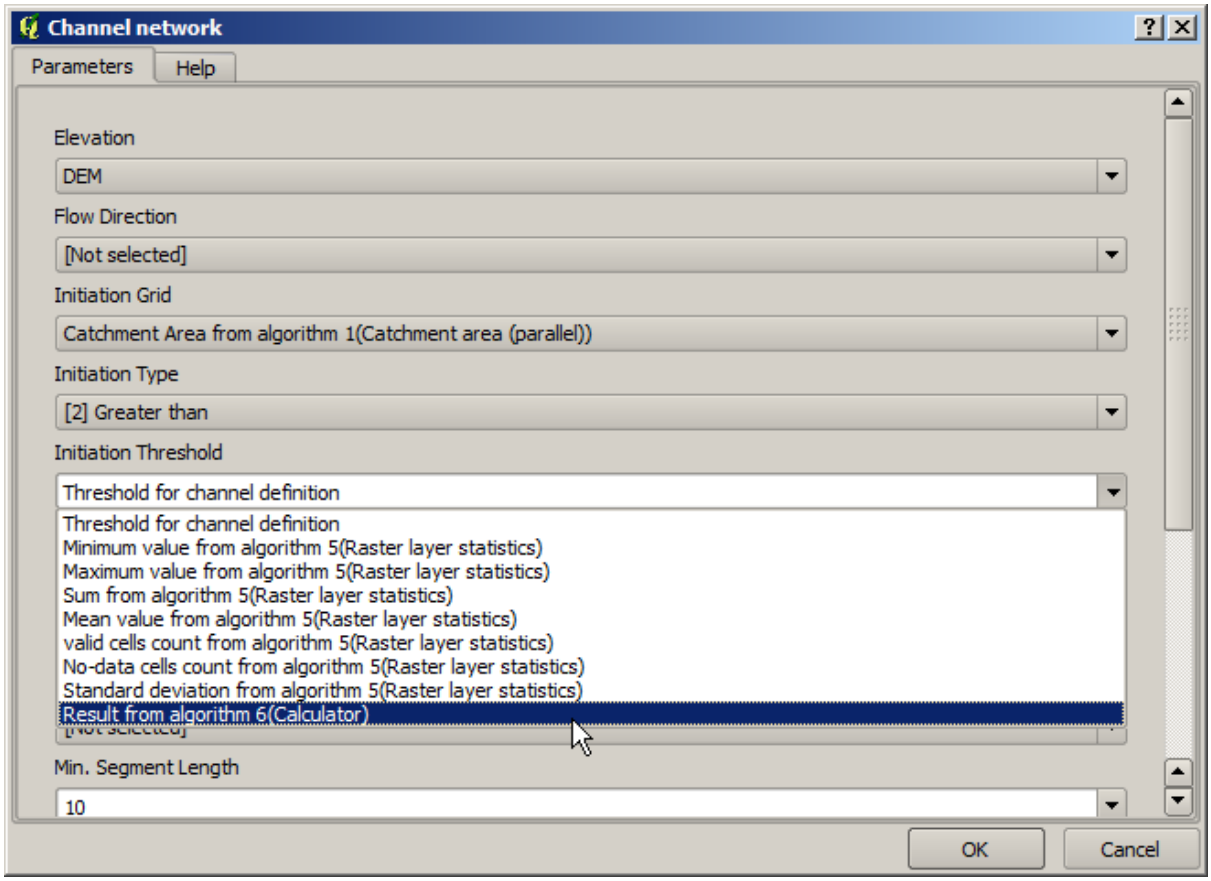


Wenn Sie den Ausgabeeintrag wie oben gezeigt erweitern, sehen Sie, dass das Modell eine Verbindung zu zwei der Werte hergestellt hat, dem Mittelwert und der Standardabweichung. Das sind die Werte, die wir in der Formel verwendet haben.

Nach dem Hinzufügen des Algorithmus steht ein neuer numerischer Wert zur Verfügung. Wenn Sie noch einmal den Algorithmus *Channel network* öffnen, können Sie diesen Wert als Parameter für *Threshold* auswählen.



Klicken Sie auf *OK* und Ihr Modell sollte wie folgt aussehen.



Wir nutzen die numerische Eingabe, die wir hinzugefügt hatten nicht mehr. Wir können Sie nun durch Klick mit der rechten Maustaste und Auswahl von *Remove* löschen.

Warnung: todo: Bild einfügen

Unser neues Modell ist nun fertig gestellt.

17.20 Ein Model in einem Modell

Warnung: Achtung, dieses Kapitel wurde nicht weiter getestet. Bitte melden Sie auftrende Probleme. Es sind keine Bilder enthalten.

Bemerkung: In dieser Lektion lernen wir, wie man ein Modell innerhalb eines größeren Modells verwendet.

Wir haben bereits einige Modelle erstellt. In dieser Lektion lernen wir, wie man sie zu einem größeren Modell vereingen kann. Ein Modell verhält sich wie jeder andere Algorithmus. Das bedeutet, dass man ein bereits erstelltes Modell zu einem später zu erstellendem Modell hinzufügen kann.

In diesem Fall werden wir unser hydrologisches Modell so erweitern, dass es als Ergebnis den mittleren TWI für alle generierten Einzugsgebiete bestimmt. Um das zu erreichen, müssen wir den TWI und die Statistiken berrechnen. Da wir bereits ein Modell zur Berechnung des TWI erstellt haben, können wir es weiter verwenden und brauchen somit die Algorithmen die es enthält nicht manuell einfügen.

Wir beginnen mit dem Modell, das unser Ausgangspunkt in der vorherigen Lektion war.

Warnung: todo: Bild einfügen

Zuerst fügen wir das TWI Modell ein. Damit es verfügbar ist, muss es im Modellpfad abgespeichert sein. Ansonsten wäre es nicht in der Werkezeugkiste oder der Liste der Algorithmen verfügbar. Stellen Sie sicher, dass das Modell verfügbar ist.

Fügen Sie es in das aktuelle Modell ein und verwenden Sie das DEM als Eingabe. Die Ausgabe ist temporär, da wir den TWI Layer nur zur Ermittlung der Statistik benötigen. Die einzige Ausgabe des Modells wird der Vektorlayer mit den Wasserscheiden sein.

Hier ist der dazu gehörende Paramter Dialog:

Warnung: todo: Bild einfügen

Wir haben nun einen TWI Layer, den wir neben unserem Vektorlayer der Wasserscheiden verwenden können. Wir erstellen damit einen neuen Layer der den zu jeder Wasserscheide gehörenden TWI Wert ermittelt.

Diese Berechnung verwendet den Algorithmus *Grid statistics in polygons*. Nutzen Sie den oben genannten Layer als Eingabe, um das endgültige Ergebnis zu erhalten.

Warnung: todo: Bild einfügen

Die Ausgabe des Algorithmus *Vectorize grid classes* war ursprünglich unser Endergebnis. Wir werden es jetzt als Zwischenergebnis verwenden. Um das zu erreichen, müssen wir nur den Algorithmus anpassen. Klicken Sie doppelt auf den Algorithmus, um den Parameter Dialog zu öffnen. Löschen Sie den Eintrag für die Ausgabe. Damit wird wie in der Voreinstellung eine temporäre Ausgabe erzeugt.

Warnung: todo: Bild einfügen

Das fertig gestellte Modell sollte in etwa wie folgt aussehen:

Warnung: todo: Bild einfügen

Wie man sieht ist das Einfügen eines anderen Modells in ein Modell nichts besonderes. Man kann es wie einen Algorithmus in das Modell einfügen. Man muss nur daruf achten, dass das Modell im Vorgabepfad der Modelle gespeichert und in der Werkzeugkiste verfügbar ist.

17.21 Erstellung eines Modells mit modellinternen Werkzeugen

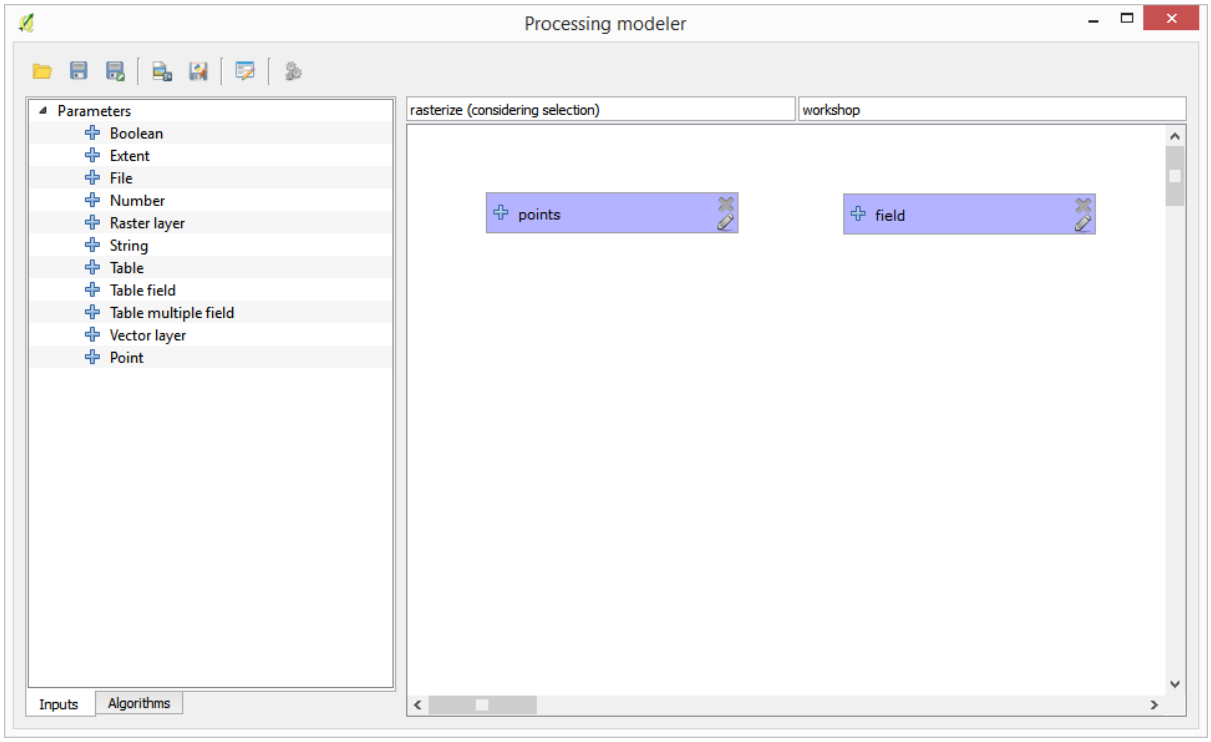
Bemerkung: Diese Lektion zeigt, wie man einige Algorithmen nutzt, die nur in der Verarbeitungsmodellierung verfügbar sind. Sie stellen zusätzliche Funktionalitäten für Modelle bereit.

Das Ziel dieser Lektion ist es, mit Hilfe der Verarbeitungsmodellierung einen Algorithmus zur Interpolation zu erstellen. Er soll dabei die aktuelle Selektion berücksichtigen, d.h. nicht nur die selektierten Objekte sondern auch deren räumliche Ausdehnung. Auf Grundlage der Ausdehnung soll ein interpolierter Rasterlayer erstellt werden.

Der Prozess der Interpolation beinhaltet zwei Prozesse, die schon in vorherigen Lektionen beschrieben wurden: die Rasterung von Punktdaten und das Füllen von Leerwerten in gerasterten Layern. Im Fall von Punkten die eine Auswahl enthalten, werden nur diese Punkte verwendet. Wenn die Ausdehnung so eingestellt ist, dass sie automatisch

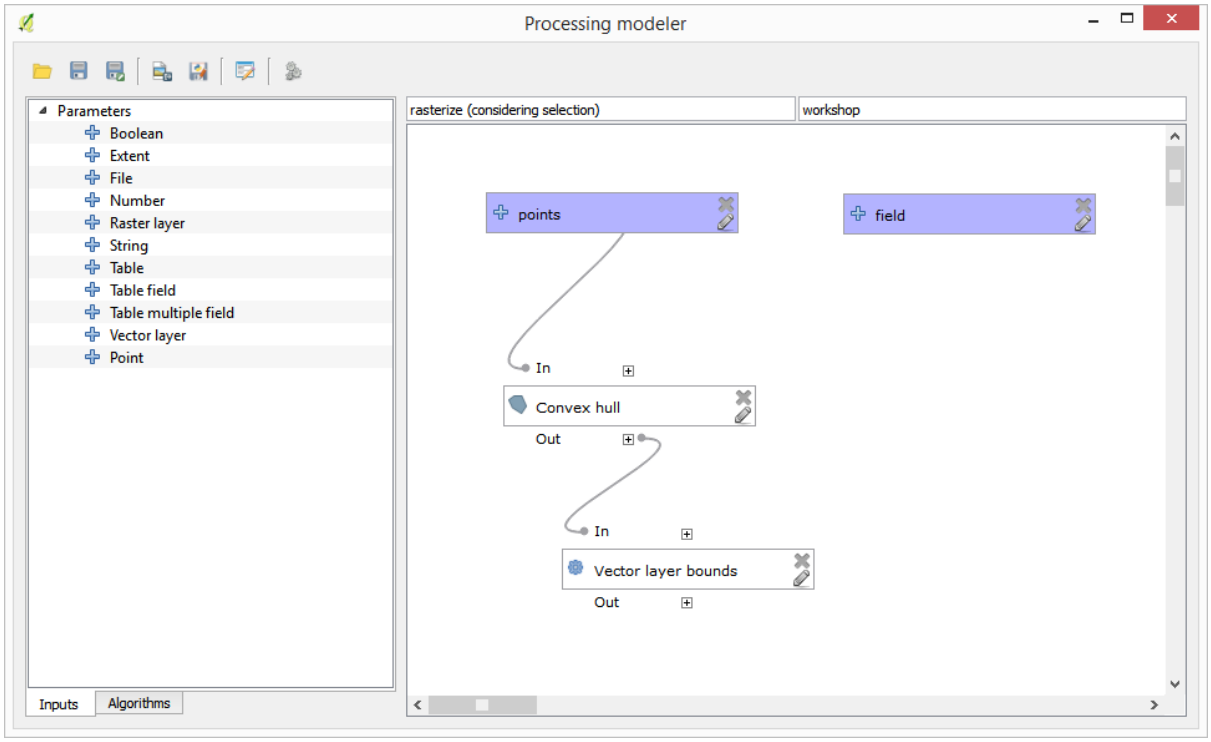
angepasst wird, wird aber die gesamte Ausdehnung des Layers verwendet und nicht nur die, die sich aus der Selektion ergibt. Wir werden versuchen, das mit Hilfe weiterer Werkzeuge zu beheben.

Öffnen Sie die Verarbeitungsmodellierung und fügen die benötigten Eingabedaten hinzu. Wir benötigen einen Vektorlayer (nur Punkte) und ein Attribut des Layers, das wir für die Rasterung verwenden werden.

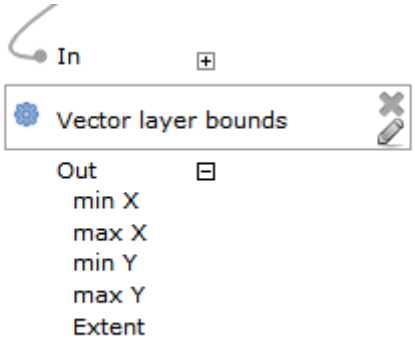


Der nächste Schritt ist die Ermittlung der Ausdehnung der selektierten Objekte. Dafür verwenden wir das modellinterne Werkzeug *Layerausdehnung extrahieren*. Als Erstes müssen wir einen Layer erstellen, der die Ausdehnung der gewählten Objekte hat. Danach können wir das Werkzeug mit diesem Layer verwenden.

Ein einfacher Weg zur Erstellung eines Layers mit der Ausdehnung der selektierten Objekte ist die Berechnung einer konvexen Hülle für den Punktlayer. Dabei werden nur die gewählten Punkte verwendet, so dass die konvexe Hülle dieselbe Ausdehnung wie der Begrenzungsrahmen der Auswahl hat. Wir nutzen dann den Algorithmus *Layerausdehnung extrahieren* mit der konvexen Hülle als Eingabe. Unser Arbeitsbereich sollte nun in etwa wie folgt aussehen:

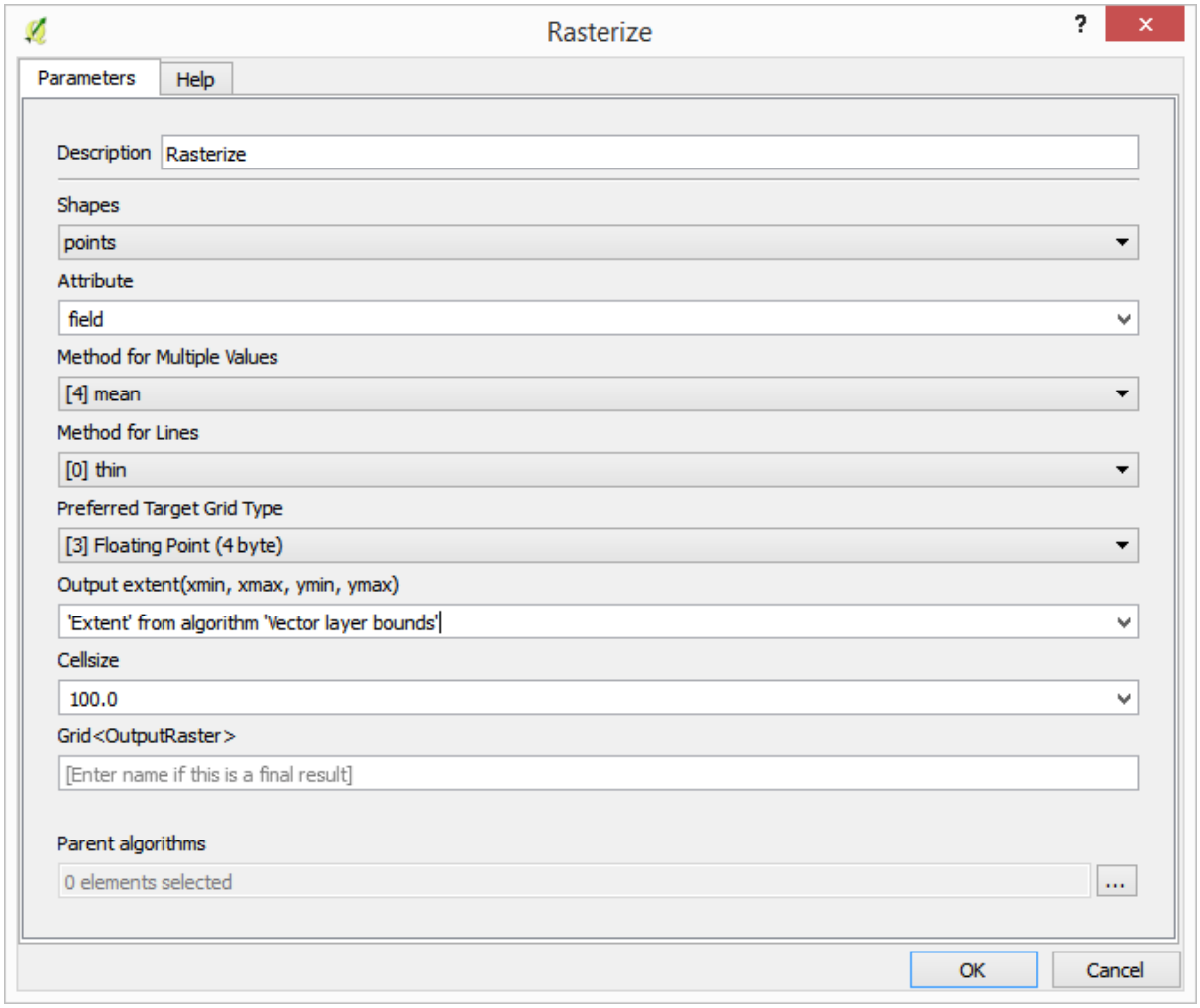


Das Ergebnis von *Layerausdehnung extrahieren* ist eine Menge von vier Zahlen und ein Ausdehnungsobjekt. Wir werden beides in dieser Übung verwenden.

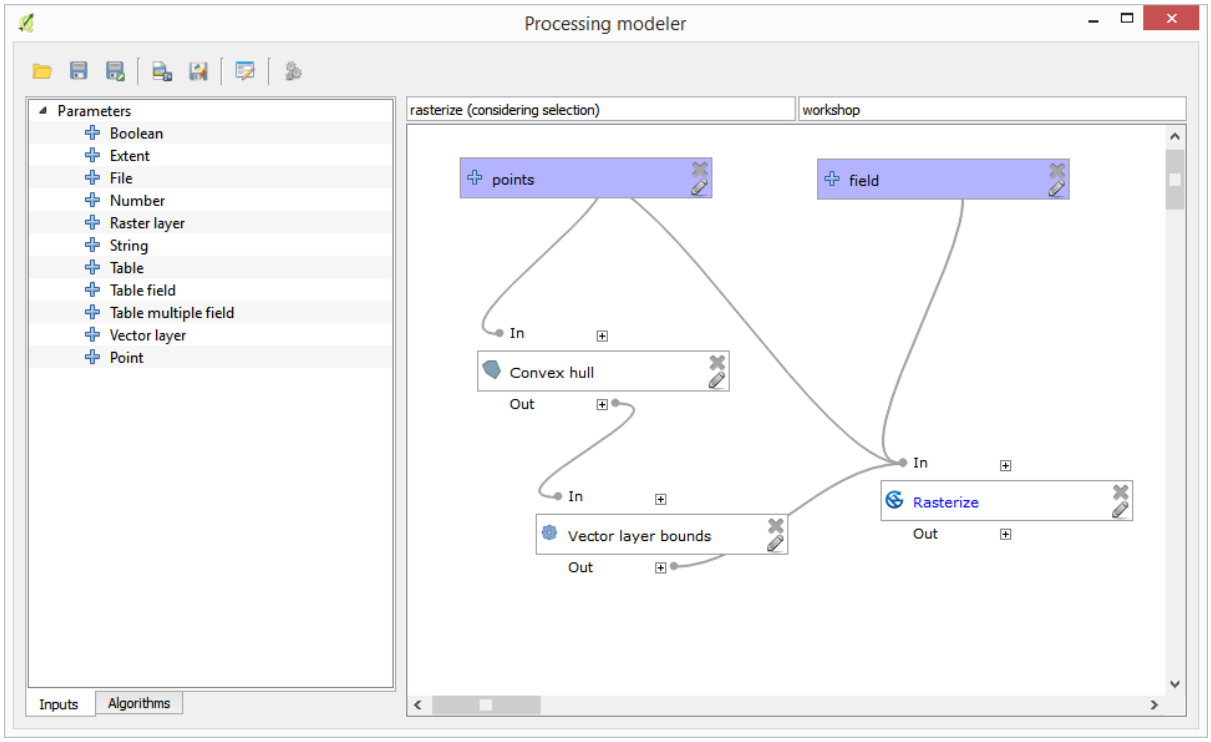


Wir können nun den Algorithmus zur Rasterung unter Nutzung der Ausdehung aus dem Algorithmus *Layerausdehnung extrahieren* einfügen.

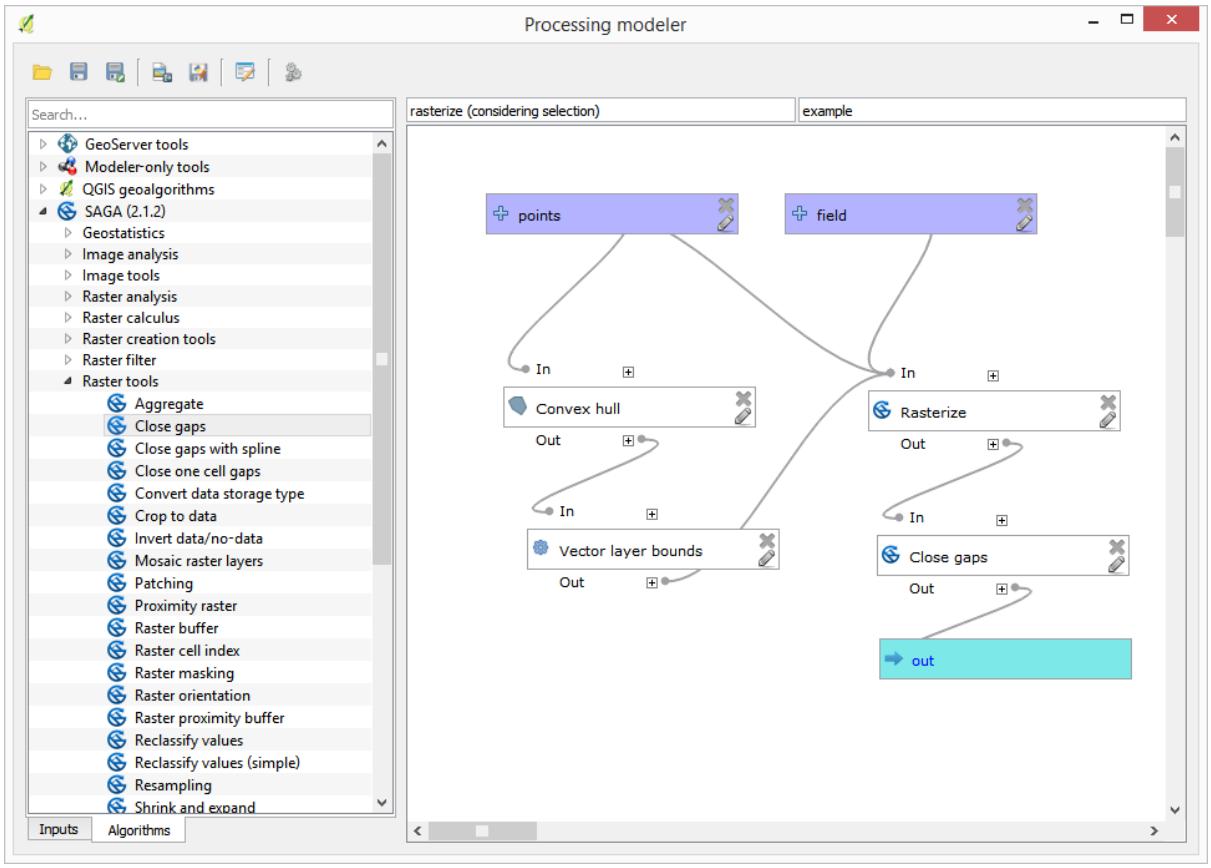
Füllen Sie die Paramter des Algorithmus wie folgt:



Der Arbeitsbereich sollte jetzt so aussehen:



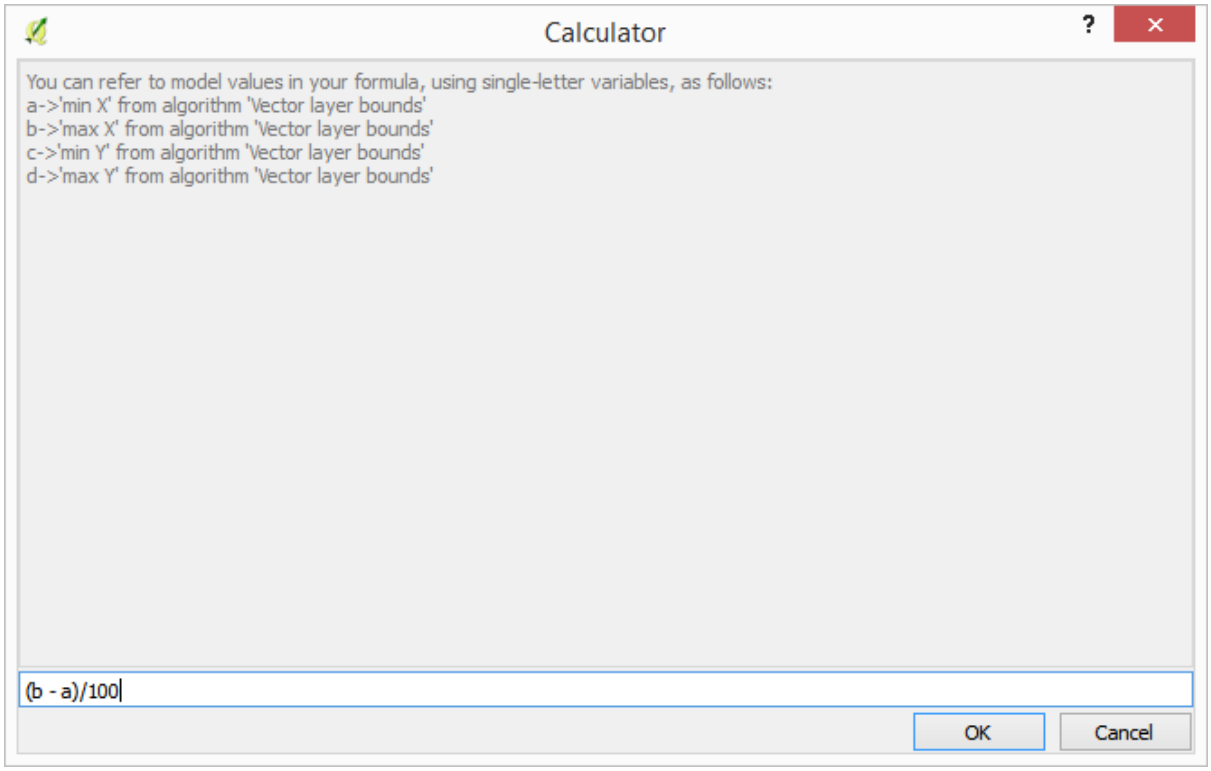
Füllen Sie zum Schluss die Leerwerte des Rasterlayers mit Hilfe des Algorithmus *Close gaps*.



Der Algorithmus kann nun gespeichert werden und ist dann in der Werkzeugkiste verfügbar. Sie können ihn starten und er wird aus der Interpolation der selektierten Punkte des Eingabelayers ein Raster erstellen. Das Rasterlayer wird dieselbe Ausdehung wie die selektierten Punkte im Eingabelayer haben.

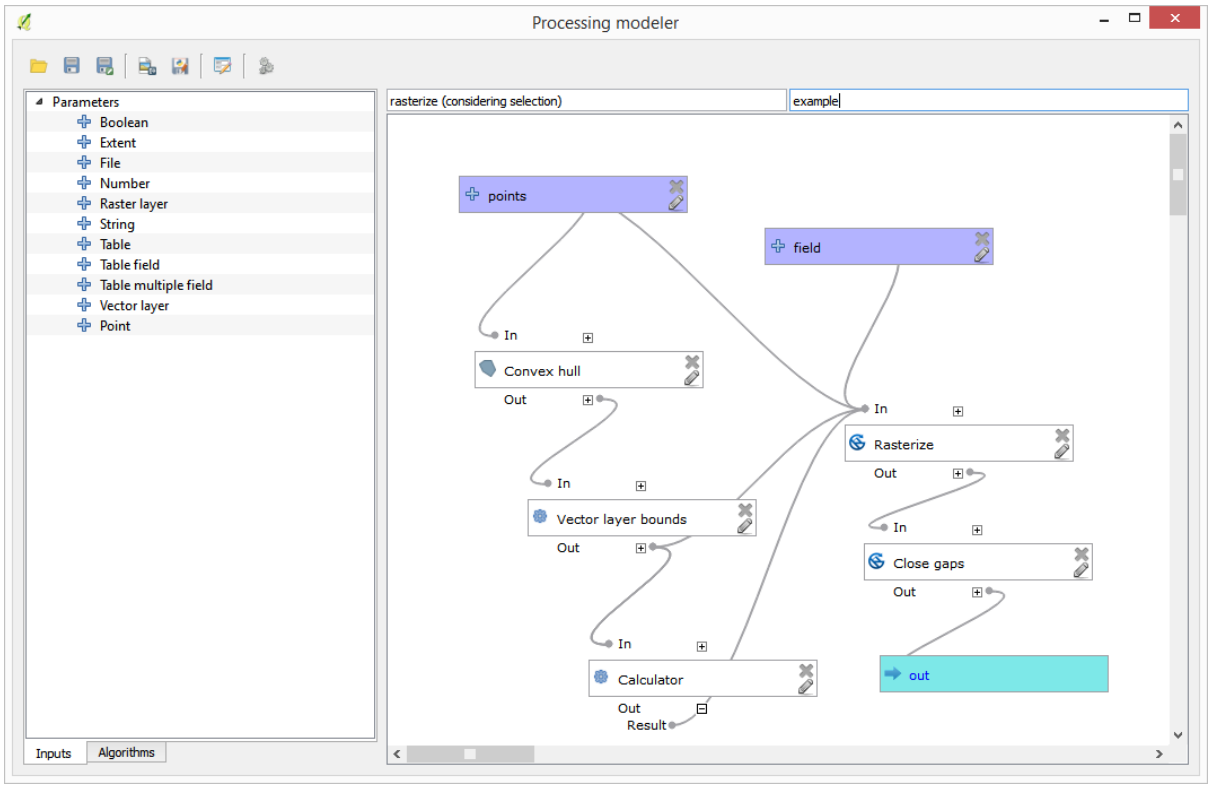
Hier ist eine Verbesserung des Algorithmus. Wir haben einen festen Wert für die Zellengröße beim Rastern verwendet. Der Wert ist für unseren Eingabelayer in Ordnung, kann aber für andere Fälle ungünstig sein. Wir könnten einen neuen Paramter einfügen, so dass der Benutzer den gewünschten Wert eingeben kann. Ein besserer Ansatz ist es, den Wert automatisch berechnen zu lassen.

Wir können den modellinternen Rechner verwenden und den Wert aus den Koordinaten der Ausdehnung berechnen. Um z.B. einen Layer mit einer festen Breite von 100 Pixeln zu erstellen können wir die folgenden Formel im Rechner verwenden.



Wir müssen nun noch den Rasterungsalgorithmus anpassen, so dass er die Ausgabe des Rechners anstelle eines festen Wertes als Eingabe verwendet.

Der fertige Algorithmus sollte wie folgt aussehen:

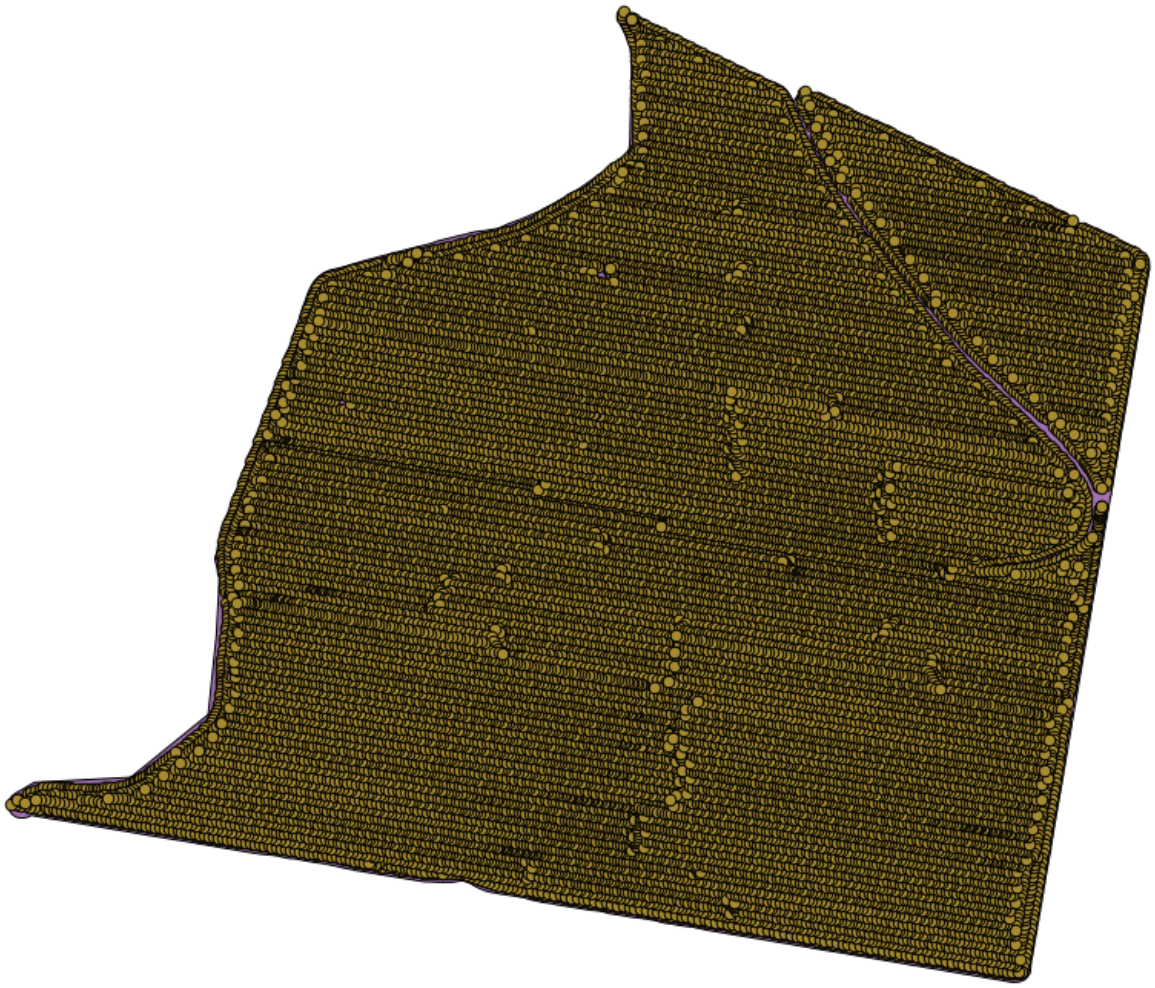


17.22 Interpolation

Bemerkung: In diesem Kapitel lernen wir, wie man Punktdaten interpoliert. Wir werden auch ein weiteres reales Beispiel zur räumlichen Analyse kennenlernen

In dieser Lektion werden wir Punktdaten interpolieren, um einen Rasterlayer zu erhalten. Bevor wir beginnen, müssen wir die Daten vorbereiten. Nach dem Interpolation werden wir den resultierenden Layer noch weiter verarbeiten, so dass sich eine vollständige Analyseroutine ergeben wird.

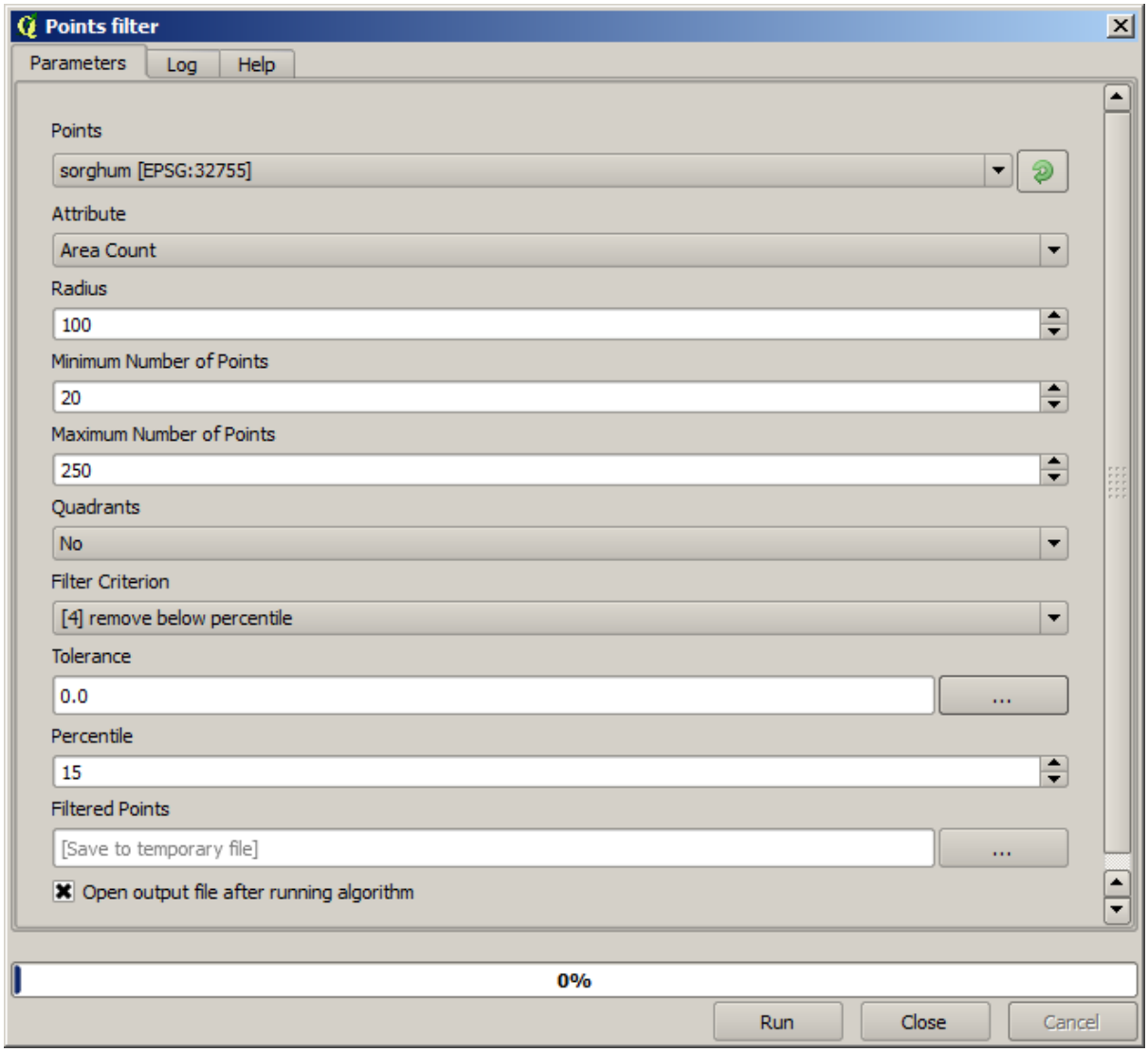
Öffnen Sie die Beispieldaten für diese Lektion, die wie folgt aussehen sollten.



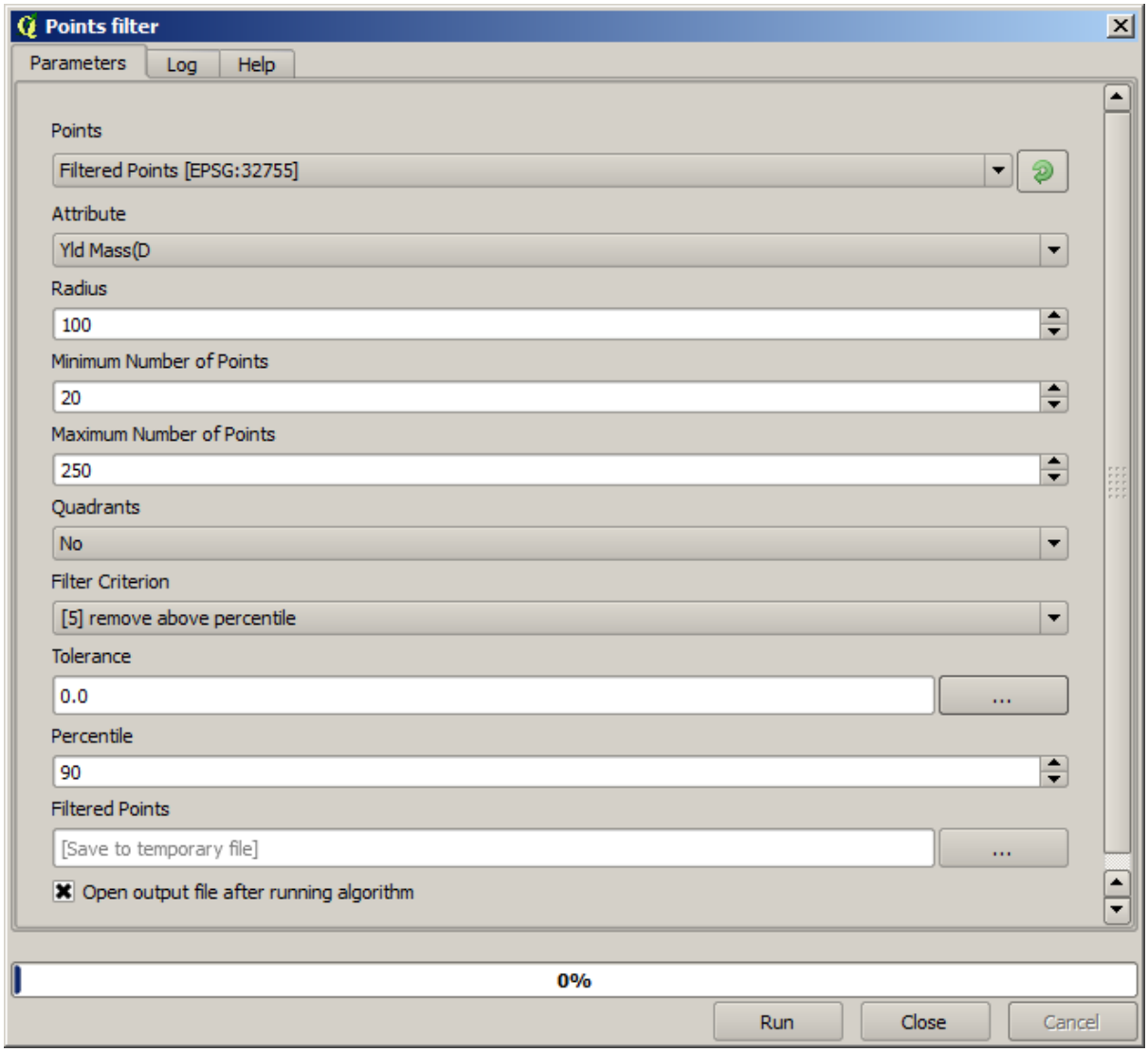
Der Datensatz enthält Daten zum Ernteertrag, so wie er von modernen Erntemaschinen erstellt wird. Wir werden aus diesen Daten einen Rasterlayer des Ernteertrages erstellen. Wir wollen keine weiteren Analysen mit diesem Layer ausführen. Er soll als Hintergrundlayer dienen, der die produktivsten Bereiche und die weniger produktiven Bereiche einfach kennzeichnet.

Als erstes müssen wir den Layer bereinigen, da er redundante Punkte enthält. Sie entstehen durch die Bewegung der Erntemaschine, wenn sie z.B. wendet oder aus bestimmten Gründen ihre Geschwindigkeit verändert. Wir können dafür den Algorithmus *Points filter* benutzen. Wir verwenden ihn zweimal, um Ausreißer sowohl nach oben als auch nach unten zu entfernen.

Verwenden Sie bei der ersten Ausführung die folgenden Parameter.



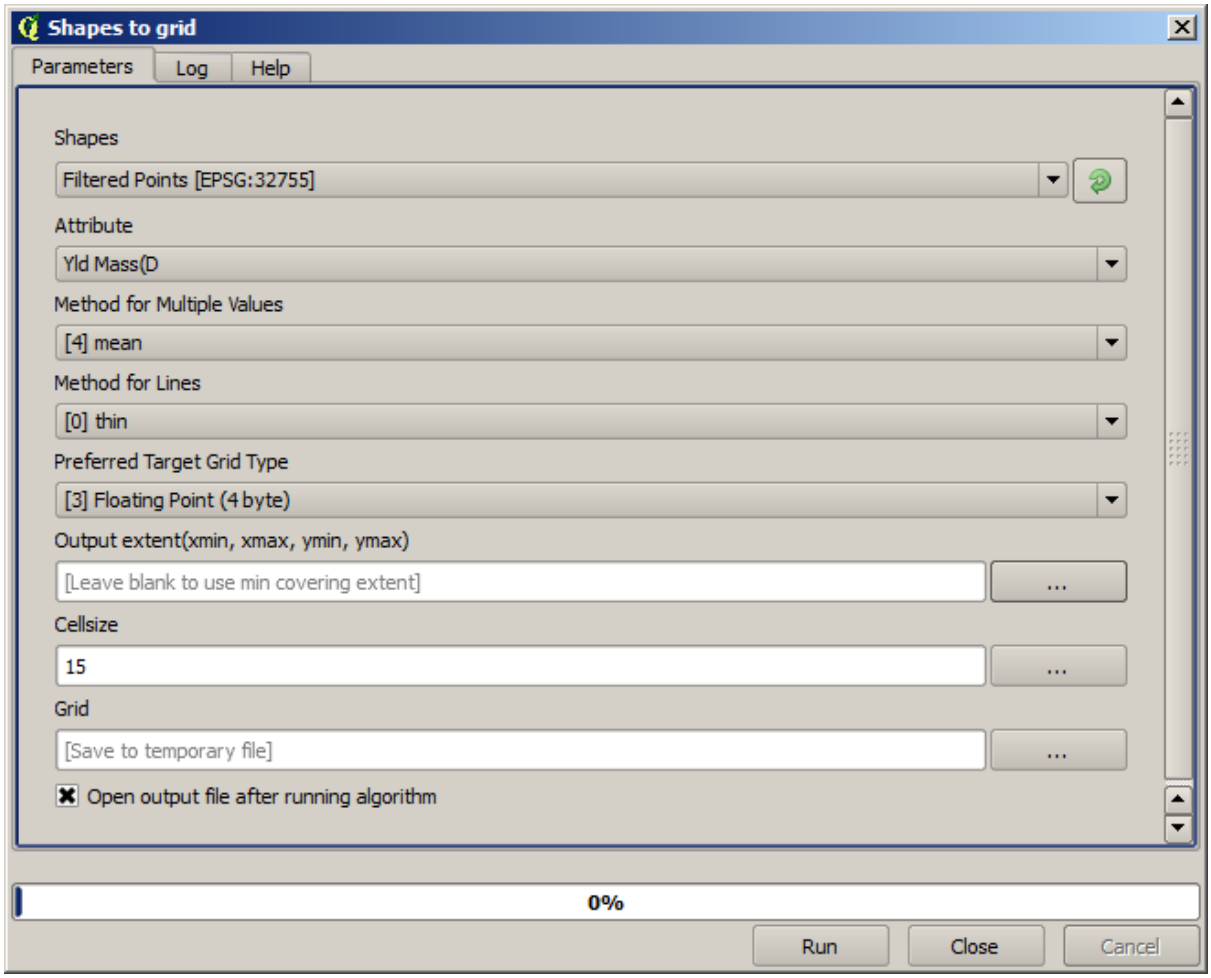
Übernehmen Sie bei der zweiten Ausführung die unten dargestellen Einstellungen.



Beachten Sie, dass wir als Eingabe nicht den Ausgangslayer sondern die Ausgabe des ersten Durchlaufes verwenden.

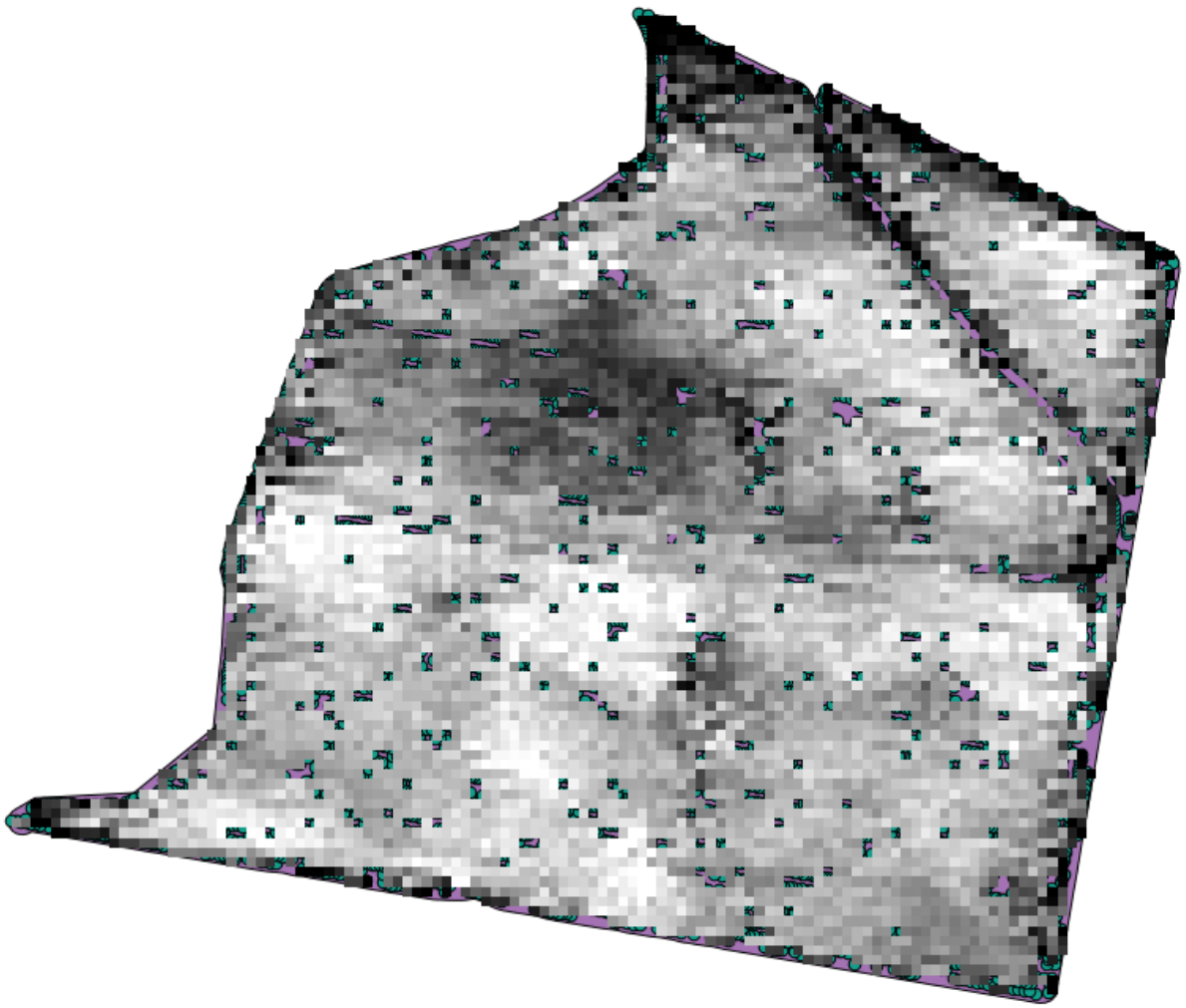
Der Ergebnislayer sollte dem Ausgangslayer ähneln. Er enthält nur weniger Punkte. Wir können das anhand der Attributtabellen überprüfen.

Wir werden den Layer jetzt mit Hilfe des Algorithmus *Rasterize* rastern.

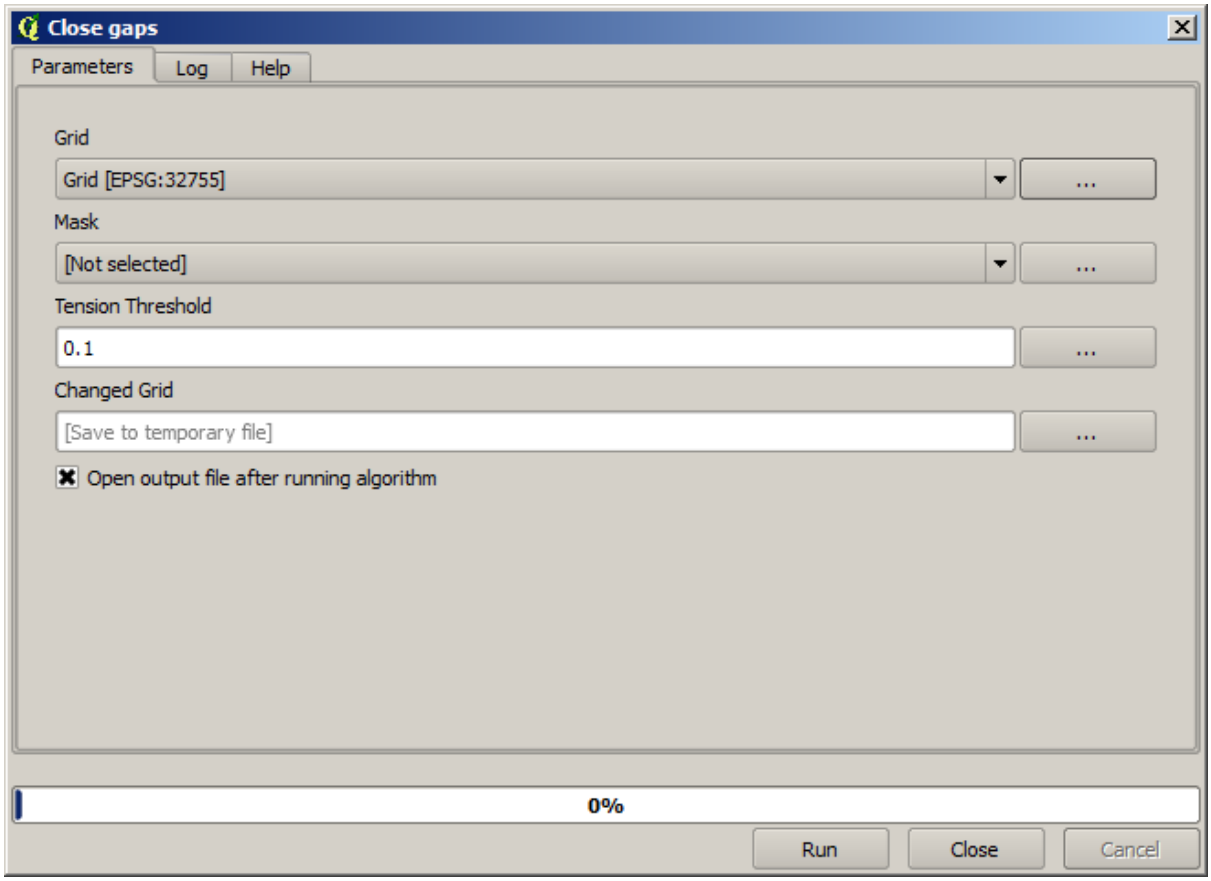


Der Layer *Filtered points* bezieht sich auf das Ergebnis des zweiten Filters. Er hat denselben Namen wie das Ergebnis des ersten Filters, da der Name vom Algorithmus vergeben wird. Sie sollten das erste Ergebnis nicht verwenden. Da wir den Layer nicht mehr benötigen, können wir das Ergebnis des ersten Filters aus dem Projekt löschen und nur das Ergebnis des zweiten Filters behalten. Wir vermeiden damit weitere Verwechselungen.

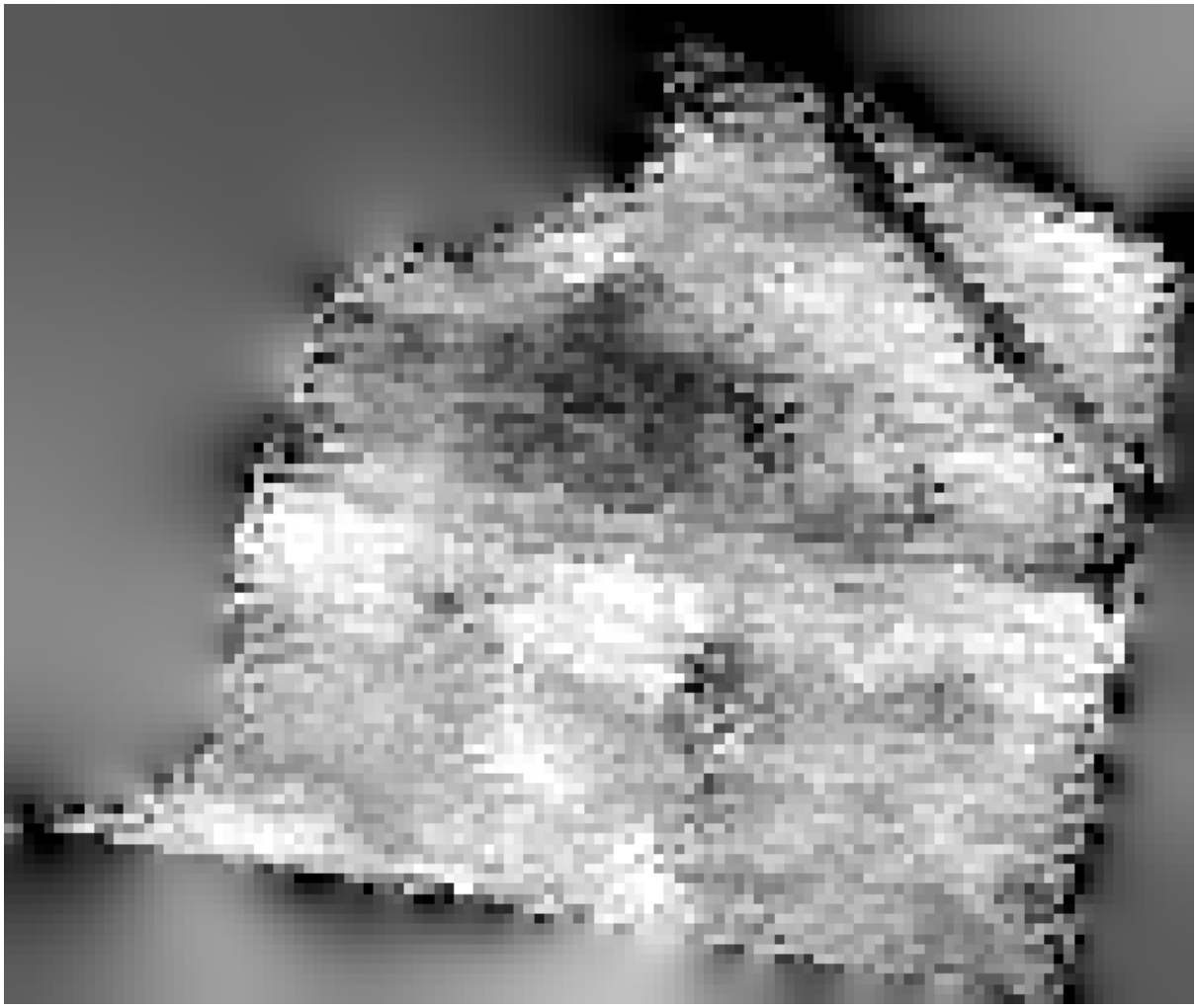
Der sich ergebene Rasterlayer sieht wie folgt aus.



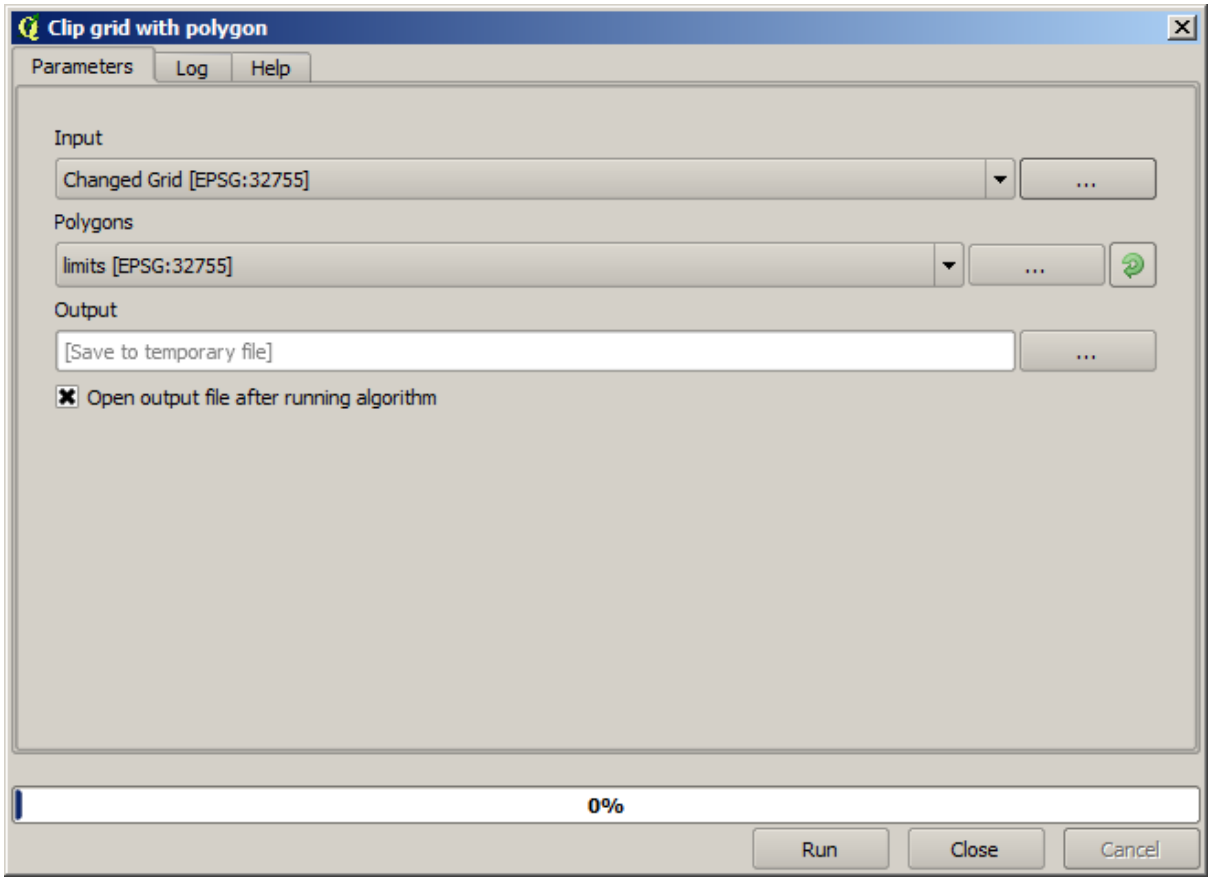
Es handelt sich schon um einen Rasterlayer. Allerdings fehlen die Daten in einigen Zellen. Er enthält nur in den Zellen gültige Werte, in denen ein Punkt aus dem Vektorlayer liegt. Die anderen Zellen enthalten einen Leerwert. Um die fehlenden Werte zu füllen, können wir den Algorithmus *Close gaps* verwenden.



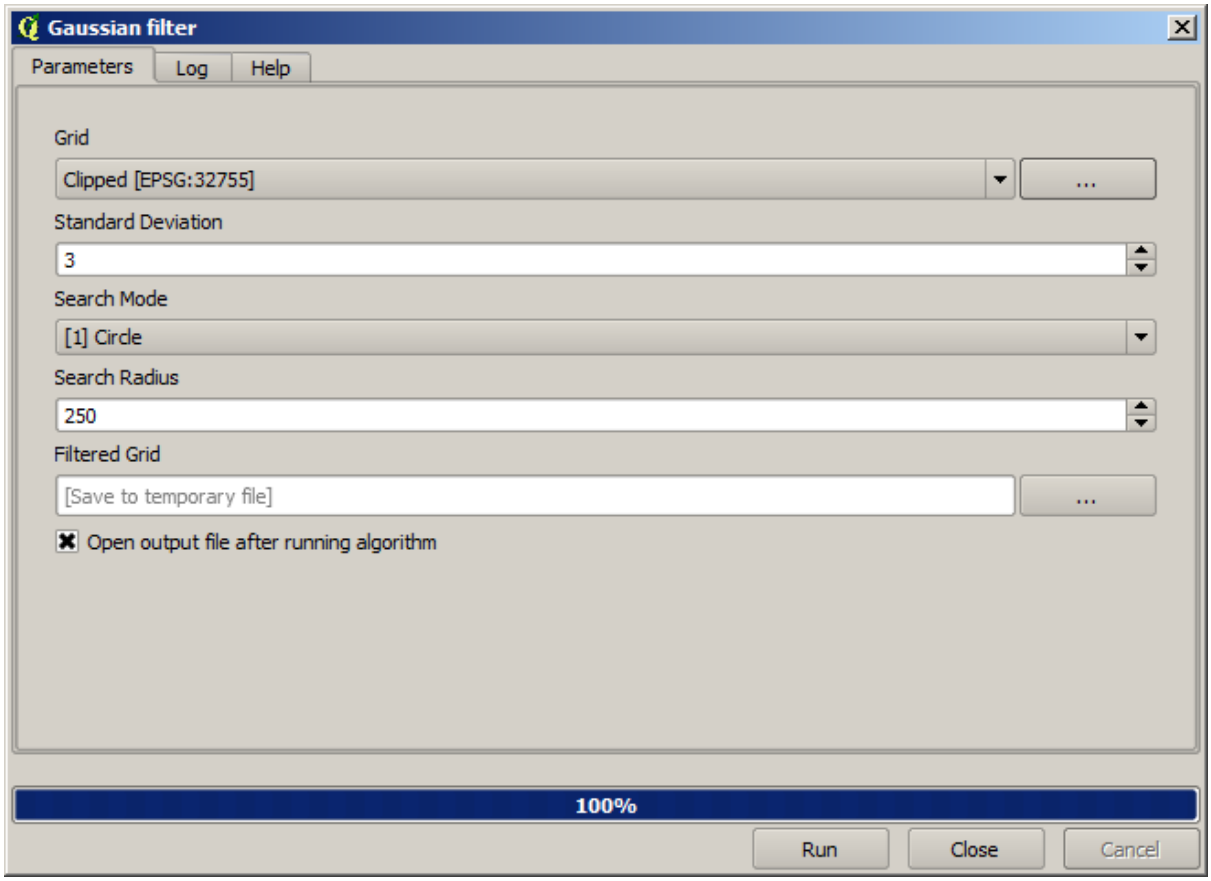
Der Layer ohne Leerwerte sieht wie folgt aus.



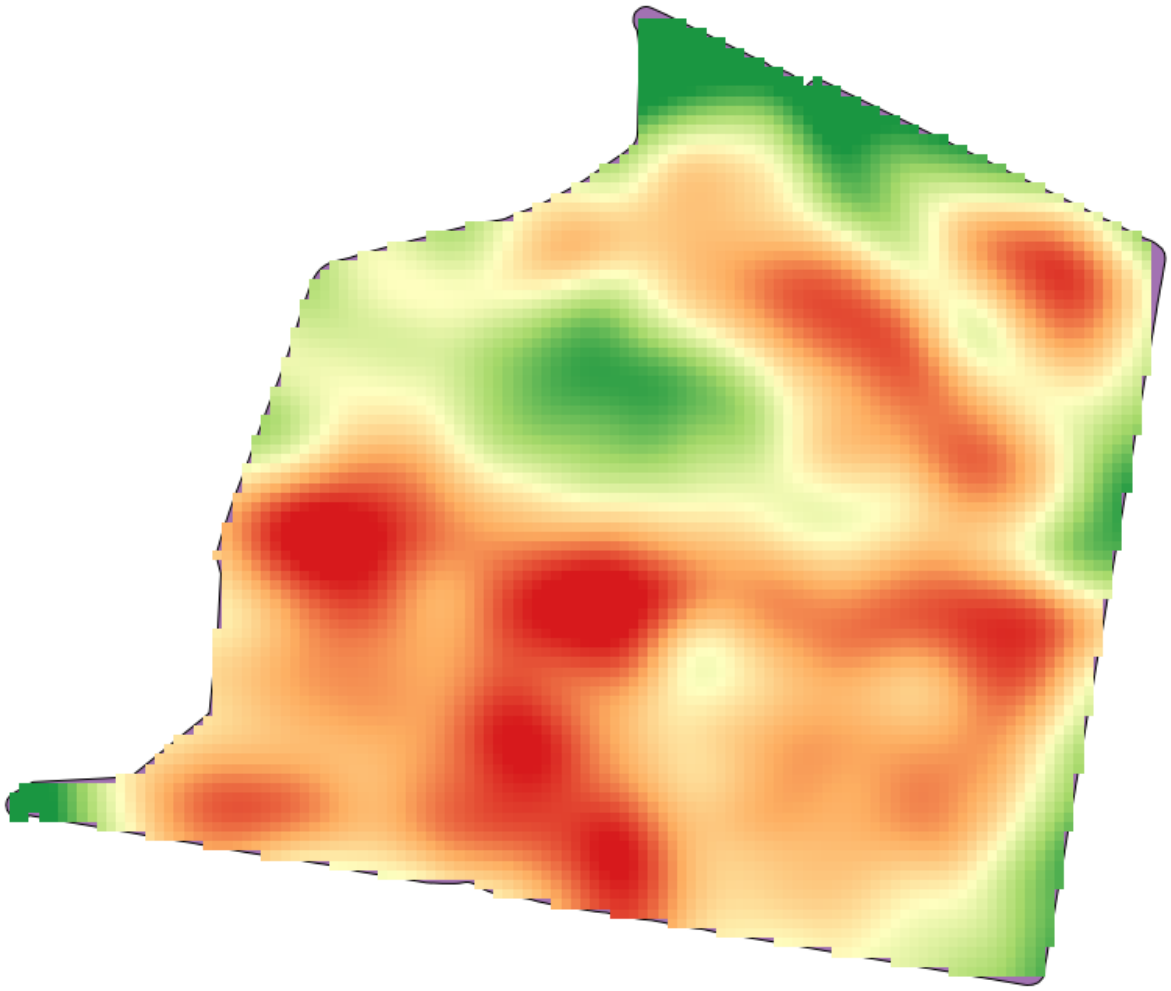
Um die Fläche des Layers genau auf die Region der Messung der Erntemenge zu beschränken, können wir den Layer mit Hilfe des Layer limits zuschneiden.



Um eine geglättetes Ergebnis (weniger akkurat aber besser für die Darstellung als Hintergundlayer geeignet) zu erhalten, können wir den *Gaussian filter* auf den Layer anwenden.



Bei Verwendung der oben aufgeführten Parameter erhalten wir das folgende Ergebnis



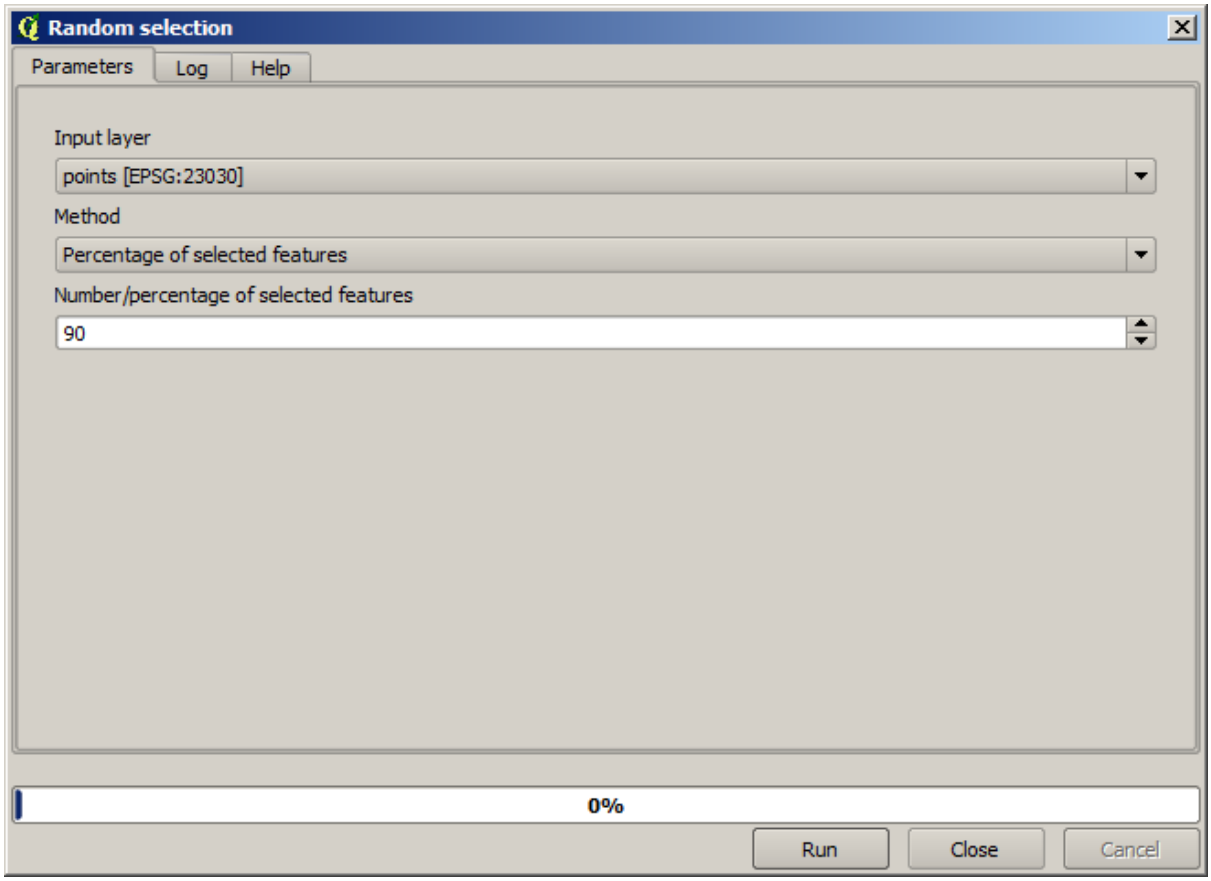
17.23 Noch mehr Interpolation

Bemerkung: Dieses Kapitel behandelt ein weiteres praktisches Beispiel zur Nutzung von Algorithmen zur Interpolation.

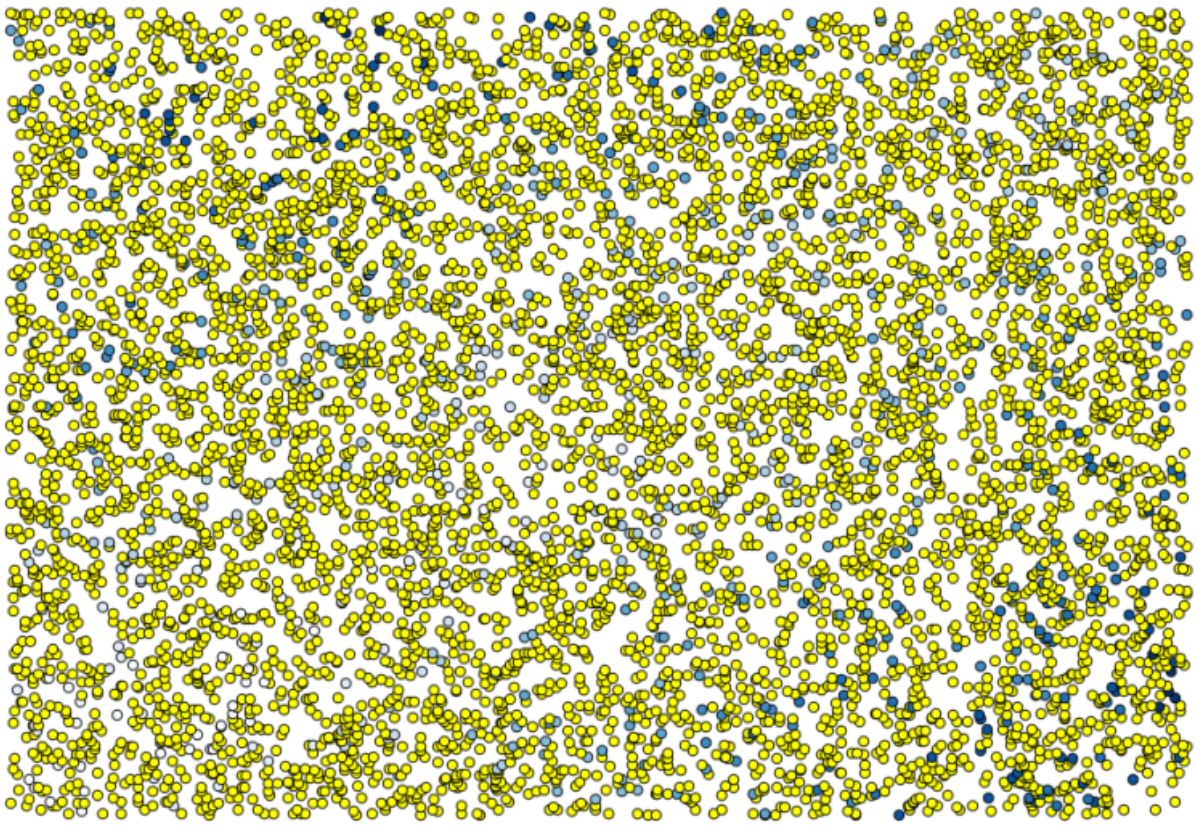
Interpolation ist eine verbreitete Methode. Sie kann auch zur Demonstration verschiedener Techniken in Zusammenhang mit der QGIS Verarbeitungsumgebung verwendet werden. Diese Lektion zeigt einige schon bekannte Algorithmen zur Interpolation, verfolgt aber ein andere Herangehensweise.

Die Daten für diese Lektion enthalten auch Punktdaten, in diesem Fall Höhenwerte. Wir werden die Daten in ähnlicher Weise wie in der vorherigen Lektion interpolieren. Dieses Mal werden wir Teile der Ausgangsdaten speichern und zur Bewertung des Interpolationsprozesses verwenden.

Als Erstes müssen wir den Punktlayer rastern und die sich ergebenen Leerwerte füllen. Wir verwenden dazu aber nicht alle Punkte des Layers. Wir werden 10% der Punkte für die spätere Bewertung verwenden, d.h. wir nutzen nur 90% der Punkte für die Interpolation. Um das zu erreichen, könnten wir den Algorithmus *Split shapes layer randomly* verwenden, den wir aus einer vorherigen Lektion kennen. Aber es gibt noch einen besseren Weg, bei dem wir keinen temporären Layer erstellen müssen. Wir können vor der Ausführung des Algorithmus die gewünschten Punkte selektieren (90% der Punkte). Bei der Rasterung werden dann nur diese Punkte berücksichtigt. Die restlichen Punkte werden ignoriert. Die Selektion kann mit Hilfe des Algorithmus *Zufällige Auswahl* erfolgen. Starten Sie ihn mit den folgenden Parametern.

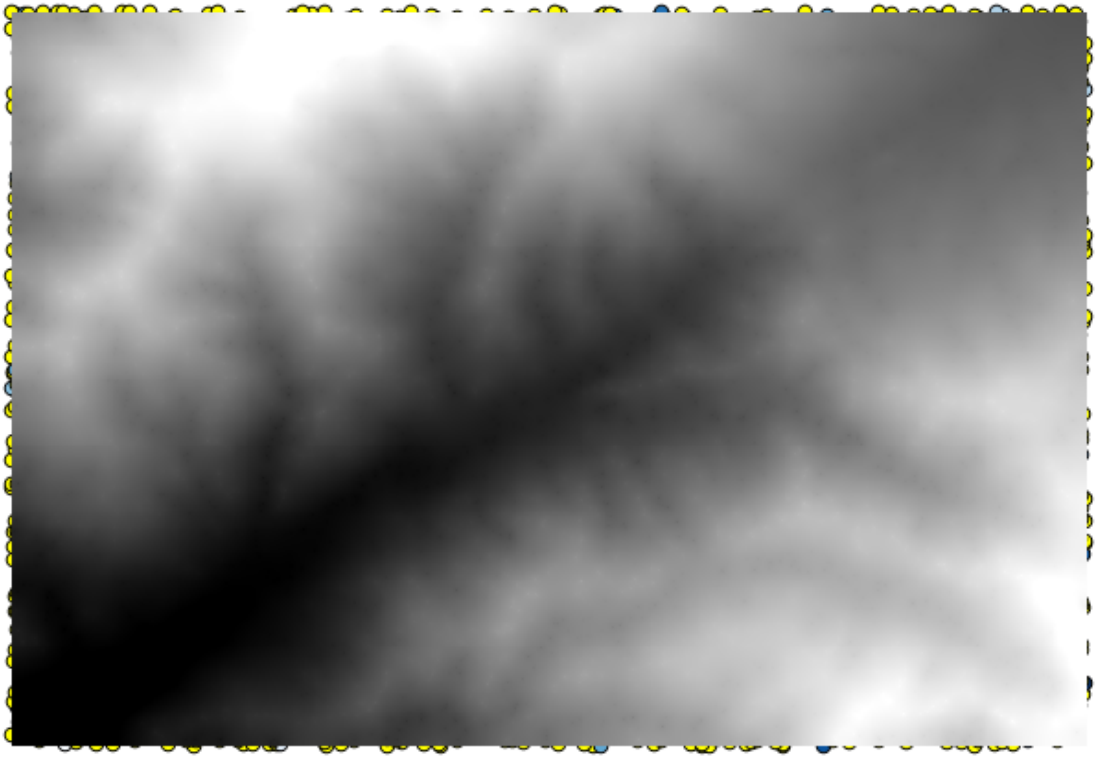


Damit werden 90% der Punkte des Layer für die Rasterung ausgewählt



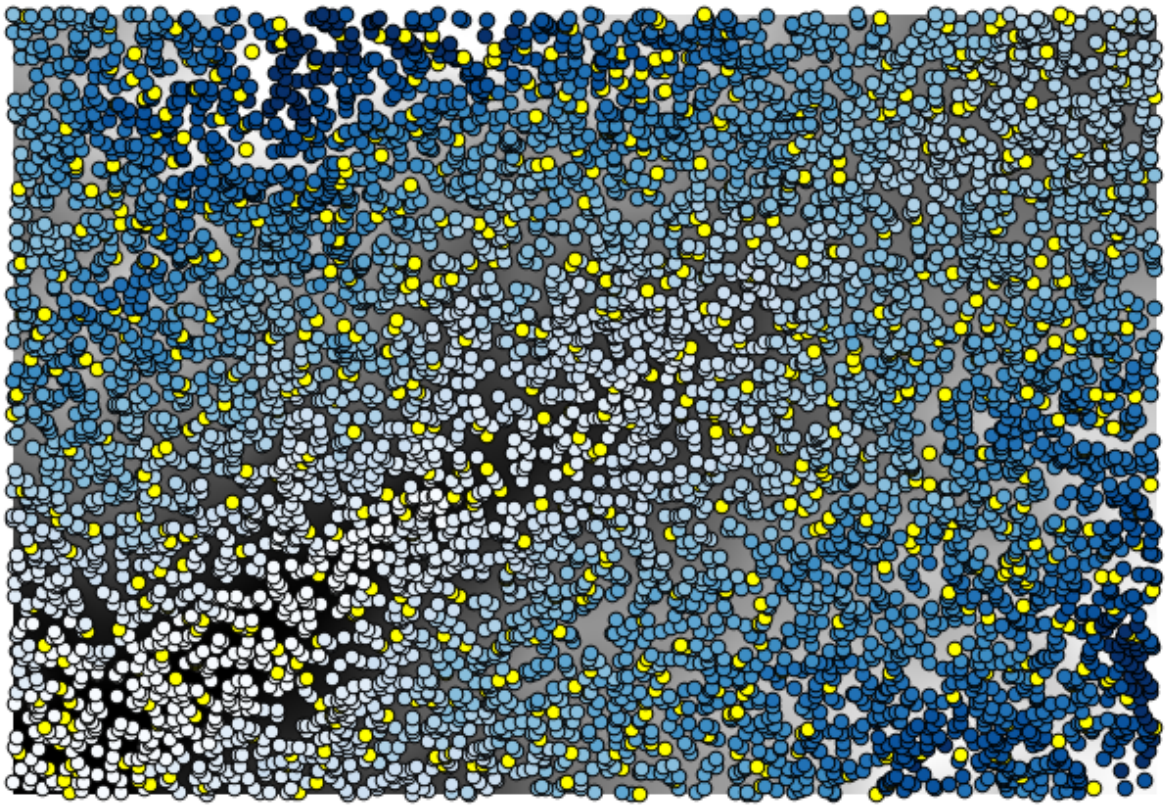
Die Auswahl ist zufällig, so dass Ihre Auswahl vermutlich von der hier gezeigten Auswahl abweicht.

Starten Sie nun den Algorithmus *Rastern*, um den ersten Rasterlayer zu erhalten. Füllen Sie anschließend die Leerwerte mit Hilfe des Algorithmus *Close gaps* [Cell resolution: 100 m].

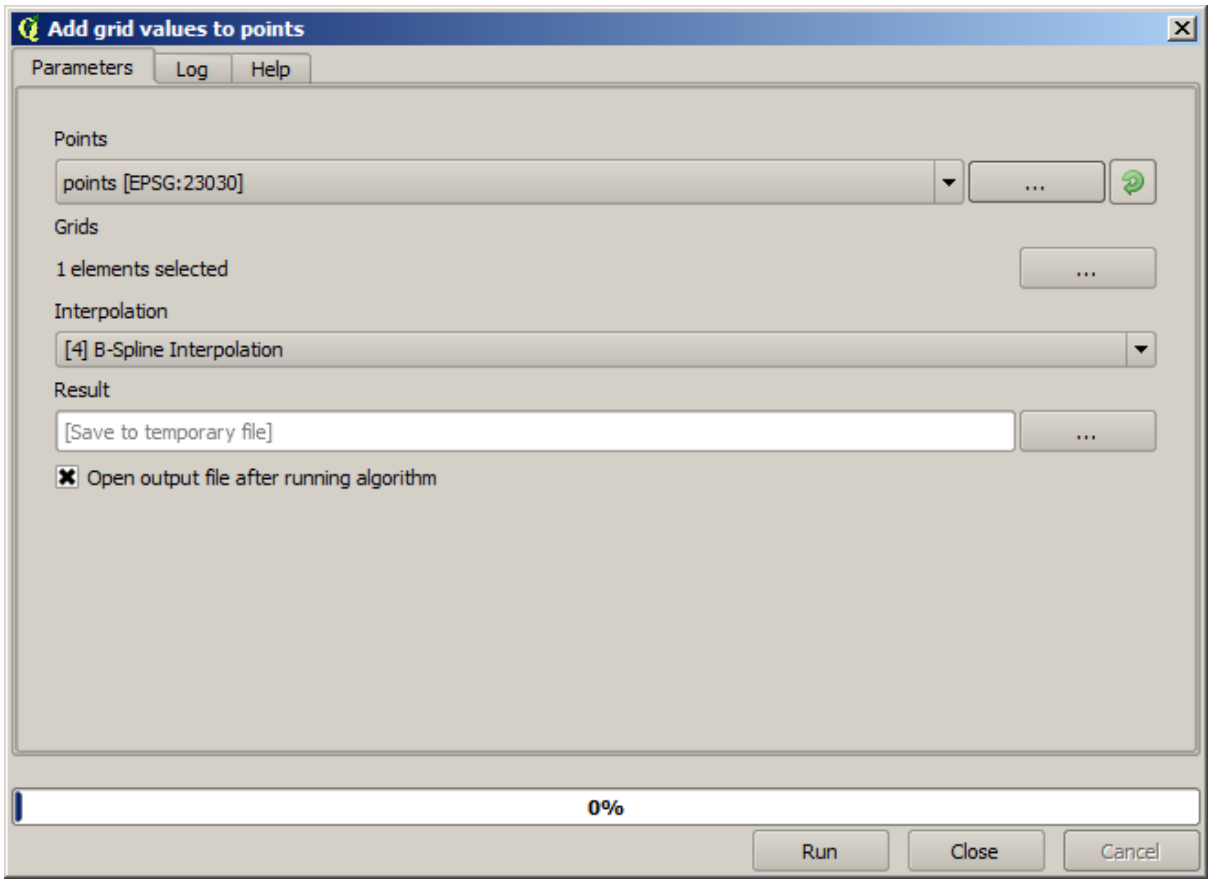


Um die Qualität der Interpolation zu überprüfen, verwenden wir nun die nicht selektierten Punkte. An diesen Punkten kennen wir die tatsächliche Geländehöhe (der Wert im Punktlayer) und die interpolierte Geländehöhe (der Wert im interpolierten Rasterlayer). Wir können die beiden durch Berechnung der Differenz der Werte vergleichen.

Da wir die nicht selektierten Punkte nutzen möchten, müssen wir als Erstes die Selektion umkehren.

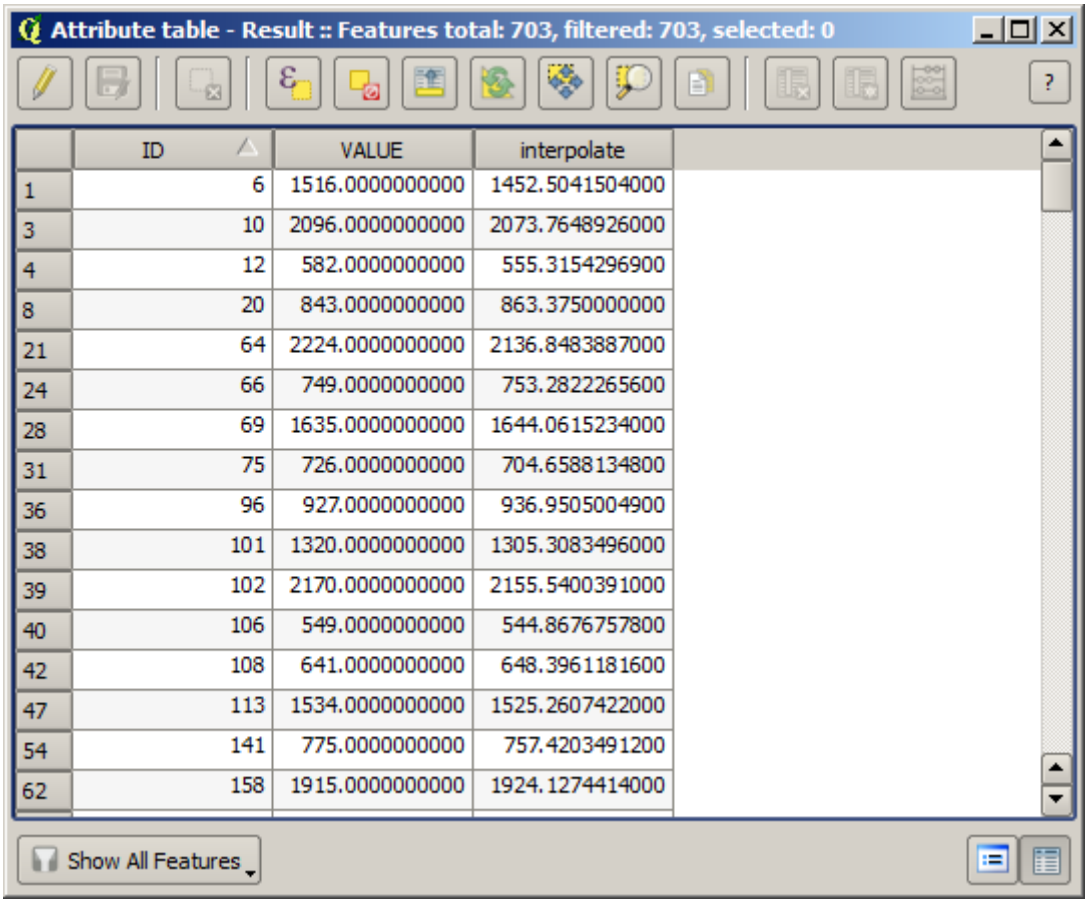


Die Punkte enthalten die Ausgangswerte aber nicht die interpolierten. Um sie in einem neuen Feld hinzuzufügen, können wir den Algorithmus *Add Raster values to points* verwenden.



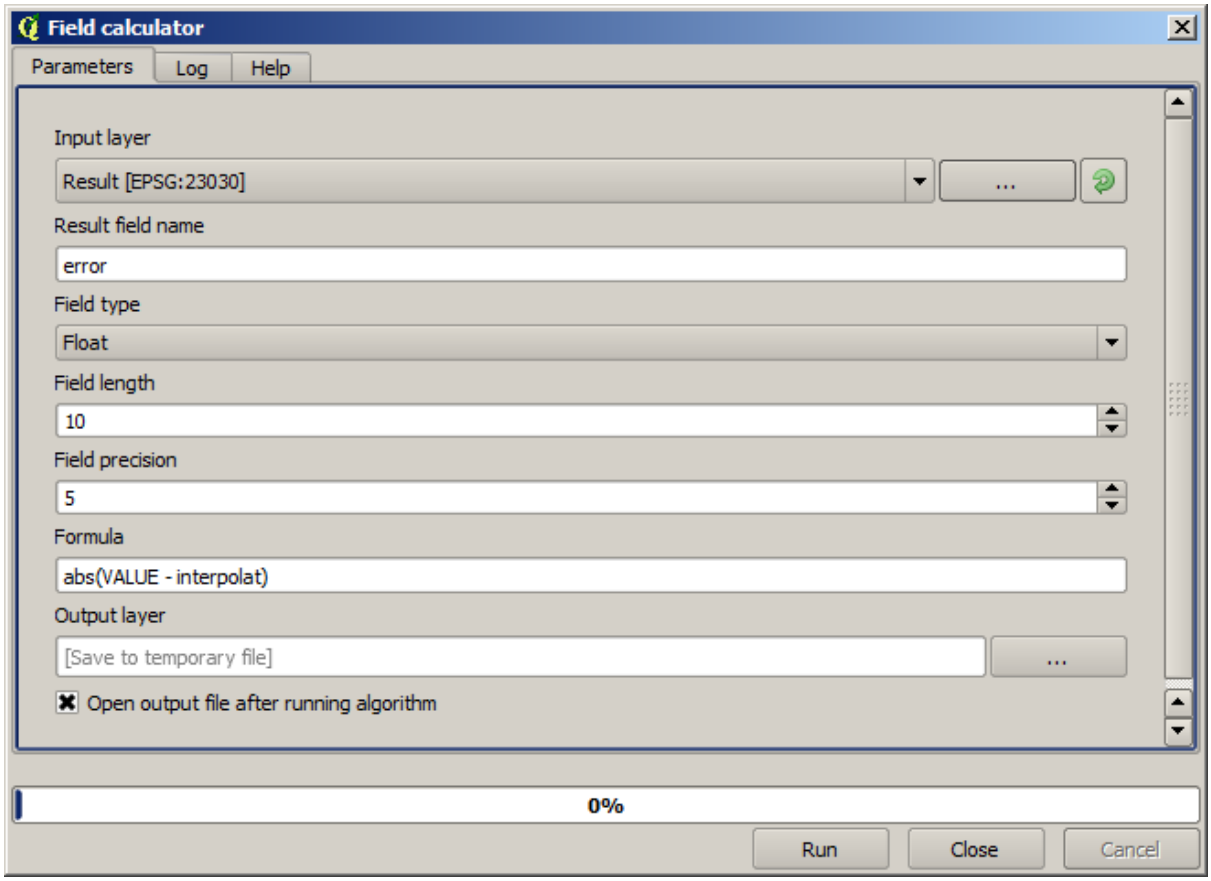
Der zu wählende Rasterlayer (der Algorithmus unterstützt mehrere Raster, wir benötigen nur eins) ist das Ergebnis der Interpolation. Wir haben den Layer in *interpolate* umbenannt. Dieser Layernamen wird auch als Name für das hinzugefügte Feld verwendet.

Wir haben nun einen Vektorlayer, der für Punkte die nicht für die Interpolation verwendet wurden, beide Werte enthält.



	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

Wir werden für die nächste Aufgabe den Feldrechner verwenden. Öffnen Sie den Algorithmus *Feldrechner* und starten ihn mit den folgenden Parametern.



Wenn Ihr Feld mit den Werten aus dem Rasterlayer einen anderen Namen hat, müssen Sie die Formel entsprechend anpassen. Nach dem Ausführen des Algorithmus erhalten Sie einen Layer der nicht für die Interpolation verwendeten Punkte mit der Differenz zwischen den zwei Höhenwerten.

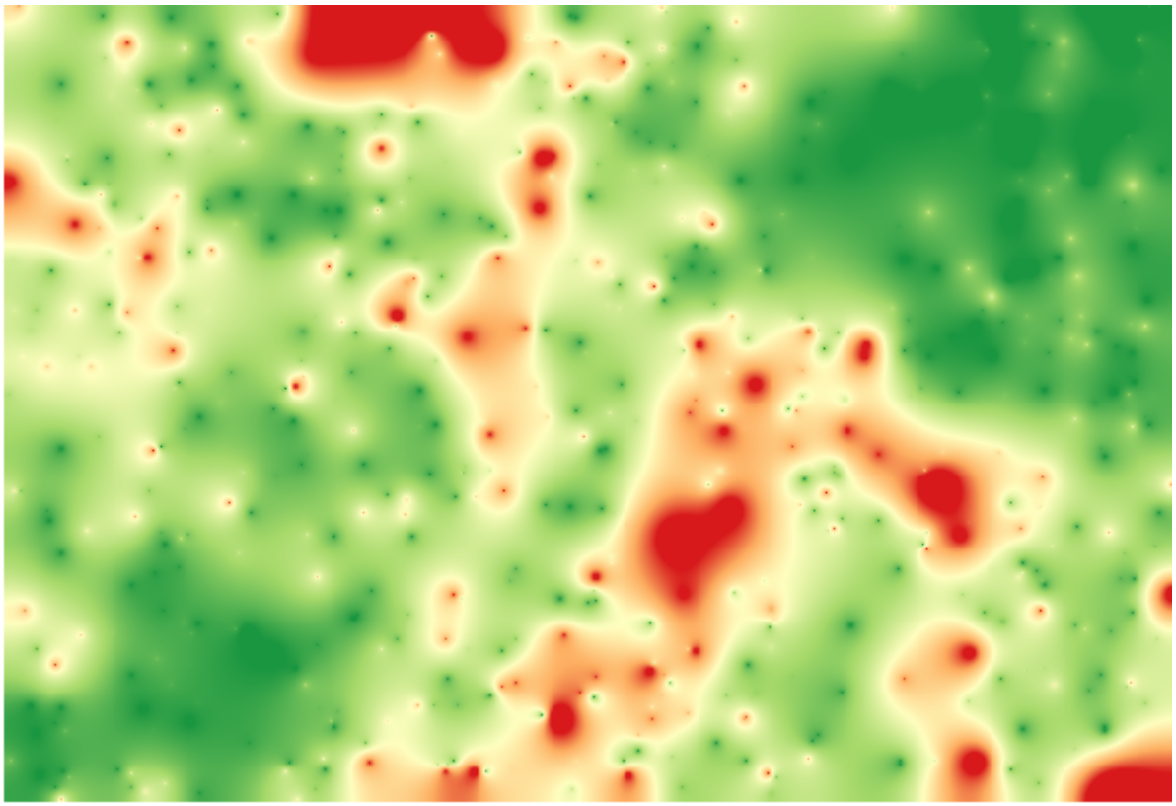
Die Darstellung des Layer entsprechend dieses Wertes gibt uns einen ersten Hinweis, wo die größten Abweichungen liegen.

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

Die Interpolation dieses Layers liefert einen Rasterlayer mit dem zu erwartenden Fehler für alle Punkte in der interpolierten Fläche.



Sie können dieselbe Information (Differenz zwischen den Ausgangspunktdaten und den interpolierten Werten) auch direkt über *GRASS* ► *v.sample* erhalten.

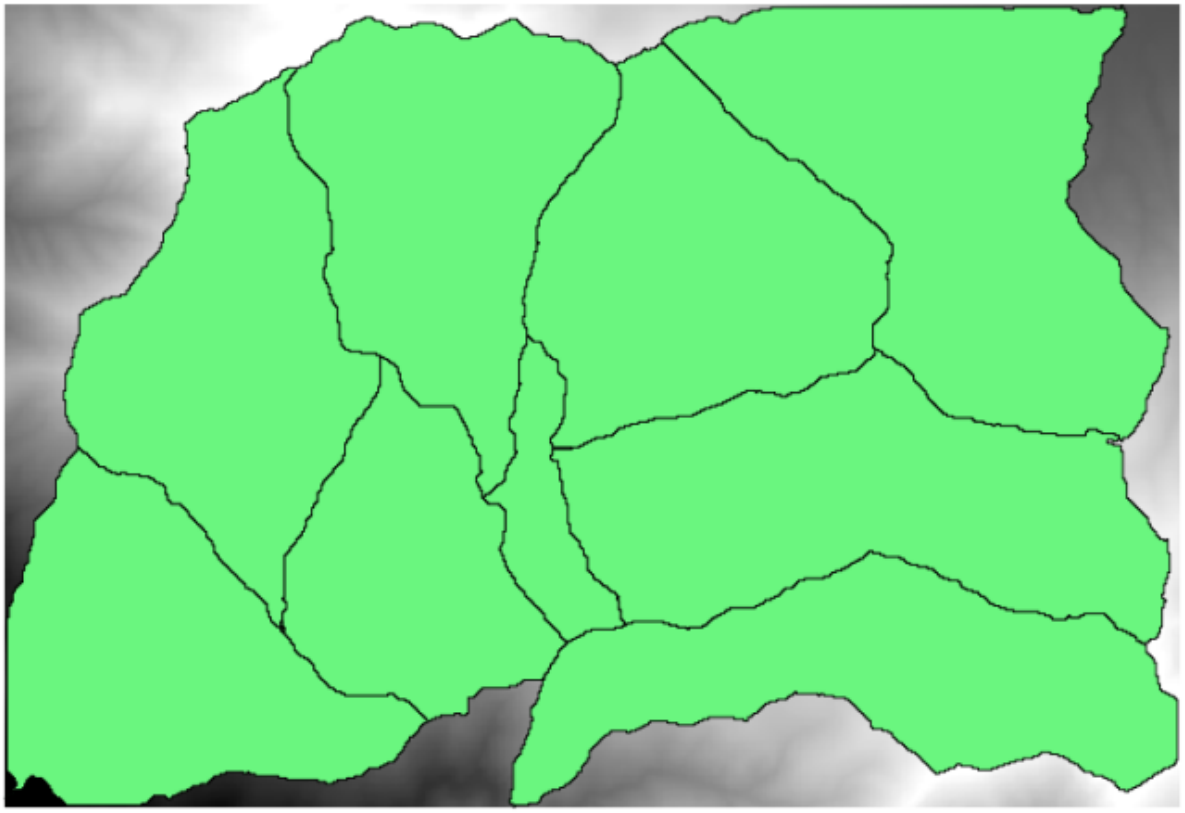
Ihre Resultate weichen unter Umständen von den hier ermittelten ab. Dies liegt an der Zufallskomponente, die wir mit der anfänglichen zufälligen Auswahl eingeführt hatten.

17.24 Die iterative Ausführung von Algorithmen

Bemerkung: Diese Lektion zeigt einen anderen Weg, Algorithmen für Vektorlayer wiederholt zu starten. Dabei wird über Objekte des Eingabevektorlayers iteriert.

We already know the model designer, 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.

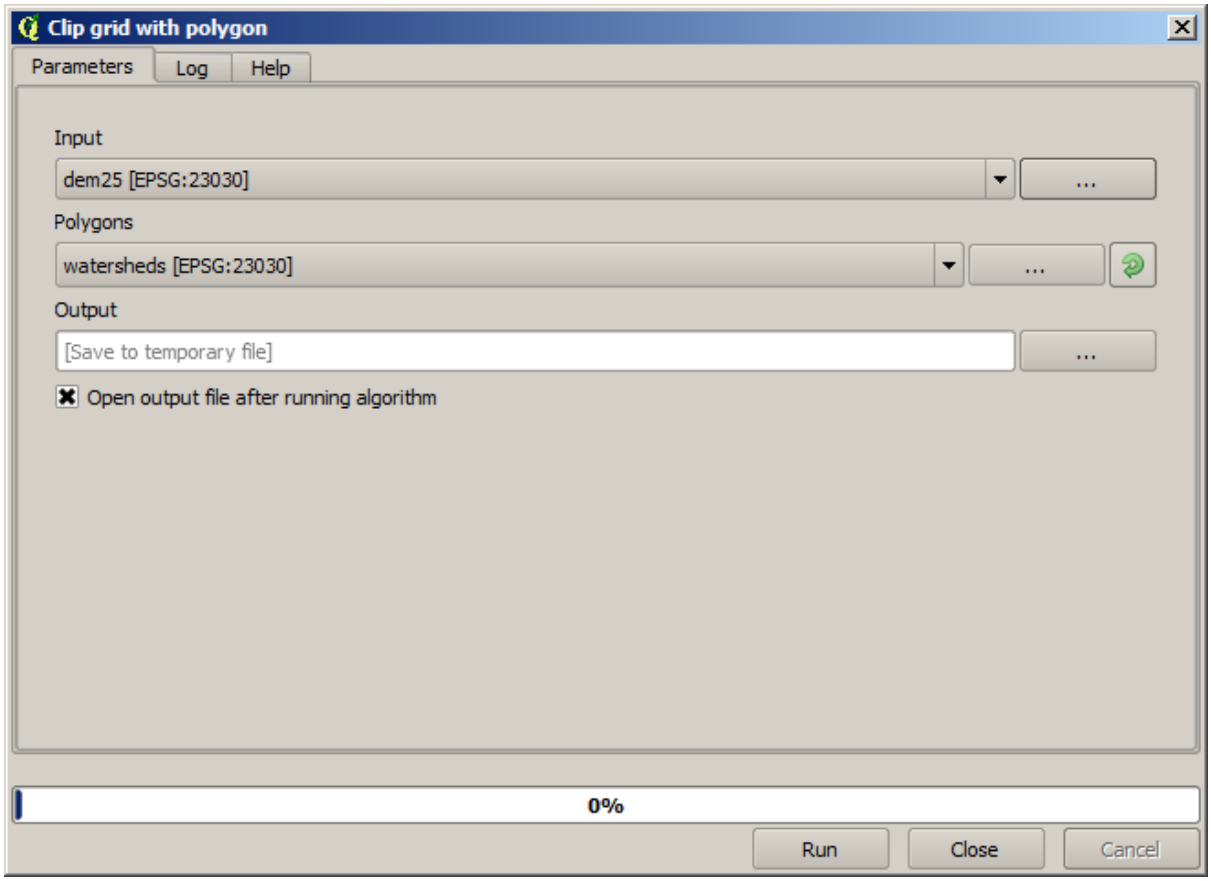
Öffnen Sie die zu diesem Kapitel gehörenden Daten. Sie sollten in etwa wie folgt aussehen.



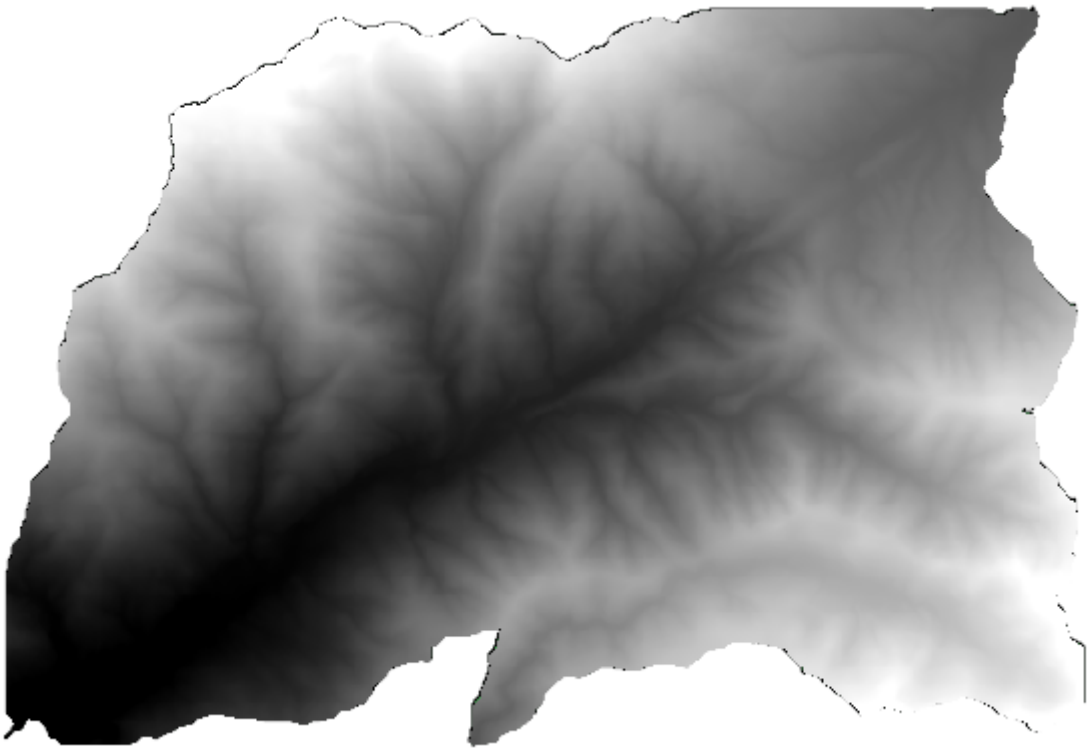
Wir erkennen unser wohlbekanntes DEM aus früheren Lektionen und einige daraus extrahierte Wasserscheiden. Stellen Sie sich vor, Sie müssen das DEM in mehrere kleinere Layer zerteilen, die jeweils das zu einer Wasserscheide gehörende Gebiet enthalten. Das ist nützlich, wenn man später für jedes Einzugsgebiet bestimmte Parameter wie z.B. die mittlere Höhe oder die hypsografische Kurve bestimmen möchte.

Das kann ein langwieriger und ermüdender Prozess sein, besonders wenn die Anzahl der Wasserscheiden hoch ist. Die Aufgabe kann aber einfach automatisert werden, wie wir gleich sehen.

Der Algorithmus, der zum Zuschneiden eines Rasterlayers mit einem Polygonlayer verwendet wird, nennt sich *Clip raster with polygons*. Er beinhaltet den folgenden Parameter Dialog.

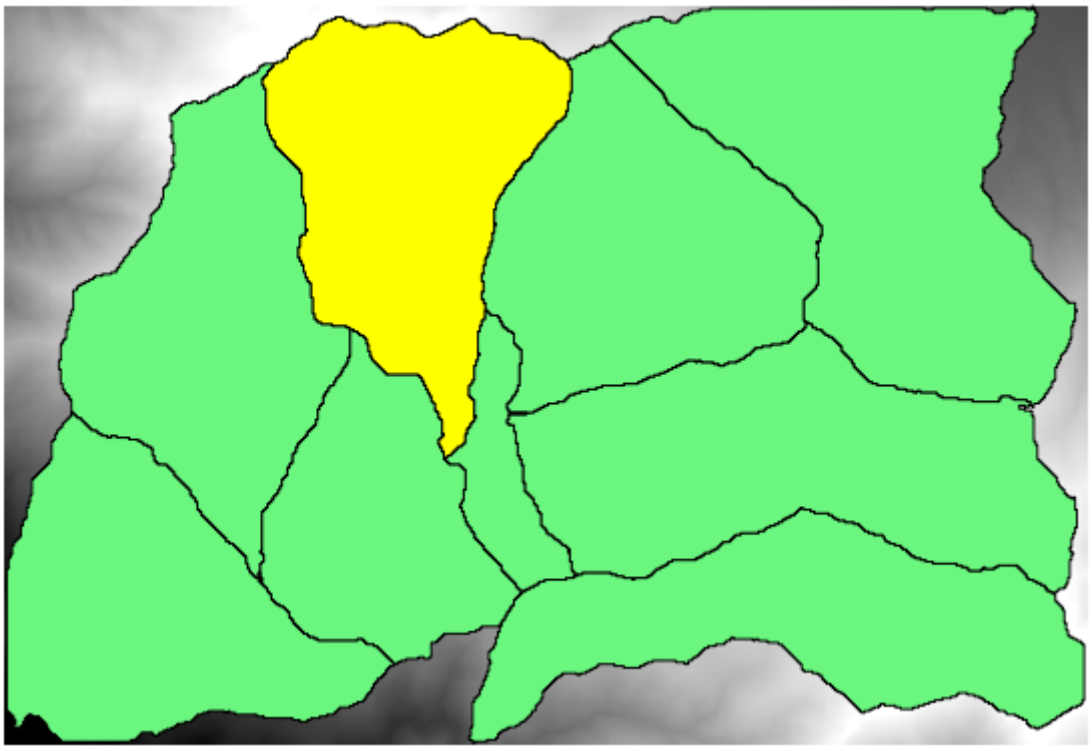


Wenn Sie ihn mit dem Layer der Wasserscheiden und dem DEM als Eingabe starten, erhalten Sie das folgende Ergebnis.

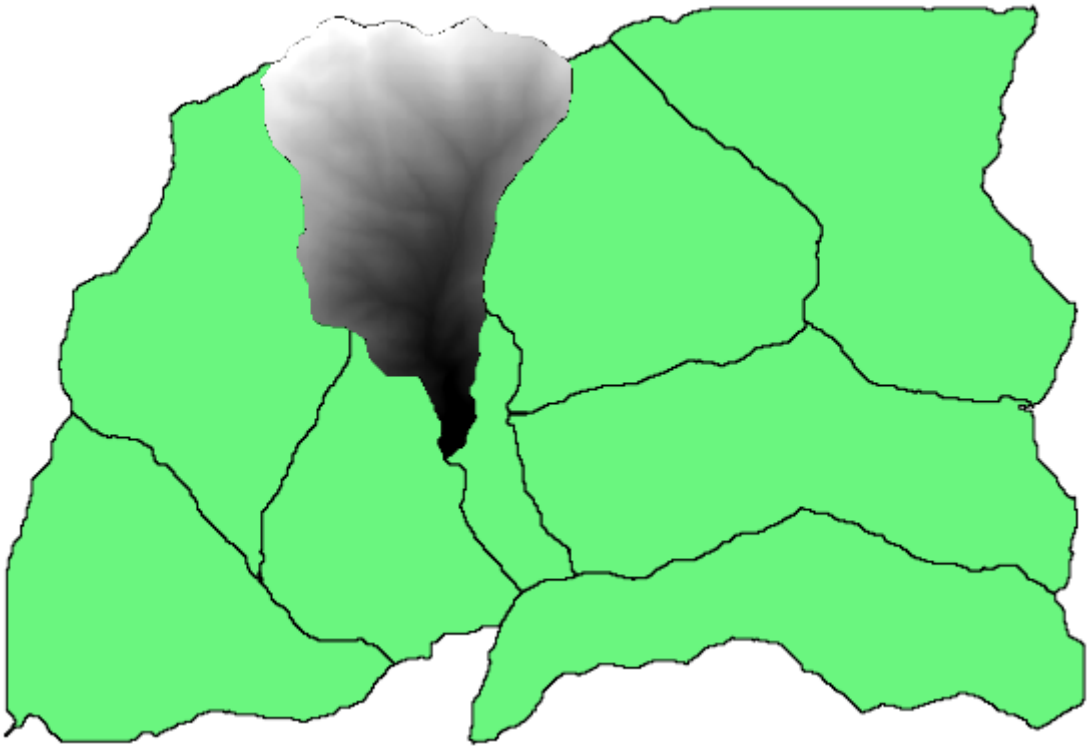


Wir sehen, dass die Fläche aller Polygone von Wasserscheiden für den Zuschnitt verwendet wird.

Man hätte auch das Polygon einer Wasserscheide auswählen und anschließend den Algorithmus zum Zuschneiden starten können.

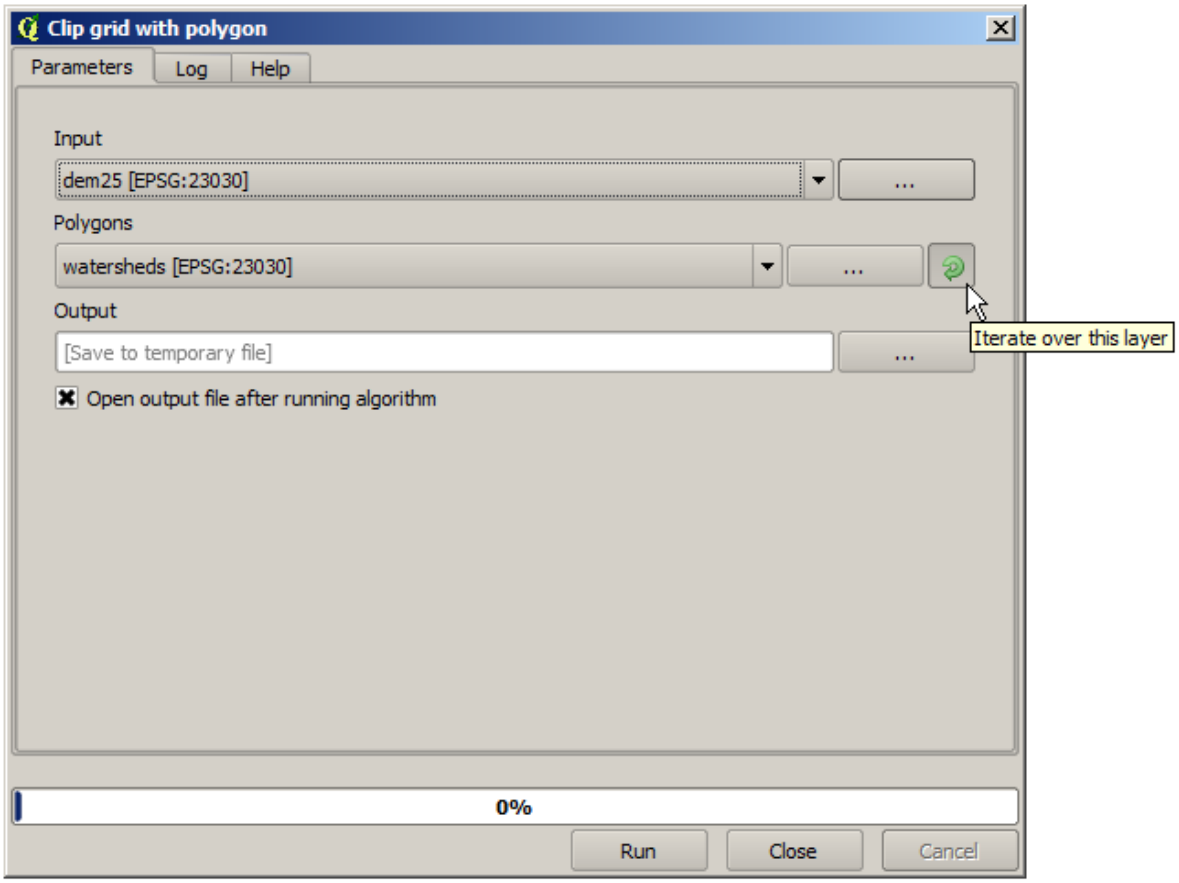


Da nur gewählte Objekte verwendet werden, wird nur das ausgewählte Polygon zum Zuschneiden des Rasterlayers genutzt.



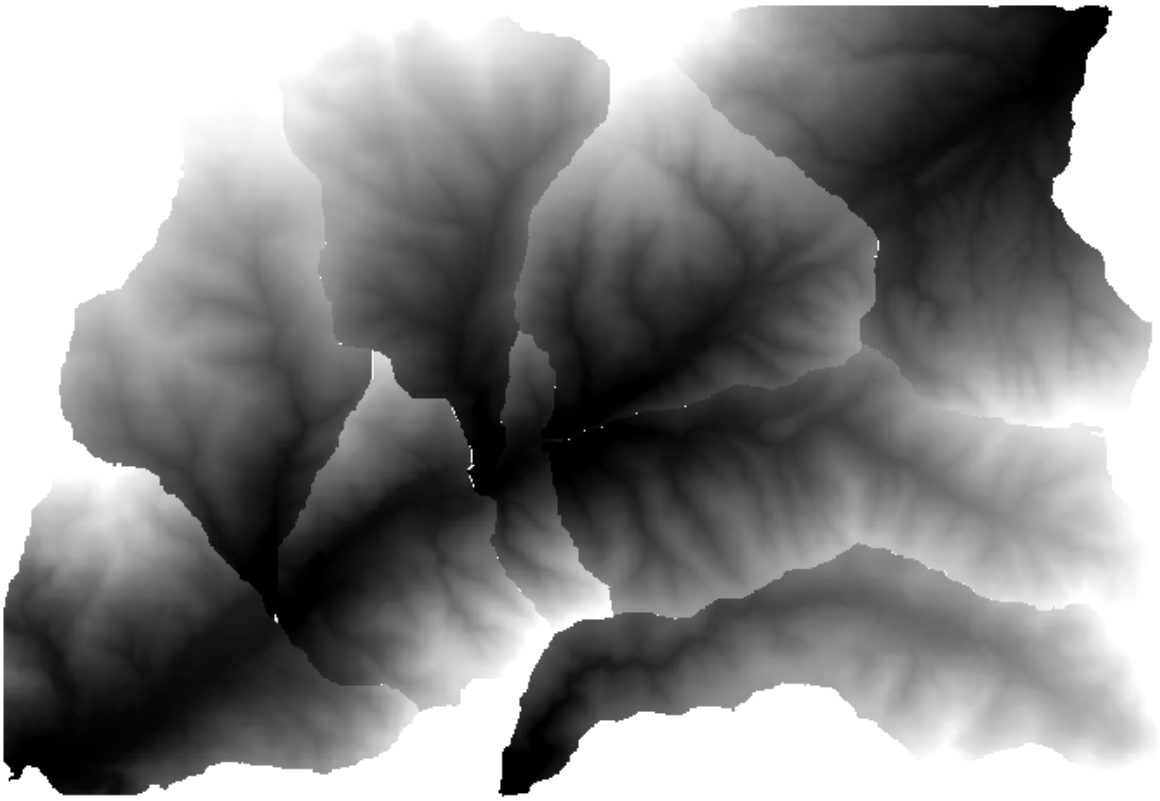
Wenn man so für alle Polygone vorgeht, erhält man das gewünschte Ergebnis. Allerdings ist das kein sehr praktikables Vorgehen. Wir werden stattdessen sehen, wie man die *select and crop* Routine automatisiert.

Entfernen Sie als Erstes die vorherige Auswahl, so dass wieder alle Polygone genutzt werden. Öffnen Sie nun den Algorithmus *Clip raster with polygon* und wählen dieselben Eingabewerte wie vorher. Klicken Sie dann auf die Schaltfläche auf der rechten Seite des Eingabefeldes in dem der Vektorlayer ausgewählt wurde.



Die Schaltfläche bewirkt, dass der gewählte Eingabelayer sooft geteilt wird, wie er Objekte enthält. Der Algorithmus wird dann für jedes enthaltene Objekt wiederholt aufgerufen. Als Ergebnis erhält man nicht einen Rasterlayer sondern eine Menge an Rasterlayern. Die Anzahl der Rasterlayer entspricht der Anzahl der Aufrufe des Algorithmus.

Hier ist das Ergebnis, dass Sie erhalten wenn Sie den Algorithmus wie beschrieben ausführen.



Die Palette zwischen Weiß und Schwarz (oder auch jede andere verwendete Palette) wird für jeden Layer anders eingestellt, abhängig von den enthaltenen Minima und Maxima. Das ist der Grund, warum man die verschiedenen Teile sieht und die Farben an der Grenze der Layer nicht ineinander übergehen. Die Werte passen trotzdem zueinander.

Wenn Sie einen Ausgabedateinamen angeben, werden die ausgegebenen Dateien mit diesem Namen und einem Suffix für die jeweilige Iteration benannt.

17.25 Noch mehr iterative Ausführung von Algortihmen

Bemerkung: In dieser Lektion lernen wir, wie man die Ausführung mehrerer iterativer Algorithmen mit Hilfe der grafischen Modellierung kombiniert.

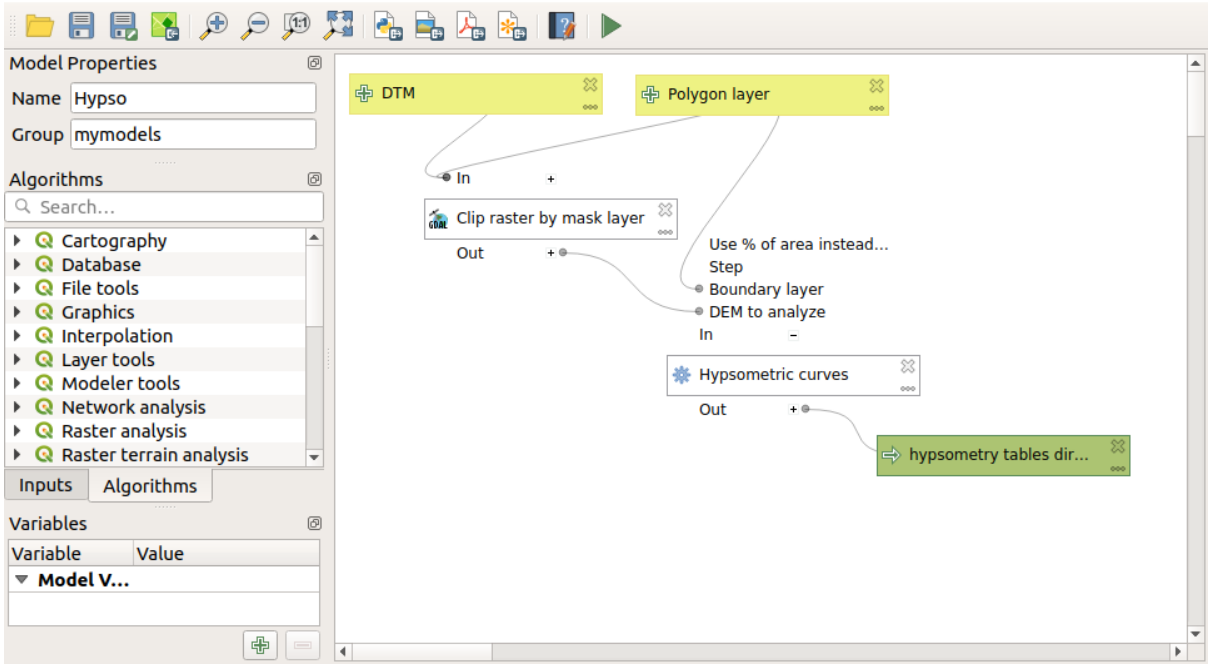
Die iterative Ausführung von Algorithmen ist nicht nur für die integrierten Algorithmen verfügbar sondern auch für selbst erstellte Algorithmen wie z.B. Modelle. Wir werden lernen, wie man ein Modell und die interative Ausführung von Algorithmen miteinander kombiniert. Auf diese Art und Weise erhält man leicht auch komplexe Ergebnisse.

Die Daten, die wir in dieser Lektion verwenden sind dieselben wie in der letzten Lektion. Diesmal werden wir nicht nur das DEM auf die Einzugsgebiete zuschneiden, sondern zusätzlich für jedes Gebiet eine hypsografische Kurve erstellen und die Verteilung der Höhenwerte auswerten.

Da wir einen Arbeitsablauf mit mehreren Schritten verwenden (zuschneiden + berechnen der hypsografischen Kurve), öffnen wir die grafische Modellierung und erstellen ein Modell.

Sie finden das fertige Modell im Datenordner zu dieser Lektion. Es wäre aber gut, wenn Sie zuerst selbst versuchen, das Modell zu erstellen. Der zugeschnittene Layer ist hier kein endgültiges Ergebnis, da wir nur an den Kurven interessiert sind. Das Modell wir daher keine Layer erstellen sondern nur eine Tabelle mit den Daten der Kurven.

Das Modell sollte in etwa so aussehen:

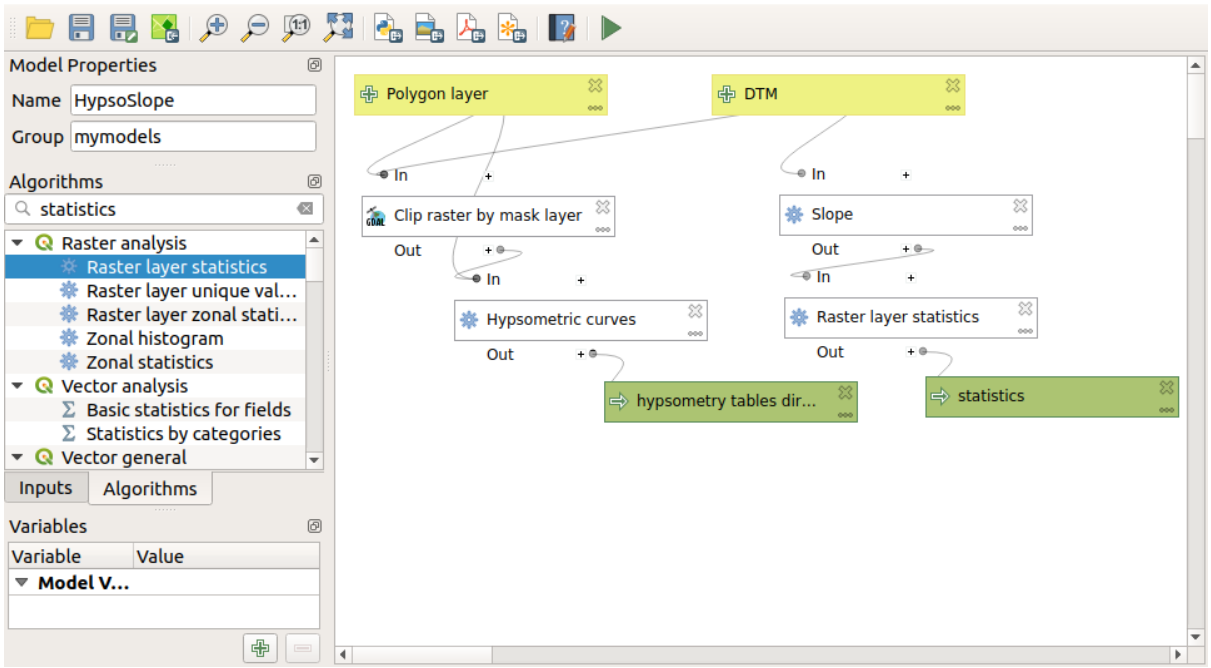


Fügen Sie das Modell Ihrem Modellordner hinzu, damit es in der Toolbox verfügbar ist, und führen Sie es aus.

Wählen Sie die DEM- und Wasserscheidenbecken aus.

Der Algorithmus generiert Tabellen für alle Becken und platziert sie im Ausgabeverzeichnis.

Wir können dieses Beispiel komplexer machen, indem wir das Modell erweitern und einige Steigungsstatistiken berechnen. Fügen Sie dem Modell den *Slope*-Algorithmus und dann den *Raster-Statistiken*-Algorithmus hinzu, der die Steigungsausgabe als einzige Eingabe verwenden sollte.



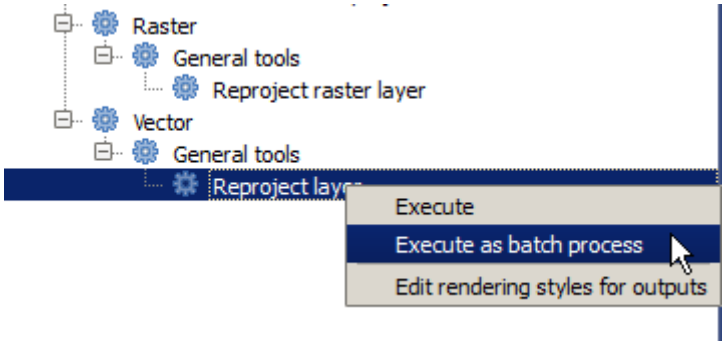
Wenn Sie das Modell nun starten, erhalten Sie außer den Tabellen einige Seiten mit Statistiken. Die Seiten sind in der Ergebnisanzeige verfügbar.

17.26 Die Schnittstelle zur Stapelverarbeitung

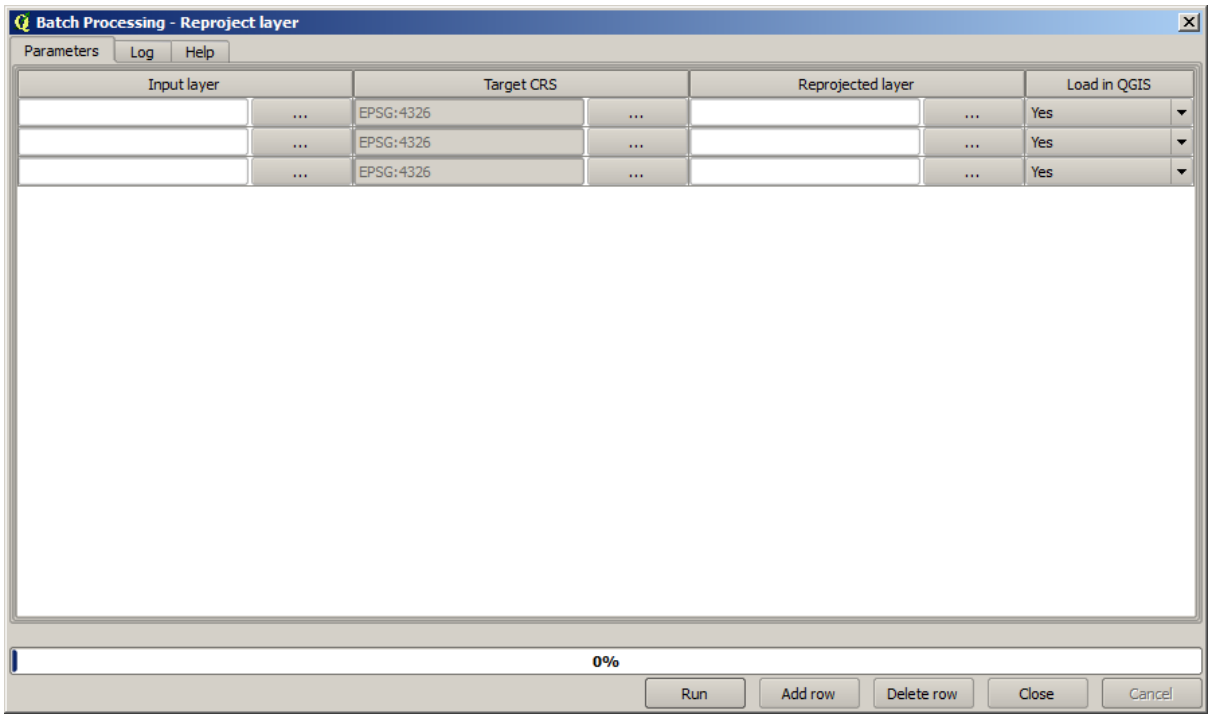
Bemerkung: In dieser Lektion lernen wir die Schnittstelle zur Stapelverarbeitung kennen. Mit ihrer Hilfe können wir einen einzelnen Algorithmus mit mehreren Eingabewerten starten.

Manchmal ist es erforderlich, einen Algorithmus mehrmals mit verschiedenen Eingaben auszuführen. Das ist z.B. der Fall, wenn mehrere Eingabedateien von einem Format in ein anderes konvertiert werden sollen oder wenn Dateien von einer Projektion in eine andere umgewandelt werden müssen.

In diesem Fall ist der wiederholte Aufruf des Algorithmus aus der Werkzeugkiste nicht der optimale Weg. Stattdessen kann man die Stapelverarbeitung verwenden, die die wiederholte Ausführung eines Algorithmus stark vereinfacht. Um einen Algortihmus aus der Werkzeugkiste im Rahmen der Stapelverarbeitung zu starten, klickt man nicht wie sonst doppelt auf den Algorithmus sonder mit der rechten Maustaste und wählt dann *Als Stapelprozess ausführen…*.



In diesem Beispiel verwenden wir den Algorithmus *Reprojizieren*. Finden Sie ihn und gehen Sie dann wie beschrieben vor. Wir sehen nun den folgenden Dialog.

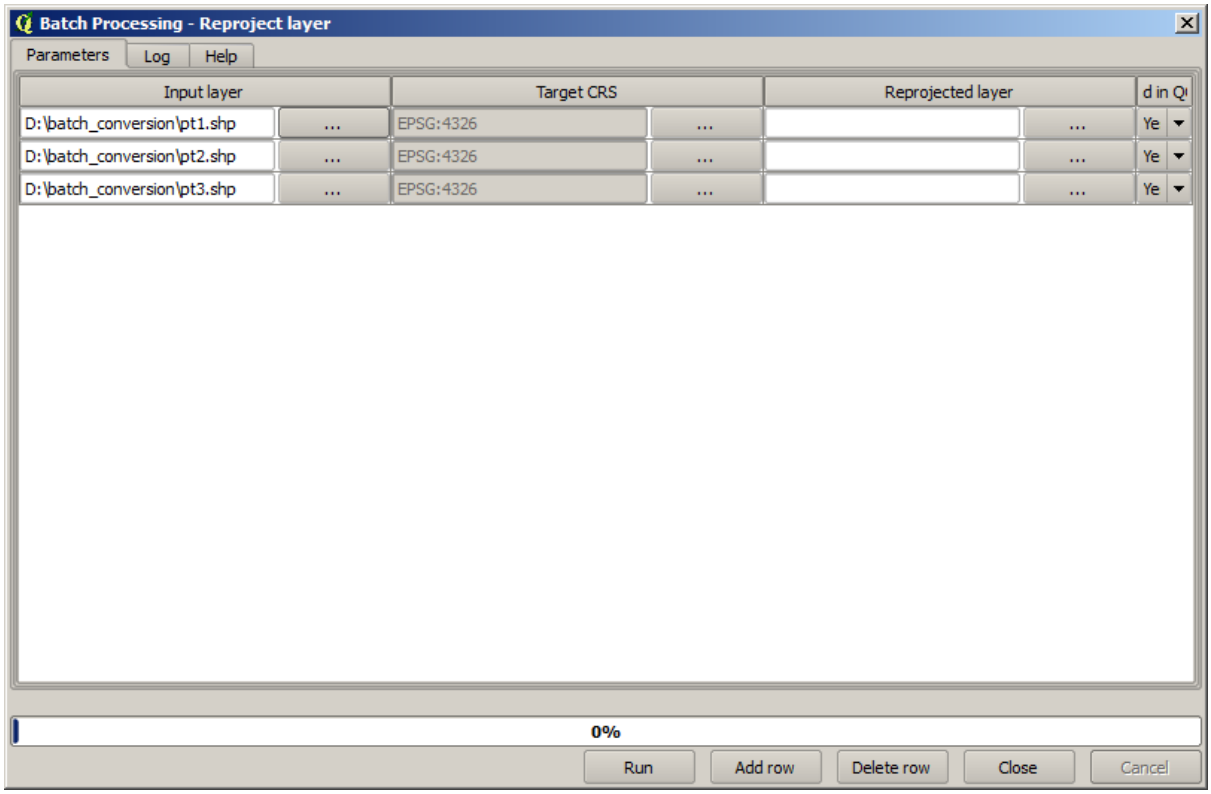


Wenn wir uns die Daten dieser Lektion ansehen, bemerken wir drei shape-Dateien aber keine QGIS Projektdatei. Wenn ein Algorithmus als Stapelprozess aufgerufen wird, können die Layer sowohl aus dem QGIS Projekt als auch aus einer Datei gewählt werden. Das macht es einfacher, eine große Anzahl an Layern zu verarbeiten, z.B. alle Layer unter einem bestimmten Dateipfad.

Jede Zeile in der Tabelle der Stapelverarbeitung steht für die einmalige Ausführung des Algorithmus. Die Zellen in einer Zeile stehen für die vom Algorithmus verwendeten Parameter. Sie sind nicht einer über dem anderen, wie sonst üblich, sondern nebeneinander angeordnet.

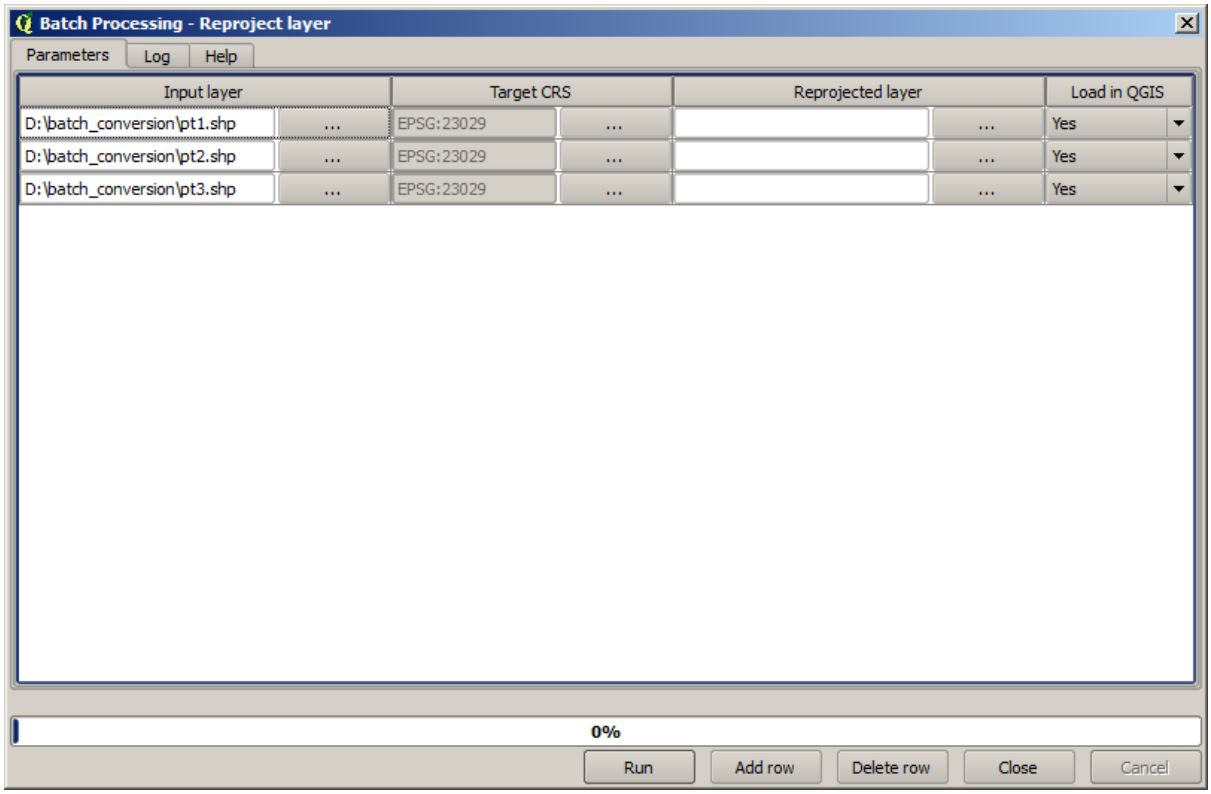
Defining the batch process to run is done by filling the table with the corresponding values, and the dialog itself contains several tools to make this task easier.

Lassen Sie uns die Felder eines nach dem anderen füllen. Die erste zu füllende Spalte ist *Eingabelayer*. Anstelle jeden Layer einzeln einzugeben, können wir alle Layer gleichzeitig wählen. Die Stapelverarbeitung fügt sie in jeweils neue Zeilen ein. Klicken Sie auf die Schaltfläche in der Zelle oben links. Wählen Sie in der nun geöffneten Dateiauswahl die drei zu reprojizierenden Dateien aus. Da nur eine Datei je Zeile benötigt wird, werden die anderen Dateien zum Füllen der Zeilen darunter verwendet.



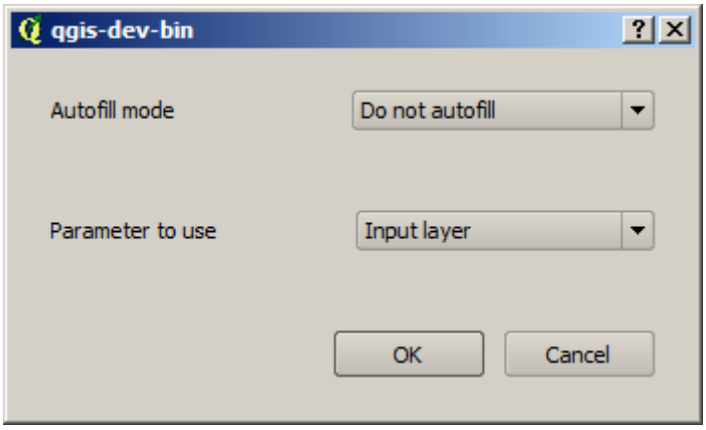
Als Voreinstellung werden 3 Zeilen angezeigt, genau so viele wie wir benötigen. Wenn Sie mehr Layer auswählen werden zusätzliche Zeilen angelegt. Wenn Sie die Einträge manuell füllen möchten, können Sie neue Zeilen mit Hilfe der Schaltfläche *Zeile hinzufügen* einfügen.

Wir werden alle Layer in das KBS EPSG:23029 konvertieren. Wir müssen daher dieses KBS im zweiten Feld eintragen. Wir wollen dasselbe KBS für alle Zeilen verwenden. Es reicht, das KBS in der ersten Zeile (ganz oben) einzutragen und den Eintrag anschließend durch Doppelklick auf den Tabellenkopf in die anderen Zellen zu übernehmen.

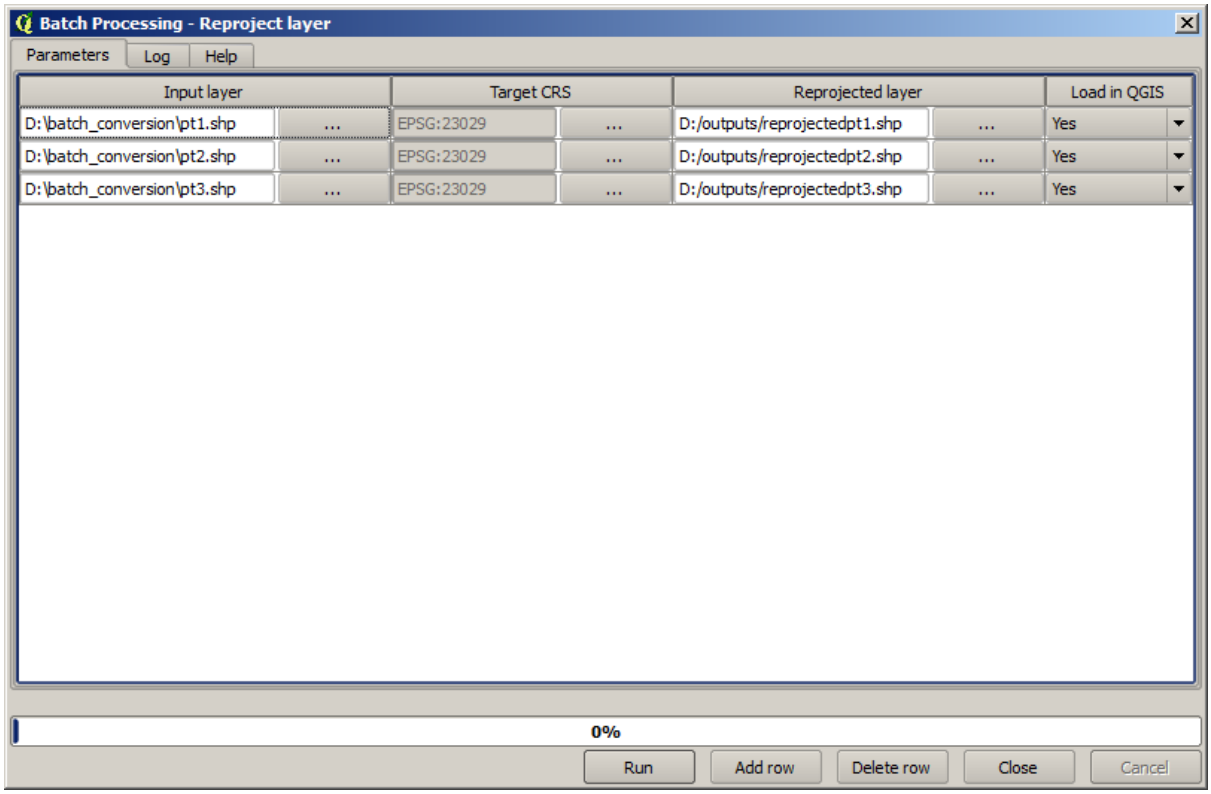


Als Letztes müssen wir eine Ausgabedatei für jede Ausführung wählen. Wir beginnen mit der ersten Zeile. Klicken Sie auf die Schaltfläche in der obersten Zelle und vergeben Sie einen Dateinamen im gewünschten Ordner (z.B. `reprojiziert.shp`).

Nachdem Sie auf die *OK* Schaltfläche gedrückt haben, erscheint ein neues Fenster wie dieses.



Wenn Sie die erste Option wählen, wird nur die aktuelle Zelle gefüllt. Wenn Sie etwas anderes wählen, werden alle folgenden Zeilen nach einem bestimmten Muster gefüllt. Wir wählen *Mit Parameterwerten füllen* und in der Auswahl darunter *Eingabelayer*. Das führt dazu, dass an den von uns gewählten Dateinamen der Wert *Eingabelayer* (das ist der Layername) angehängt wird. Damit erhält jede Ausgabe einen anderen Dateinamen. Die Stapelverarbeitung sollte nun wie folgt aussehen.



In der letzten Spalte wird eingestellt, ob der Ergebnislayer im aktuellen QGIS Projekt geladen werden soll. Belassen Sie die Einstellung bei *Ja*, um die Ergebnisse anzuzeigen.

Klicken Sie auf *Starte*, um die Stapelverarbeitung zu beginnen. Wenn alles klappt, werden alle Ihrer Layer verarbeitet und 3 neue Layer erstellt.

17.27 Modelle innerhalb der Stapelverarbeitung

Warnung: Achtung, dieses Kapitel wurde nicht weiter getestet. Bitte melden Sie auftrende Probleme. Es sind keine Bilder enthalten.

Bemerkung: Diese Lektion zeigt eine weiteres Beispiel für die Stapelverarbeitung. Dieses Mal wird ein Modell anstelle eines integrierten Algorithmus verwendet.

Modelle funktionieren genauso wie andere Algorithmen. Sie können innerhalb der Stapelverarbeitung verwendet werden. Um das zu demonstrieren, folgt ein kurzes Beispiel, das unser schon gut bekanntes hyrologisches Modell verwendet.

Stellen Sie sicher, dass das Modell in Ihrer Werkzeugkiste enthalten ist und starten es dann als Stapelprozess. Der Dialog der Stapelverarbeitung sollte wie folgt aussehen.

Warnung: todo: Bild einfügen

Fügen Sie weitere Zeilen ein, so dass insgesamt 5 Zeilen enthalten sind. Wählen Sie das zu dieser Lektion gehörende DEM als Eingabe für alle Zeilen. Geben Sie dann 5 verschiedene Schwellenwerte wie hier gezeigt ein.

Warnung: todo: Bild einfügen

Wie man sieht, kann die Stapelverarbeitung nicht nur mit demselben Prozess und verschiedenen Eingabedaten sondern auch mit denselben Daten aber verschiedenen Parametern gestartet werden.

Nach dem Bestätigen mit *OK* sollten wir 5 neue Layer mit Wasserscheiden entsprechend der 5 vergebenen Schwellenwerte erhalten.

17.28 Aktionsskripte vor oder nach Ausführung

Bemerkung: Diese Lektion ziegt die Ausführung von Aktionsskripten die die Ausführung weiterer Operationen vor oder nach der Verarbeitung eines Algorithmus erlauben.

Aktionsskripte sind Verarbeitungsskripte die direkt vor oder nach einer Prozessierung laufen. Sie können verwendet werden, um Aufgaben die in Verbindung mit einem Algorithmus abgearbeitet werden, zu automatisieren.

Die Syntax von Aktionsskripten ist mit der von Verarbeitungsskripten identisch. Weiter Details findet man unter Kapitel im QGIS Benutzerhandbuch.

In Ergänzung zu den Eigenschaften der normalen Skripte, kann in Aktionsskripten eine besondere globale Variable mit dem Namen `alg` verwendet werden. Sie repräsentiert den gerade ausgeführten (bzw. den in Kürze auszuführenden) Algorithmus.

Hier ist ein Beispiel eines Aktionsskriptes, das nachfolgend ausgeführt wird. In der Voreinstellung wird bei der Verarbeitung nur temporär gespeichert. Dieses Skript kopiert die Ausgaben in ein bestimmtes Verzeichnis, so dass die Dateien beim Beenden von QGIS nicht verlorengehen.

```
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 den ersten beiden Zeilen werden die benötigten Python Pakete importiert: `os` — für den Umgang mit Dateipfaden, wie z.B. die Gewinnung eines Dateinamens und `shutil` — für verschiedene Dateioperationen wie z.B. das Kopieren von Dateien. In der dritten Zeile importieren wir die Verarbeitungssausgaben. Was das bedeutet wird später in dieser Lektion erklärt.

Als Nächstes definieren die Konstante `MY_DIRECTORY`. Dabei handelt es sich um den Pfad zum Verzeichnis, in das wir die Ergebnisse unserer Analyse ablegen möchten.

Am Ende des Skriptes finden wir den eigentlichen Programmkode. In einer Schleife wird über alle Ausgaben von Algorithmen geprüft, ob sie dateibasiert sind und kopiert werden können. Wenn ja wird das oberste Verzeichnis der Ausgabedateien bestimmt und dann alle Datei in unser Verzeichnis kopiert.

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.

Auf ähnliche Weise können wir Aktionsskripte implementieren, die vor der Ausführung eines Algorithmus gestartet werden. Lassen Sie uns als Beispiel ein Aktionsskript erstellen, das Eingabevektorlayer auf Geometriefehler prüft.

```
from qgis.core import QgsGeometry, QgsFeatureRequest
from processing.core.parameters import ParameterVector

for param in alg.parameters:
    if isinstance(param, ParameterVector):
        layer = processing.getObject(param.value)
        for f in layer.getFeatures(QgsFeatureRequest().setSubsetOfAttributes([])):
            errors = f.geometry().validateGeometry()
            if len(errors) > 0:
                progress.setInfo('One of the input vectors contains invalid␣
↪geometries!')
```

Wie im vorherigen Beispiel werden zuerst die benötigten QGIS- und Prozessierungspakete importiert.

Anschließend wird iterativ über alle Parameter des Algorithmus geprüft, ob ein Vektorparameter vorhanden ist. Wenn ja wird der zugehörige Vektorlayer als Objekt zurückgegeben. Danach werden alle Objekte des Vektorlayers auf Geometriefehler geprüft. Wenn mindestens ein Objekt eine ungültige Geometrie enthält, wird eine Fehlermeldung ausgegeben.

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 Andere Programme

Module contributed by Paolo Cavallini - Faunalia

Bemerkung: Dieses Kapitel zeigt wie man zusätzliche Programme innerhalb der Verabeitungsumgebung verwenden kann. Die für dieses Kapitel relevanten Pakete müssen vorher mit Hilfe des verwendeten Betriebssystems installiert worden sein.

17.29.1 GRASS

GRASS ist eine frei verfügbare Open Source GIS Anwendung für Datenmanagement, Analyse, Bildverarbeitung, Bild- und Kartenerstellung, räumliche Modellierung und Visualisierung.

Unter Windows wir die Anwendung mit Hilfe des OSGeo4W Installationsprogrammes (32 und 64 bit) in der Voreinstellung installiert. Für Linux ist das Programm für alle bedeutenden Distributionen paketiert.

17.29.2 R

R ist eine freie Open Source Anwendung für statistische Berechnungen und Grafiken.

Es mus separat zusammen mit einigen erforderlichen Bibliotheken (**LIST**) installiert werden. Um R in QGIS nutzen zu können, muss zusätzlich die Erweiterung *Processing R Provider* installiert sein.

Das schöne an der Umsetzung der Verarbeitungsumgebung ist, dass Sie eigene Skripte - einfache oder komplizierte - in die Umgebung einfügen können. Sie können dann wie jedes andere Modul, z.B. innerhalb komplexer Arbeitsabläufe, genutzt werden.

Testen Sie einige der vorinstallierten Beispiele, wenn Sie R schon installiert haben (denken Sie daran die R Module in den Verarbeitungseinstellungen zu aktivieren).

17.29.3 Andere

Die für mehrere Betriebssystem verfügbaren LASTools sind eine Zusammenstellung verschiedener Programme zur Verarbeitung und Analyse von LiDAR Daten.

Weitere Werkzeuge sind als Erweiterung verfügbar, z.B.:

- LecoS: eine Umgebung für Landbedeckungsstatistik und Lanschaftsökologie
- lwgeom: die früher zu PostGIS gehörende Bibliothek enthält einige nützliche Werkzeuge zur Geometriebereinigung
- Animove: Werkzeuge zur Analyse des Lebensraumes von Tieren.

Weitere werden folgen.

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)

Bemerkung: 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 und Kontourlinienerzeugung

Module contributed by Paolo Cavallini - Faunalia

Bemerkung: 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`:

Warnung: 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 Vektorvereinfachung und Glättung

Module contributed by Paolo Cavallini - Faunalia

Bemerkung: 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 Planung eines Solarparks

Module contributed by Paolo Cavallini - Faunalia

Bemerkung: 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 Nutzung von R-Skripten in der Prozessierung

Modul erstellt von Matteo Ghetta - finanziert von der Scuola Superiore Sant'Anna

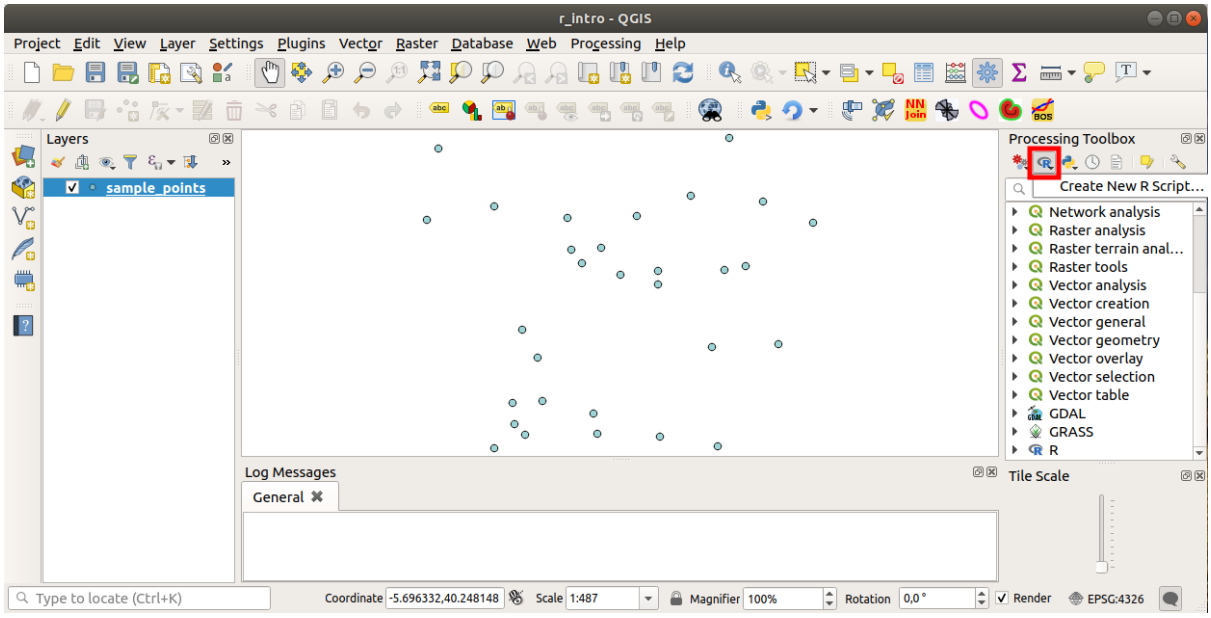
Die Verarbeitungsumgebung (mit der Erweiterung `Processing R Provider`) macht das Schreiben und Ausführen von R Skripten innerhalb von QGIS möglich.

Warnung: R muss auf Ihrem Rechner installiert und die PATH-Variable korrekt gesetzt sein. Die Verarbeitungsumgebung ruft die externen R Packete nur auf. Sie kann sie nicht selbst installieren. Stellen Sie daher sicher, dass die externen Pakete direkt in R installiert werden. Im Benutzerhandbuch Kapitel findet man dazu weitere Informationen.

Bemerkung: Wenn es Probleme mit *Paketen* gibt, fehlen unter Umständen für die Verarbeitung *erforderliche* Pakete wie `sp`, `rgdal` und `raster`.

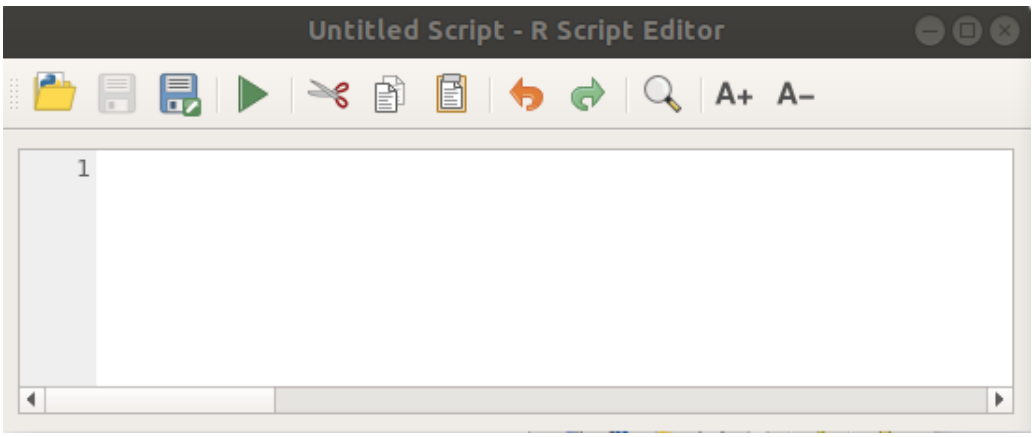
17.33.1 Hinzufügen von Skripten

Das Hinzufügen eines Skriptes ist einfach. Der einfachste Weg ist das Öffnen der Werkzeugkiste. In der Werkzeugleiste der Werkzeugkiste wählt man dann *Create new R script…* aus dem R Menü (durch das R Symbol gekennzeichnet). Man kann auch ein Skript im Texteditor erstellen und es anschließend im R Skript Ordner (`processing/rscripts`) ablegen. Sobald es dort abgelegt wurde, wird es zur Bearbeitung in der Werkzeugkiste verfügbar. Man klickt dazu mit der rechten Maustaste auf den Skriptnamen und wählt *Edit Script…*).



Bemerkung: Wenn Sie R nicht in der Verarbeitungsumgebung finden, müssen Sie es unter *Optionen* ► *Verarbeitung* ► *Datenanbieter* aktivieren.

Es öffnet sich ein Fenster *R Script Editor*. Hier müssen einige Parameter definiert werden, bevor das Skript an sich erstellt werden kann.



17.33.2 Erstellung von Plots

In dieser Anleitung erstellen wir einen **Box-Plot** für ein Feld eines Vektorlayers.

Öffnen das QGIS Projekt `r_intro.qgs` im Ordner `exercise_data/processing/r_intro/`.

Skript Parameter

Öffnen Sie den Editor und beginnen mit dem Schreiben am Anfang.

Sie **müssen** einige Parameter **vor** dem Hauptteil des Skripts festlegen:

1. Der Name der Gruppe (*plots* in diesem Fall) in der Sie Ihr Skript stellen wollen (wenn die Gruppe nicht existiert, wird sie erstellt):

   ```
   ##plots=group
   ```

   Sie finden Ihr Skript in der **plots** R Gruppe in den Verarbeitungswerkzeugen.

2. Sie müssen der Verarbeitung mitteilen, dass Sie ein Plot (in diesem Beispiel) darstellen möchten:

   ```
   ##showplots
   ```

   Sie erhalten dann in der **Ergebnisanzeige** einen Link zum Plot (das kann unter *Ansicht* ► *Bedienfelder* und mit *Verarbeitung* ► *Ergebnisanzeige* ein- oder ausgeschaltet werden).

3. Sie müssen der Verarbeitung auch die Eingabedaten vorgeben. In diesem Beispiel wollen wir ein Plot für ein Feld eines Vektorlayers erstellen:

   ```
   ##Layer=vector
   ```

   Die Verarbeitung weiß nun, dass die Eingabe eine Vektordatei ist. Der Name *Layer* ist nicht wichtig, entscheidend ist der Parameter **vector**.

4. Als Letztes müssen Sie das Eingabefeld des Vektorlayers vorgeben (unter Nutzung des Namens, den Sie oben vergeben haben - *Layer*):

   ```
   ##X=Field Layer
   ```

   Die Verarbeitung weiß nun, dass Sie ein Feld *Layer* benötigen und es **X** nennen.

5. Man kann den Namen des Skriptes auch mit Hilfe von `name` definieren:

   ```
   ##My box plot script=name
   ```

   Wenn nichts vorgegeben ist, wird der Dateiname als Name des Skriptes verwendet.

Skript Hauptteil

Nachdem wir den Kopf des Skriptes festgelegt haben, können wir die Funktionalität hinzufügen:

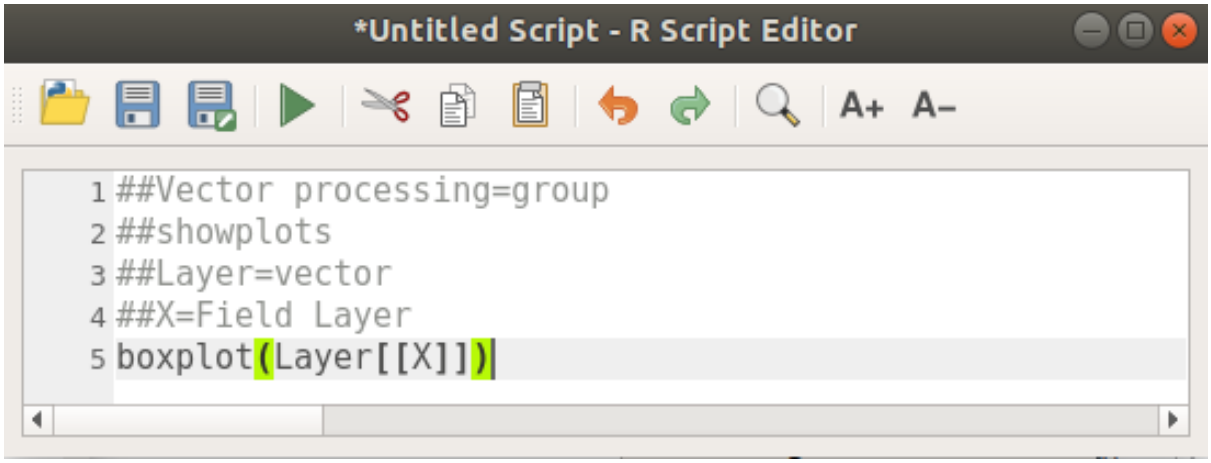
```
boxplot(Layer[[X]])
```

boxplot ist der Name der R Funktion. Der Parameter **Layer** ist der Name, den wir für den Eingabedatensatz vorgegeben haben. **X** ist der vorgegebene Name des Feldes im Datensatz.

> **Warnung:** Der Parameter **X** muss innerhalb doppelter eckiger Klammern stehen (`[[]]`).

Das fertige Skript sollte wie folgt aussehen:

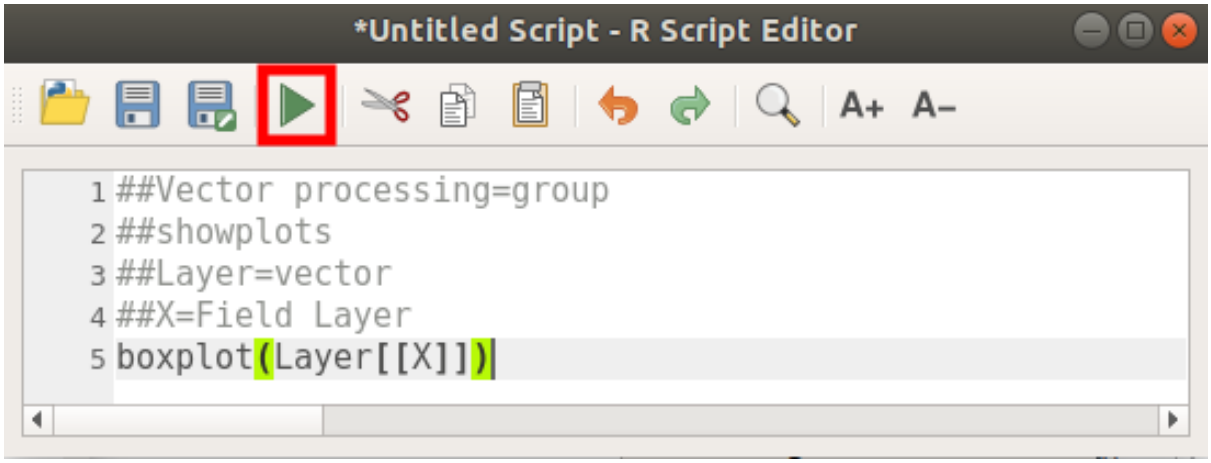
```
##Vector processing=group
##showplots
##Layer=vector
##X=Field Layer
boxplot(Layer[[X]])
```



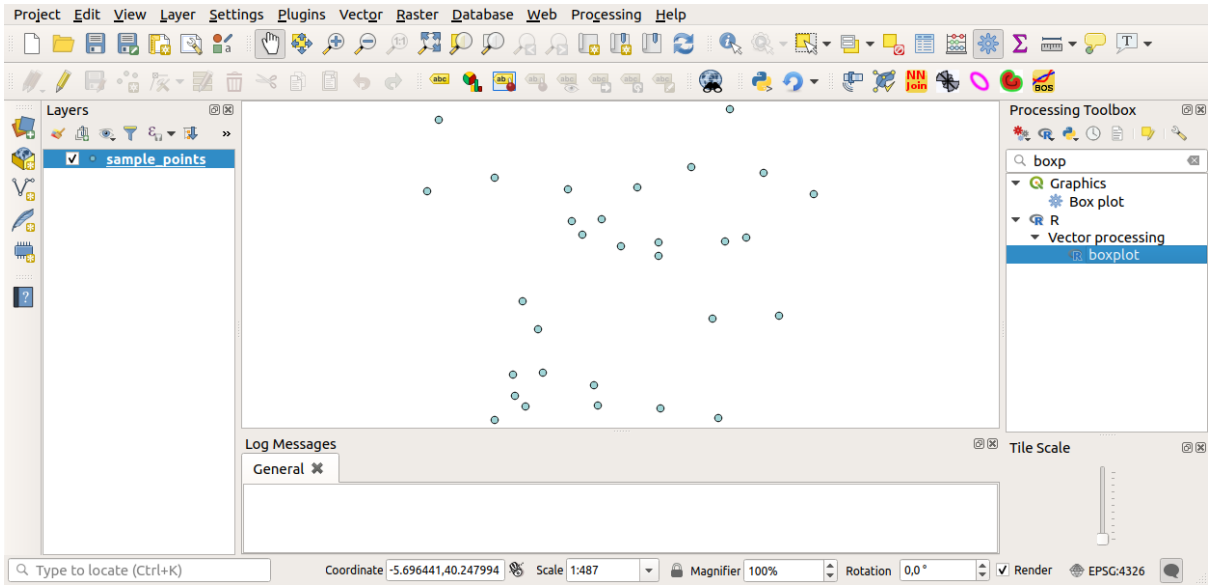
Speichern Sie das Skript im voreingestellten Pfad entsprechend des Vorschlags der Verarbeitung (processing/rscripts). Wenn Sie `name` im Kopf des Skriptes nicht definiert haben, wird das Skript in der Verarbeitungsumgebung mit dem Dateinamen bezeichnet.

Bemerkung: Sie können das Skript an einem beliebigen Ort speichern. Die Verarbeitung übernimmt das Skript dann nicht automatisch in die Verarbeitungsumgebung und es muss manuell hinzugefügt werden.

Starten Sie jetzt das Skript durch Drücken auf die ausführen Schaltfläche im Editorfenster:



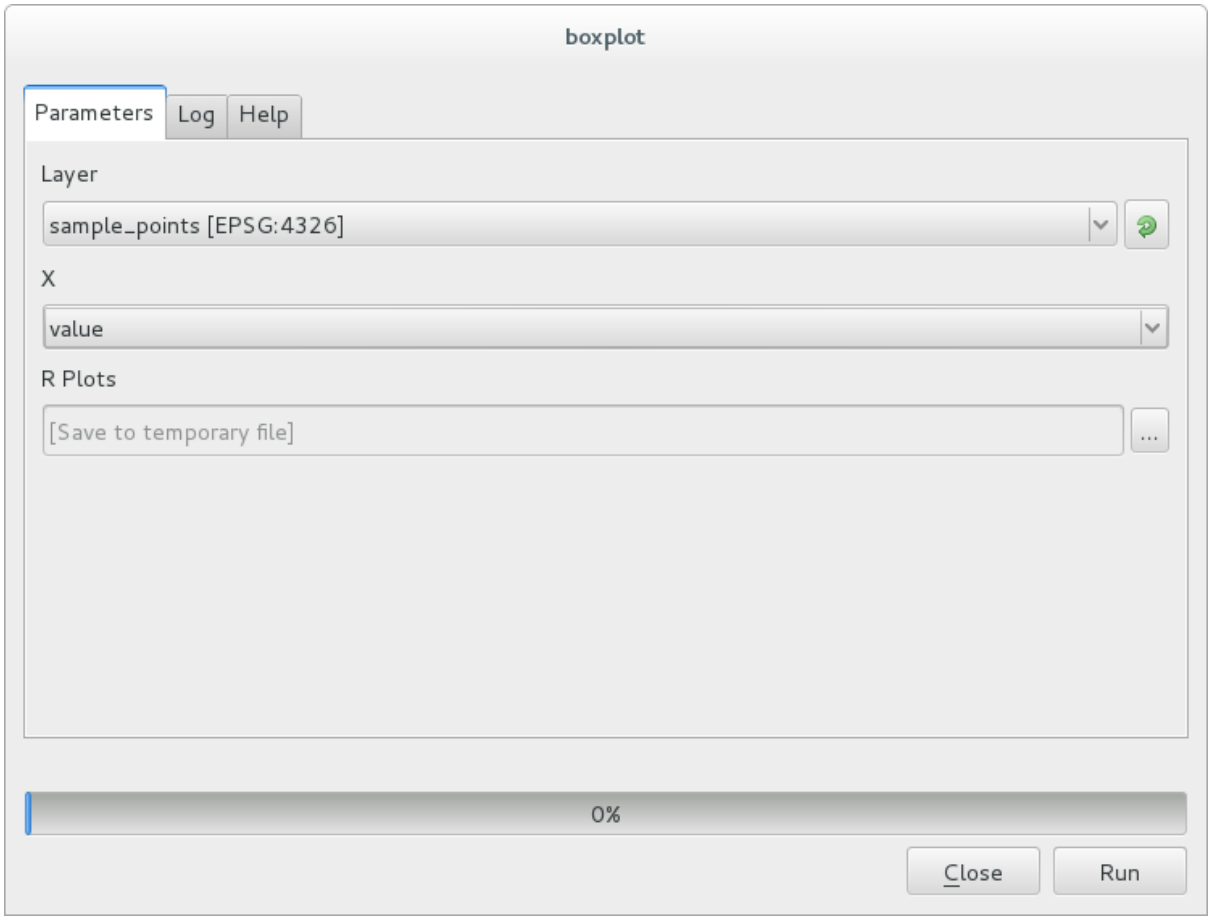
Nutzen Sie nach dem Schließen des Editorfensters die Textbox der Verarbeitung, um Ihr Skript zu finden:



Sie können nun die erforderlichen Parameter im Fenster des Verarbeitungsalgorithmus eintragen:

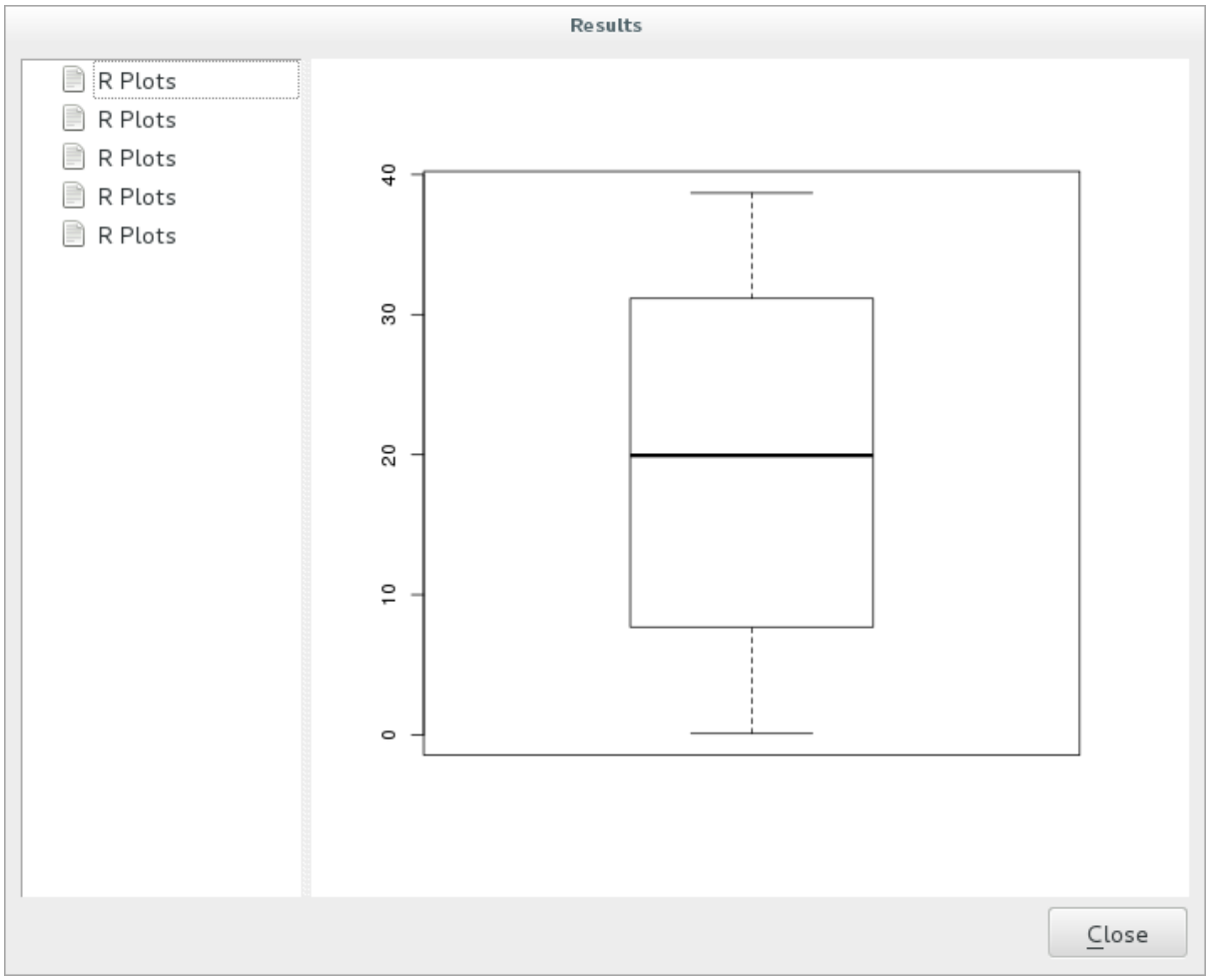
- wählen Sie *sample_points* als **Layer**
- wählen Sie *value* als **X** Feld

Klicken Sie auf **Starte**.



Die **Ergebnisanzeige** sollte sich automatisch öffnen. Falls das nicht geschieht, klicken Sie auf *Verabeitung* ► *Ergebnisanzeige*.

Wenn Sie auf den Link in der Ergebnisanzeige klicken, sehen Sie das Folgende:



Bemerkung: Sie können das Bild nun durch Rechtsklick auf den Plot kopieren und speichern.

17.33.3 Erstellen eines Vektorlayers

Sie können außerdem einen Vektorlayer erstellen und ihn automatisch in QGIS laden.

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).

Das Ziel dieser Übung ist die Erstellung eines zufälligen Punktlayers mit Hilfe der `spsample` Funktion aus dem `sp` Paket. Dabei soll die Ausdehnung über einen Eingabevektorlayer begrenzt werden.

Skript Parameter

Wie vorhin müssen wir zuerst einige Parameter oberhalb des Hauptteiles des Skriptes vorgeben:

1. Geben Sie den Namen der Gruppe vor, in die das Skript soll. In diesem Fall ist das *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. Geben Sie einen Eingabeparameter für die Anzahl der zu erzeugenden Punkte vor (`Size`, der Vorgabewert ist `10`):

   ```
   ##Size=number 10
   ```

   Bemerkung: Da ein Vorgabewert (`10`) vorhanden ist, kann der Nutzer den Parameter so belassen oder einen eigenen Wert eingeben.

   ---

4. Geben Sie vor, dass ein Ausgabevektorlayer erzeugt wird (bezeichnet mit `Output`):

   ```
   ##Output=output vector
   ```

Skript Hauptteil

Sie können jetzt den Hauptteil der Funktion hinzufügen:

1. Verwenden Sie die Funktion `spsample`:

   ```
   pts=spsample(Layer, Size, type="random")
   ```

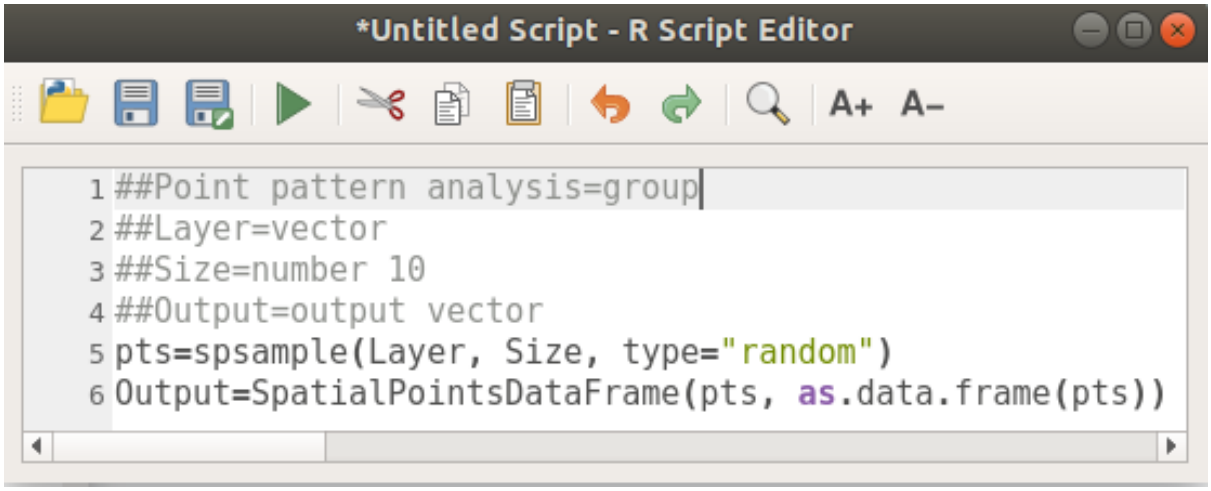
   Die Funktion verwendet den *Layer*, um die Platzierung der Punkte zu beschränken (wenn der Layer eine Linie ist, müssen die Punkte auf der Linie liegen; wenn es ein Polygon ist, müssen die Punkte innerhalb des Polygons liegen). Die Anzahl der Punkt entspricht der Vorgabe im Parameter *Size*. Das Stichprobenverfahren ist *random*.

2. Erstellen Sie die Ausgabe (der Parameter `Output`):

   ```
   Output=SpatialPointsDataFrame(pts, as.data.frame(pts))
   ```

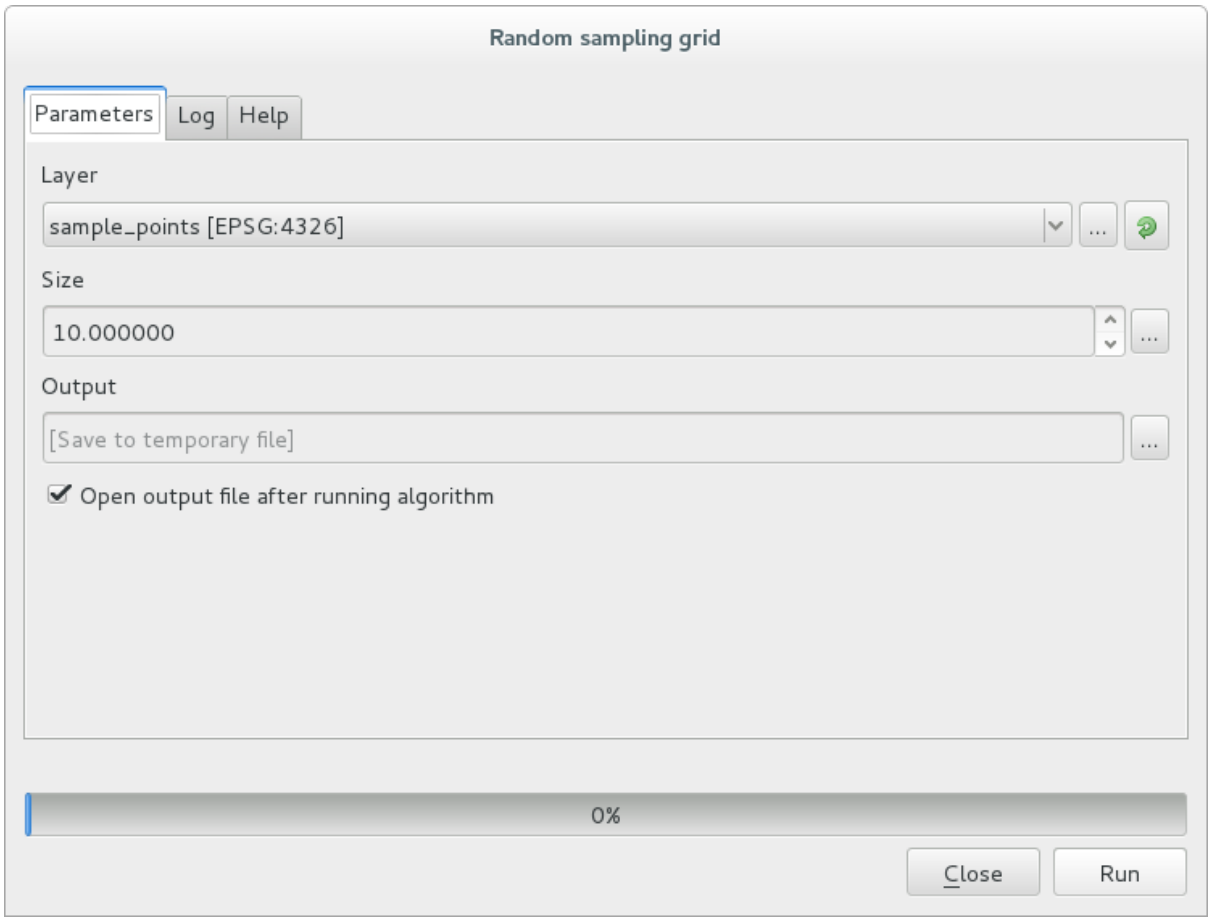
Das fertige Skript sollte wie folgt aussehen:

```
##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))
```



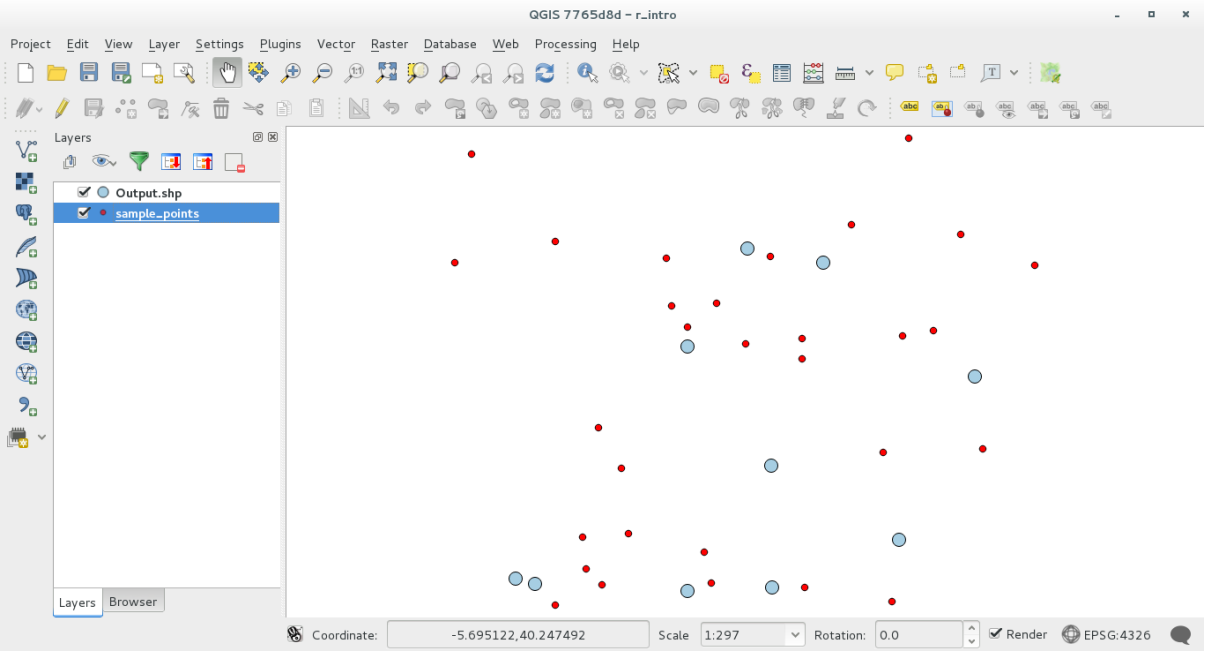
Speichern Sie es und klicken Sie auf die Schaltfläche zum Starten.

Geben Sie in dem neu geöffneten Fenster die richtigen Parameter ein:



und klicken Sie auf Starte.

Der Ergebnislayer wird im Inhaltsverzeichnis eingefügt und die Punkte werden im Kartenfenster angezeigt:



17.33.4 Textausgabe und grafische Ausgabe mit der R - Syntax

Die Verarbeitung (mit dem `Processing R Provider plugin`) nutzt eine spezielle Syntax, um die Ergebnisse aus R zu erhalten:

- > vor einem Kommando, wie in `>lillie.test(Layer[[Field]])` bedeutet, dass das Ergebnis an die R Ausgaeb gesendet wird (Ergebnisanzeige)
- + nach einem Plot erlaubt sich überlagernde Plots. Zum Beispiel `plot(Layer[[X]], Layer[[Y]]) + abline(h=mean(Layer[[X]]))`

17.34 Predicting landslides

Module contributed by Paolo Cavallini - Faunalia

Bemerkung: 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*)

KAPITEL 18

Module: Räumliche Datenbanken in QGIS nutzen

In diesem Modul lernen Sie, wie Sie räumliche Datenbanken mit QGIS verwenden, um Daten in der Datenbank zu verwalten, anzuzeigen und zu manipulieren sowie Analysen durch Abfragen durchzuführen. Wir werden in erster Linie PostgreSQL und PostGIS verwenden (die in den vorherigen Abschnitten behandelt wurden), aber die gleichen Konzepte sind auch auf andere räumliche Datenbankimplementierungen einschließlich SpatiaLite anwendbar.

18.1 Lesson: Arbeit mit Datenbanken im QGIS Browser

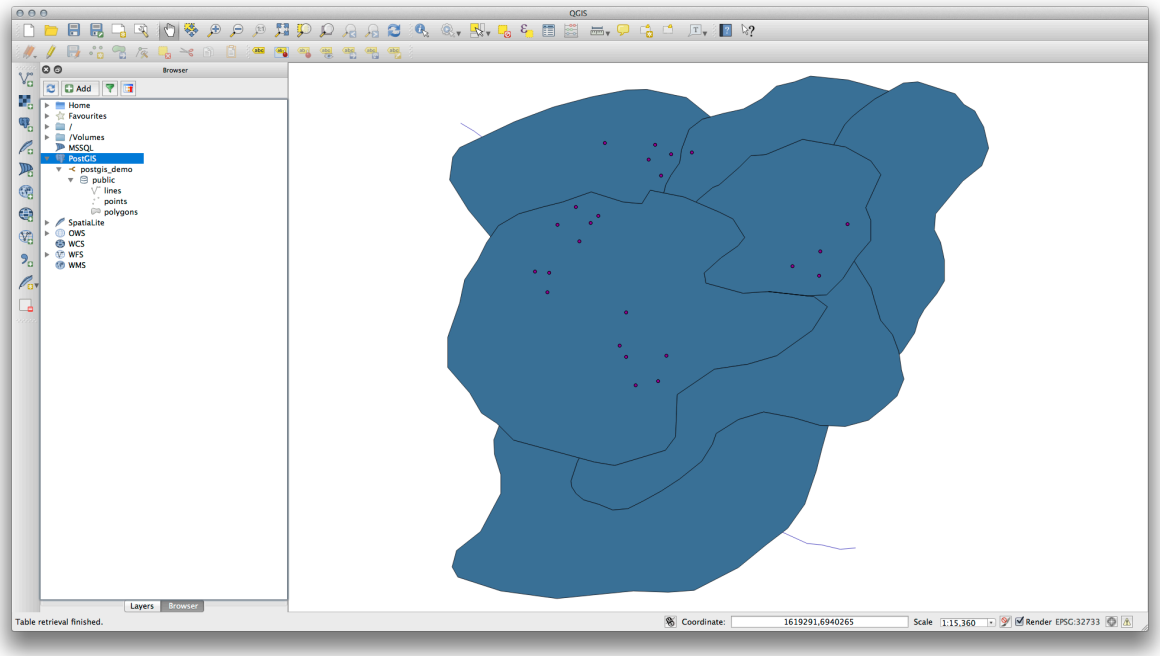
In den vorhergehenden 2 Modulen haben wir grundlegende Konzepte, Möglichkeiten und Funktionen von relationalen Datenbanken behandelt. Wir haben Erweiterungen betrachtet, die das Speichern, Verwalten, Abfragen und Verändern von räumlichen Daten in relationalen Datenbanken ermöglichen. In diesem Abschnitt gehen wir darauf ein, wie man räumliche Datenbanken effizient in QGIS verwendet.

Ziel dieser Lektion: Erlernen des Umgangs mit räumlichen Datenbanken unter Nutzung des QGIS Browsers.

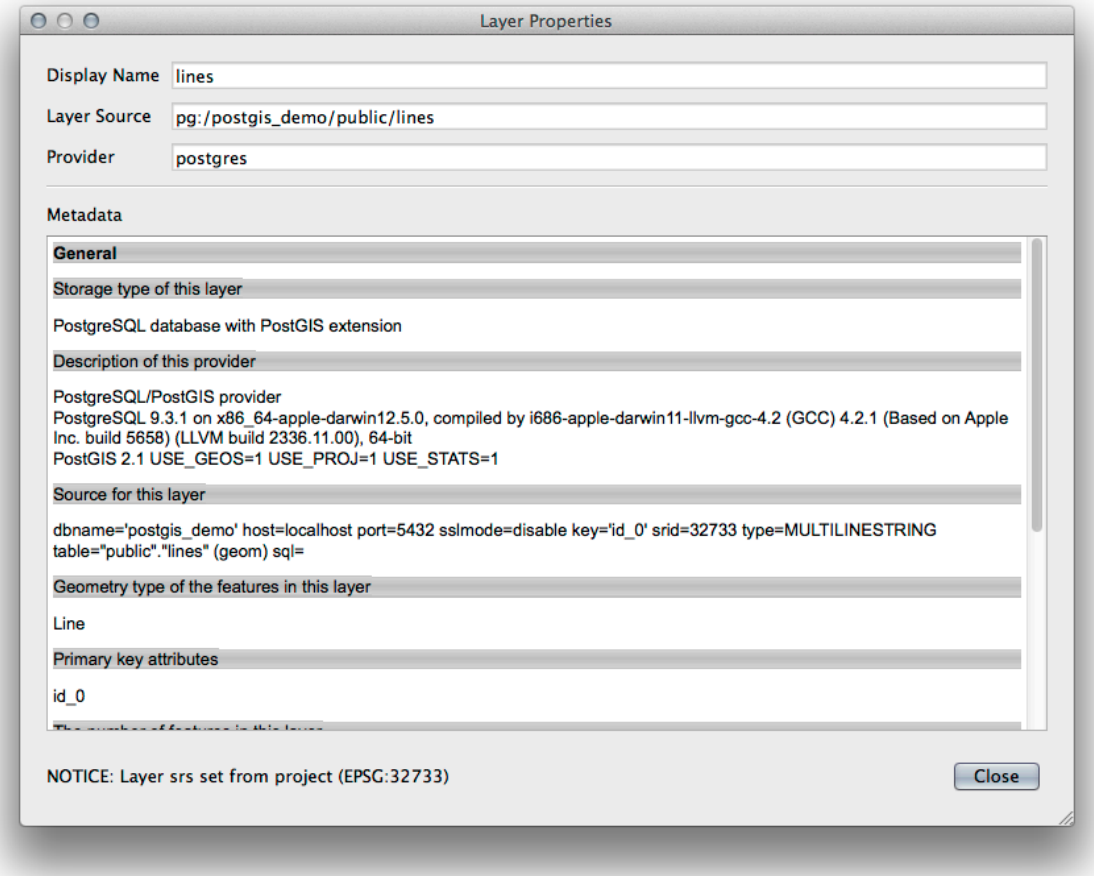
18.1.1 Follow Along: Datenbanktabellen mit Hilfe des Browsers zu QGIS hinzufügen

Wir haben schon behandelt, wie man Tabellen aus einer Datenbank als QGIS Layer hinzufügt. Wir gehen jetzt detaillierter darauf ein und werden noch andere Wege kennenlernen, wie man das in QGIS erledigen kann. Sehen wir uns zuerst die neue Browserschnittstelle an.

- Erstellen Sie einen neue leere Karte in QGIS.
- Öffnen Sie den Browser durch Klick auf den Reiter *Browser* unten im *Layer Bedienfeld*
- Öffnen Sie den PostGIS Bereich im Baum des Browsers. Hier sollte Ihre vorher konfigurierte Verbindung verfügbar sein (Sie müssen unter Umständen auf die Aktualisieren Schaltfläche oben im Browserfenster drücken).



- Ein Doppelklick auf eine beliebige Tabelle oder Layer fügt ihn in den Kartenbereich ein.
- Mit Rechtsklick auf einen Layer oder eine Tabelle in dieser Ansicht erhalten wir einige Optionen. Klicken Sie auf den Eintrag *Eigenschaften*, um die Eigenschaften des Layers anzuzeigen.



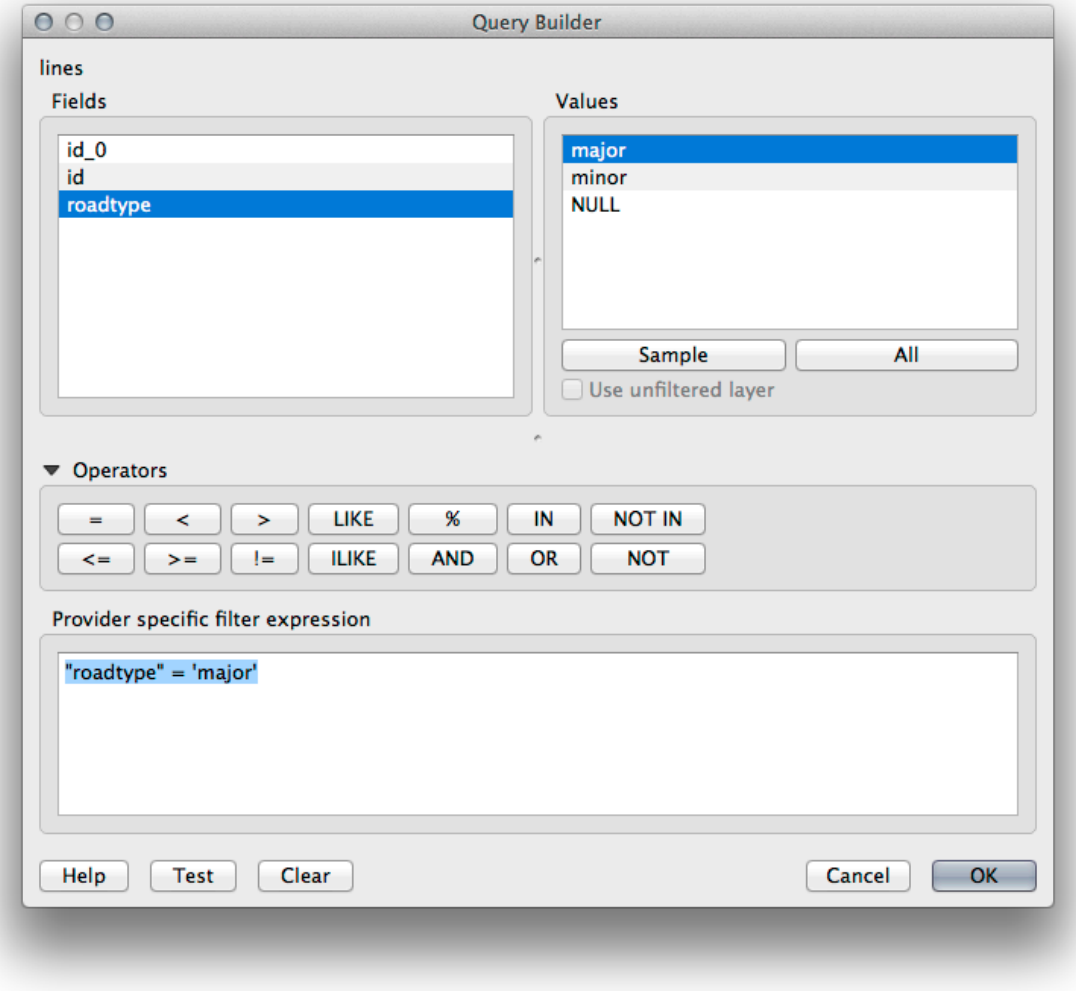
Bemerkung: Natürtlich können Sie mit Hilfe dieser Schnittstelle auch eine Verbindung zu PostGIS Datenbanken auf einem externen Server herstellen. Mit Rechtsklick auf den PostGIS Eintrag im Baum können Sie die Verbindungsparamter für eine neue Verbindung vorgeben.

18.1.2 Follow Along: Hinzufügen einer gefilterten Liste von Datensätzen als Layer

Nachdem wir gesehen haben, wie man eine vollständige Tabelle als QGIS Layer hinzufügt, wäre es schön, eine gefilterte Liste an Datensätzen aus einer Tabelle als Layer hinzuzufügen. Wir verwenden dazu Abfragen, die wir in früheren Abschnitten behandelt hatten.

- Erstellen Sie eine neue Karte ohne Layer
- Klicken Sie auf die Schaltfläche *Add PostGIS Layers* oder wählen *Layer* ► *Add PostGIS Layers* aus dem Menü.
- Verbinden Sie sich im Dialog *Add PostGIS Table(s)* mit der `postgis_demo` Verbindung.
- Durch Erweitern von `public schema` sollten Sie drei Tabellen finden, mit denen wir schon vorher gearbeitet hatten.
- Klicken sie auf den Layer `lines`, um ihn auszuwählen. Führen Sie keinen Doppelklick auf den Layer aus sondern klicken auf die Schaltfläche *Set Filter* um den *Query Builder* Dialog zu öffnen.
- Erstellen Sie den folgenden Ausdruck mit Hilfe der Schaltflächen oder durch direkte Eingabe:

```
"roadtype" = 'major'
```



- Klicken Sie auf *OK* um die Bearbeitung des Filters abzuschließen. Klicken sie auf *Add* um den gefilterten Layer in die Karte einzufügen.
- Benennen Sie den Layer `lines` in `roads_primary` um.

Wir sehen, dass nur die Hauptstraßen anstelle des kompletten Layers in unsere Karte übernommen wurden.

18.1.3 In Conclusion

Wir haben gesehen, wie man den QGIS Browser zum Umgang mit räumlichen Datenbanken verwendet und wie man mit einem Abfragefilter versehene Layer in die Karte einfügt.

18.1.4 What's Next?

Als nächstes sehen wir, wie man mit der DB-Verwaltung in QGIS umfangreichere Aufgaben zur Datenbankverwaltung erledigen kann.

18.2 Lesson: Verwendung der DB-Verwaltung zur Arbeit mit räumlichen Datenbanken in QGIS

Wir haben schon eine ganze Menge von Datenbankoperationen mit Hilfe von QGIS und anderen Werkzeugen kennengelernt. Jetzt wollen wir uns die DB-Verwaltung ansehen. Sie enthält viele ähnliche Funktionalitäten sowie zusätzlich Werkzeuge zum Management.

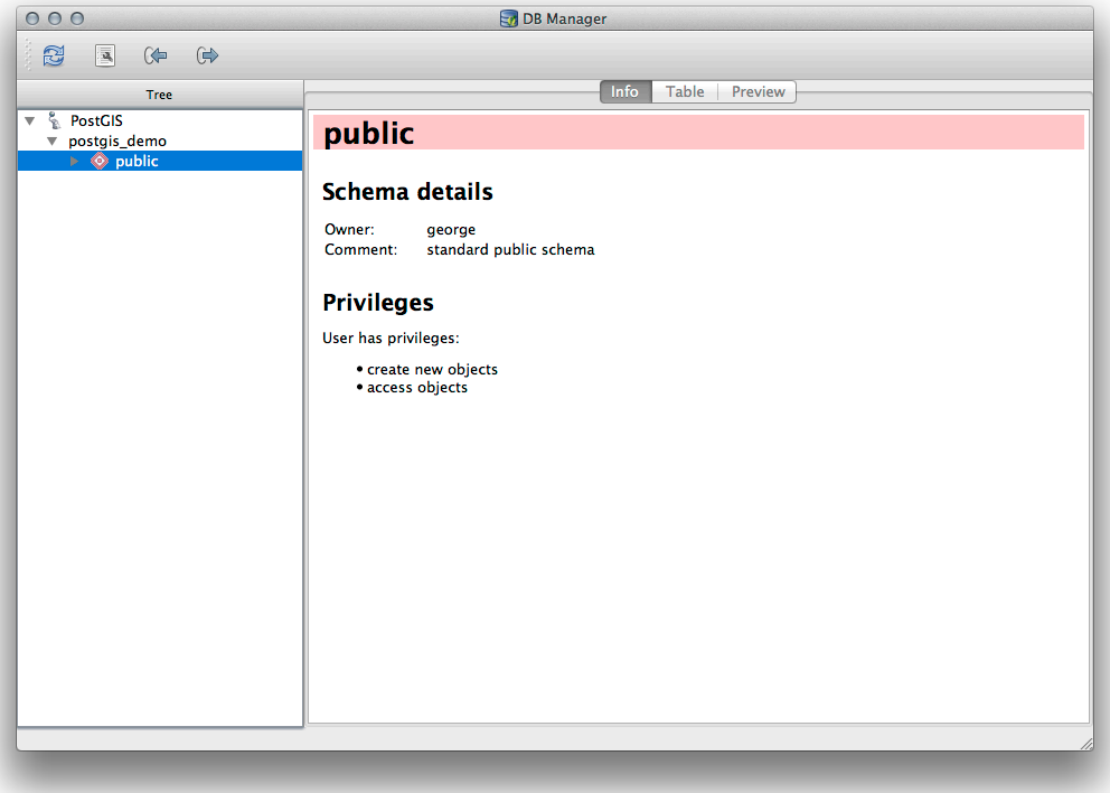
Ziel dieser Lektion: Zu erlernen, wie mit Hilfe der QGIS DB-Verwaltung mit räumlichen Datenbanken interagiert.

18.2.1 Follow Along: Managen von PostGIS Datenbanken mit der DB-Verwaltung

Öffnen Sie als Erstes die DB-Verwaltung durch Auswahl von *Datenbank –> DB-Verwaltung* im Menü durch Auswahl der Schaltfläche DB-Verwaltung in der Werkzeugleiste.

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.

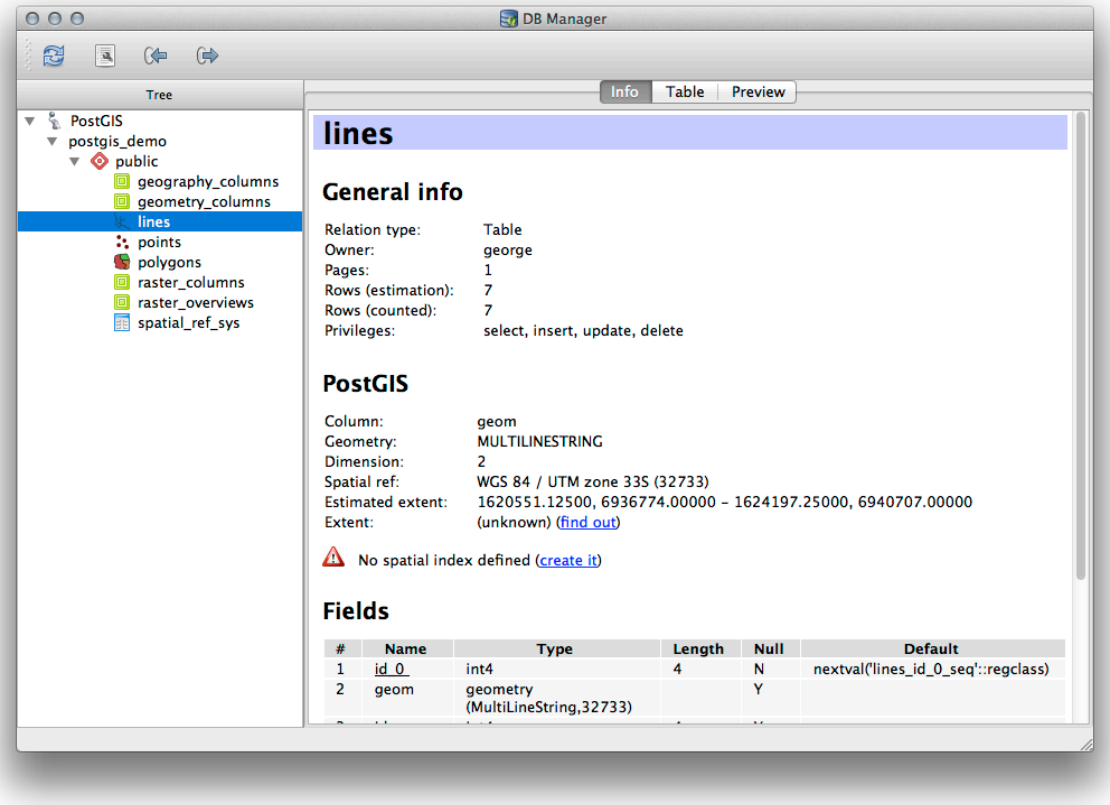
Als Erstes fällt uns vielleicht auf, dass einige Metadaten über die in der Datenbank enthaltenen Schemas angezeigt werden.



Schemas sind eine Möglichkeit, Datentabellen und andere Objekte in einer PostgreSQL Datenbank zu gruppieren. Sie sind ein Behältnis für Zugriffsrechte und andere Einschränkungen. Das Management von PostgreSQL Schemas geht über den Umfang dieser Anleitung hinaus. Weitere Informationen hierzu findet man unter PostgreSQL documentation on Schemas. Man kann die DB-Verwaltung zur Erstellung neuer Schemas verwenden. Allerdings sind für die effektive Verwaltung andere Werkzeuge wie pgAdmin III oder die Kommandozeile erforderlich.

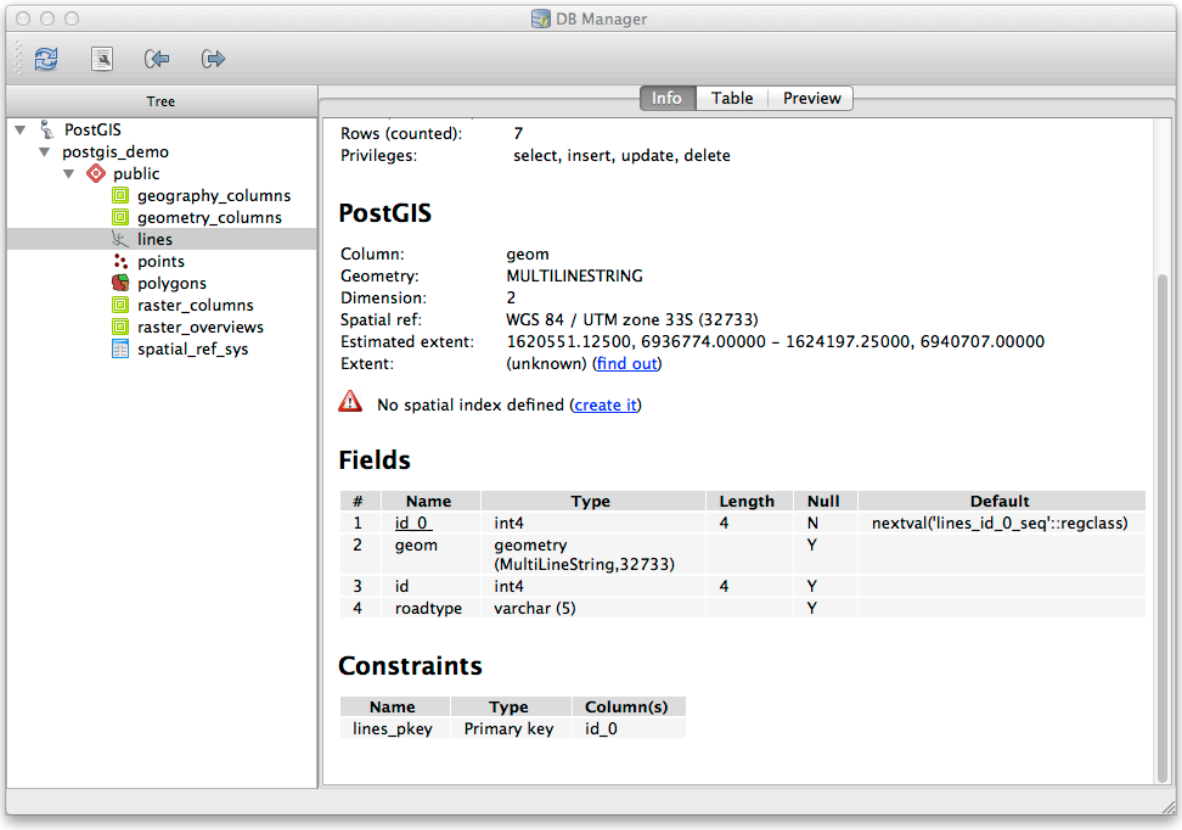
Die DB-Verwaltung kann auch zur Verwaltung der Tabellen in der Datenbank verwendet werden. Wir haben bereits einige Möglichkeiten der Erstellung und Verwaltung von Tabellen mit Hilfe der Kommandozeile behandelt. Sehen wir uns nun an, wie man das mit der DB-Verwaltung erledigen kann.

Als Erstes ist es nützlich, sich die Metadaten einer Tabelle anzusehen. Man klickt dazu im Baum auf den Namen der Tabelle und schaut sich den *Info* Reiter an.

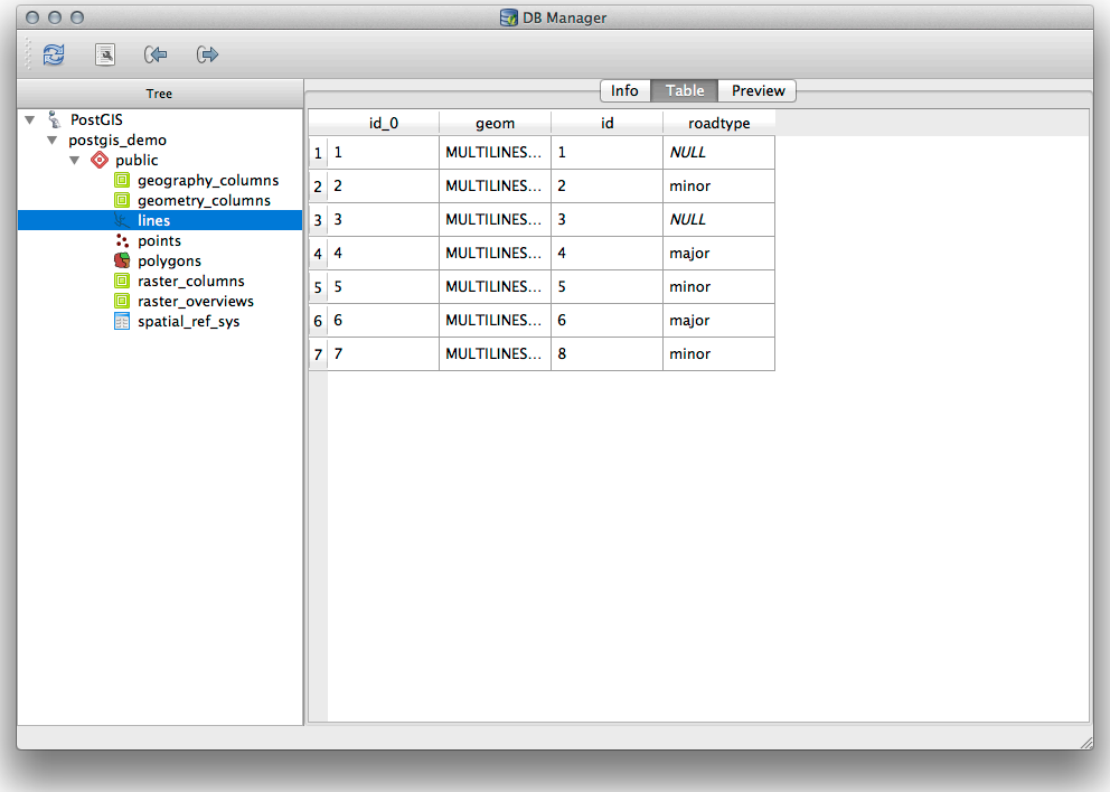


In diesem Bedienfeld sehen wie die *Allgemeinen Informationen* über die Tabelle als auch die Informationen über die Geometrie und das räumliche Bezugssystem, die von der PostGIS Erweiterung bereitgestellt werden.

Wenn man im *Info* Reiter weiter nach unten scrollt, sieht man weitere Informationen zu *Felder*, *Constraints* und *Indexes* bezüglich der aktuellen Tabelle.



Der DB-Manager ist außerdem sehr nützlich, um sich Datensätze anzusehen - ähnlich der Attributtabelle eines Layers im Layerbaum. Sie können die Daten im Reiter *Tabelle* durchsehen.

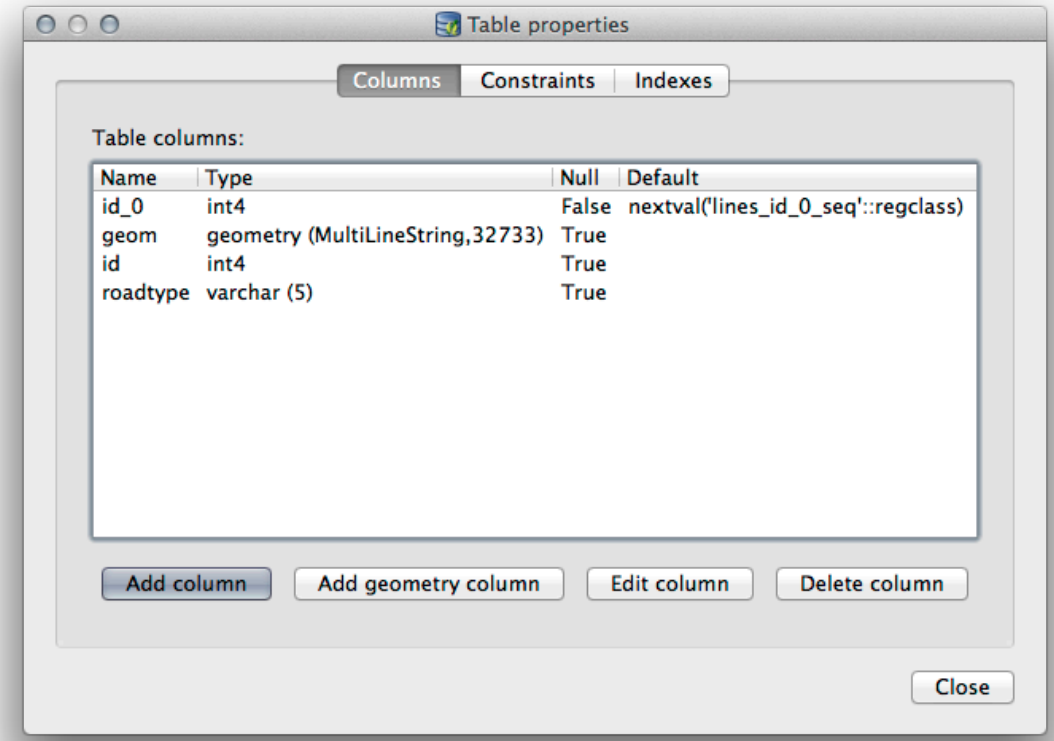


Es gibt auch einen *Vorschau* Reiter, der die Layerdaten als Kartenvorschau zeigt.

Mit Rechtsklick auf den Layer im Baum und Klick auf *Zur Karte hinzufügen* wird der Layer in Ihre Karte eingefügt.

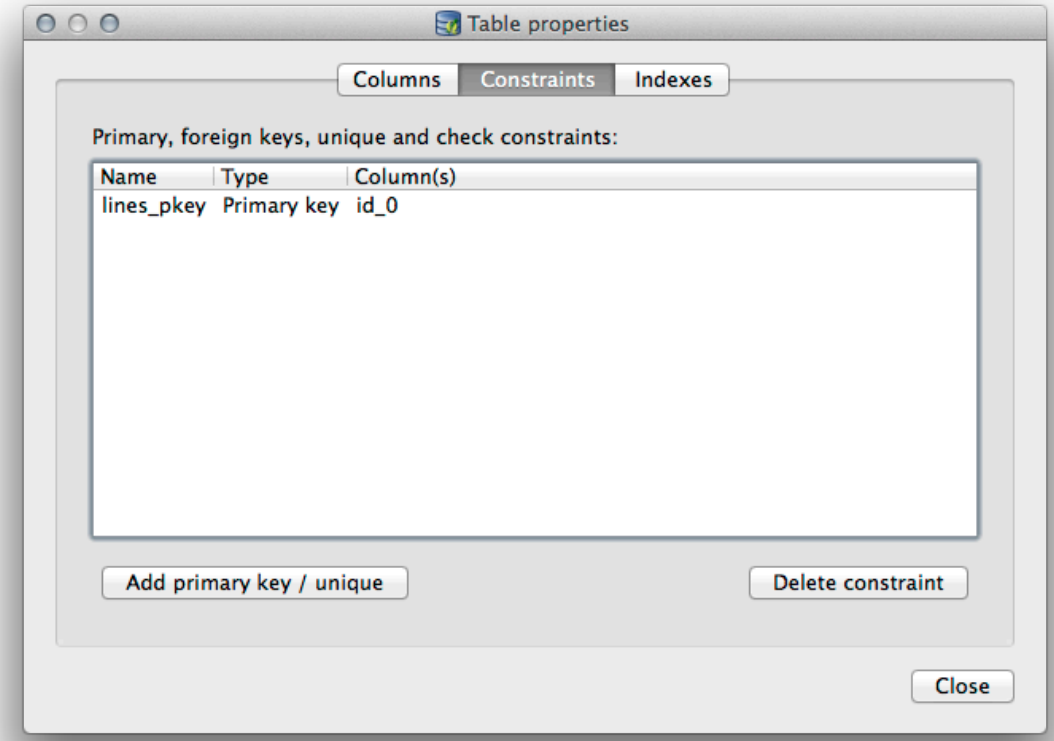
Bis hierher haben wir nur die Datenbank, ihre Schemas, Tabellen und Metadaten angezeigt. Was ist nun, wenn man Tabellen ändern möchte und z.B. eine neue Spalte hinzufügen möchte? Die DB-Verwaltung erlaubt uns, dass direkt zu machen.

1. Wählen Sie die Tabelle, die sie bearbeiten möchten im Baum
2. Select *Table* ► *Edit Table* from the menu, to open the *Table Properties* dialog.

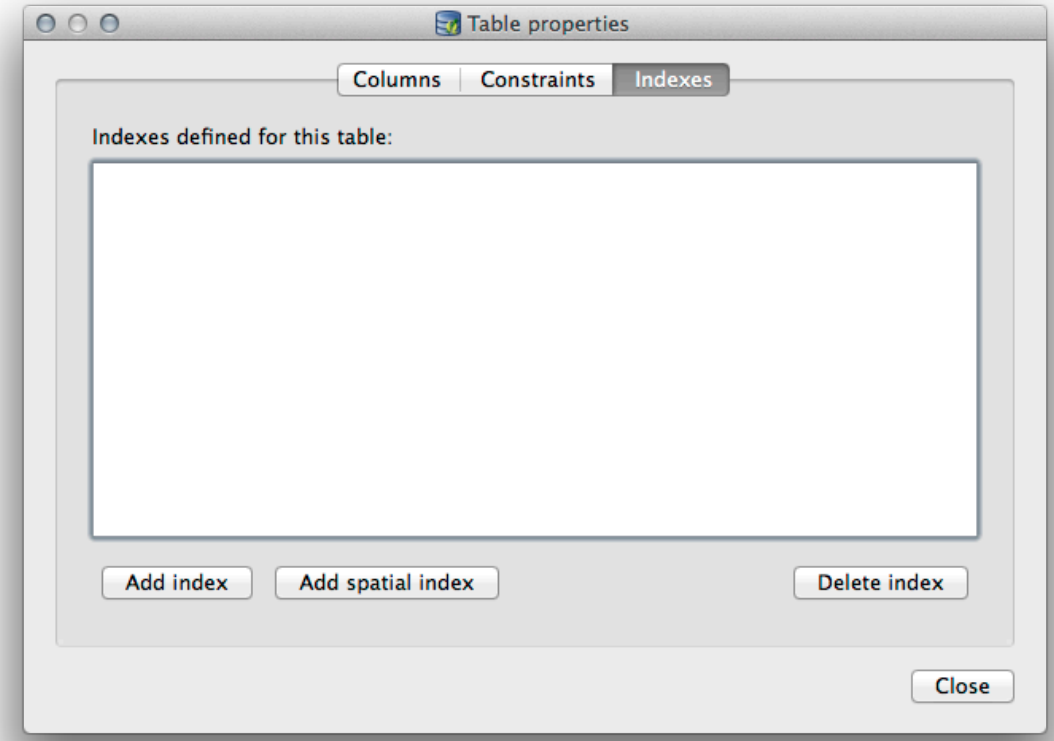


Sie können der Dialog verwenden, um Spalten oder Geometriespalten hinzuzufügen, vorhandene Spalten zu bearbeiten oder Spalten komplett zu entfernen.

Mit Hilfe des Reiters *Restriktionen* können Sie verwalten, welche Felder als Primärschlüssel verwendet werden oder um vorhandene Einschränkungen zu entfernen.



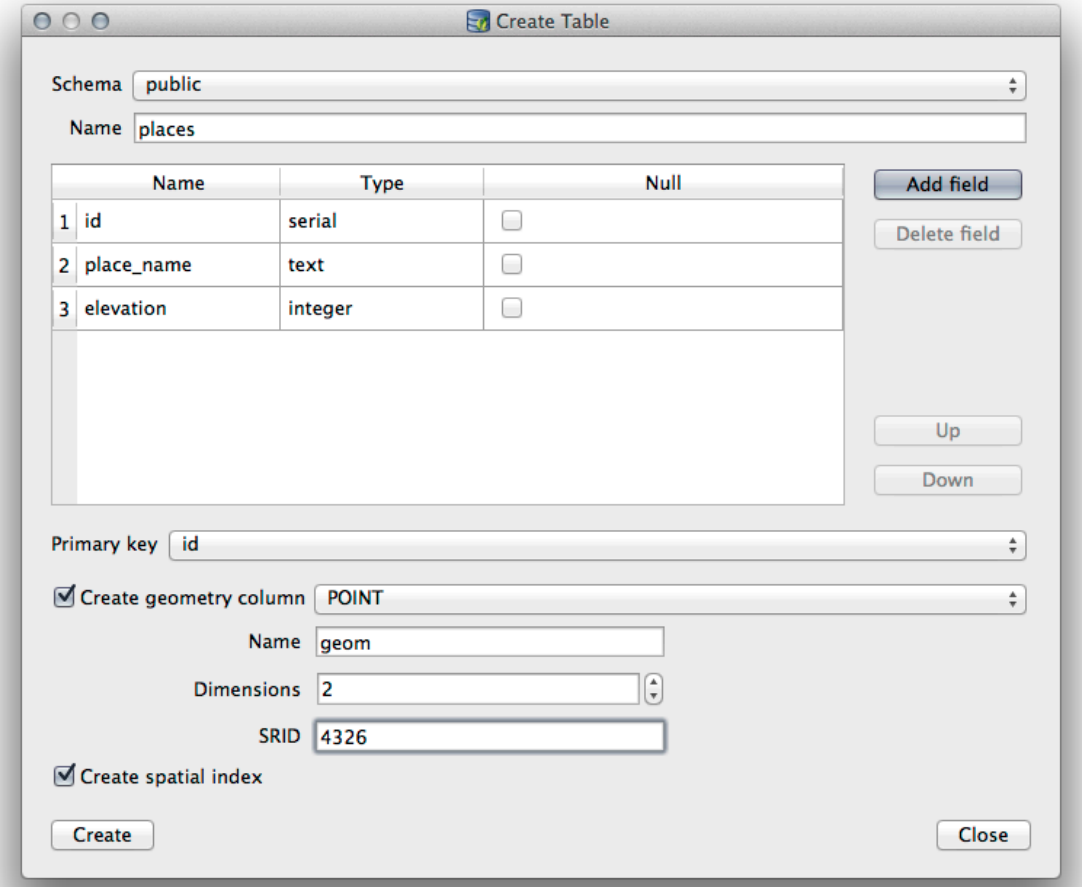
Im Reiter *Indizes* können normale oder räumliche Indizes hinzugefügt oder gelöscht werden.



18.2.2 Follow Along: Erstellen einer neuen Tabelle

Nachdem wir den Prozess zur Arbeit mit vorhandenen Tabellen durchgegangen sind, nutzen wir die DB-Verwaltung jetzt zur Erstellung einer neuen Tabelle.

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. Wählen Sie *Tabelle –> Tabelle erzeugen* um den Dialog zu öffnen.
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. Setzen Sie den Haken bei *Räumlichen Index erzeugen* und klicken dann auf *Erzeugen*, um die Tabelle zu erstellen.



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: Einfache Datenbankadministration

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.

Datenbanktabellen können oftmals sehr groß werden und oft veränderte Tabellen können Überbleibsel zurücklassen, die nicht mehr von PostgreSQL benötigt werden. Der Befehl *VACUUM* stellet eine Art Müllabfuhr dar. Er verringert die Größe der Tabellen. Bei Bedarf analysiert er die Tabellen, um eine höhere Geschwindigkeit der Datenbank zu erreichen.

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.

Weitere Informationen zum Prozess VACUUM ANALYZE finden Sie unter PostgreSQL Documentation on VACUUM ANALYZE.

18.2.4 Follow Along: Executing SQL Queries with DB Manager

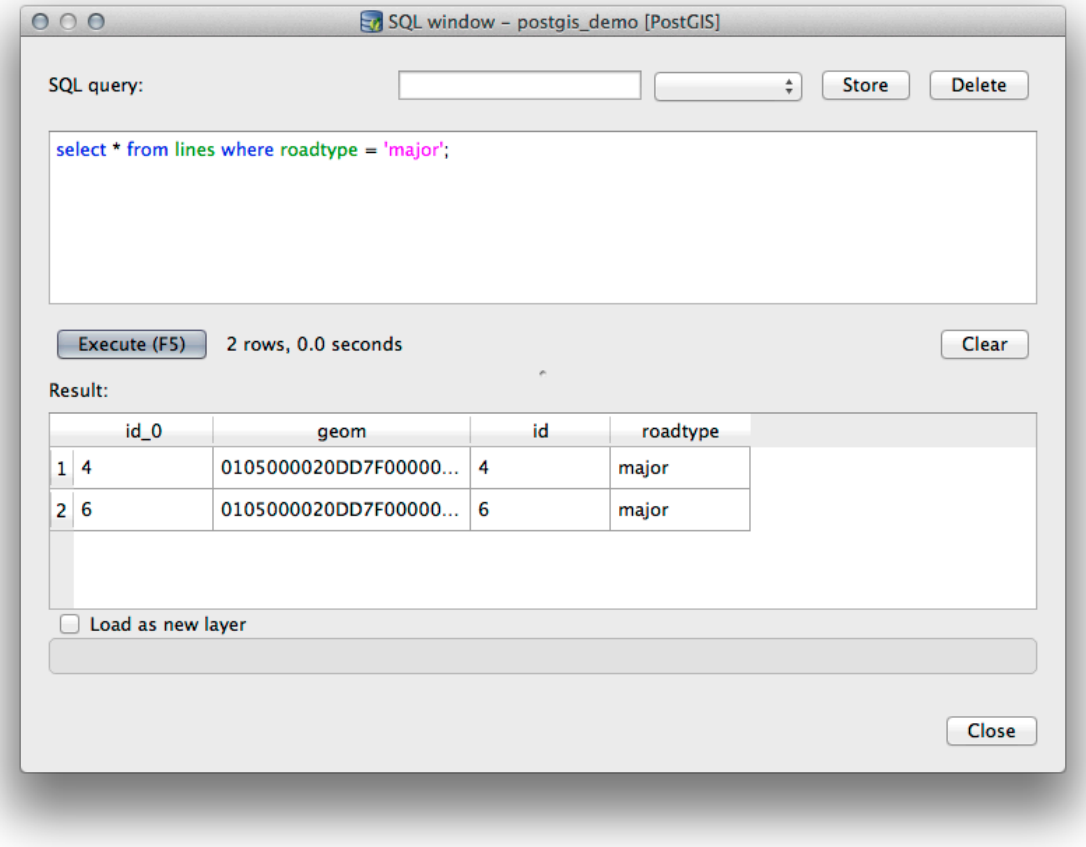
DB Manager also provides a way for you to write queries against your database tables and to view the results. We have already seen this type of functionality in the *Browser* panel, but lets look at it again here with DB Manager.

1. Select the `lines` table in the tree.
2. Select the *SQL window* button in the DB Manager toolbar.

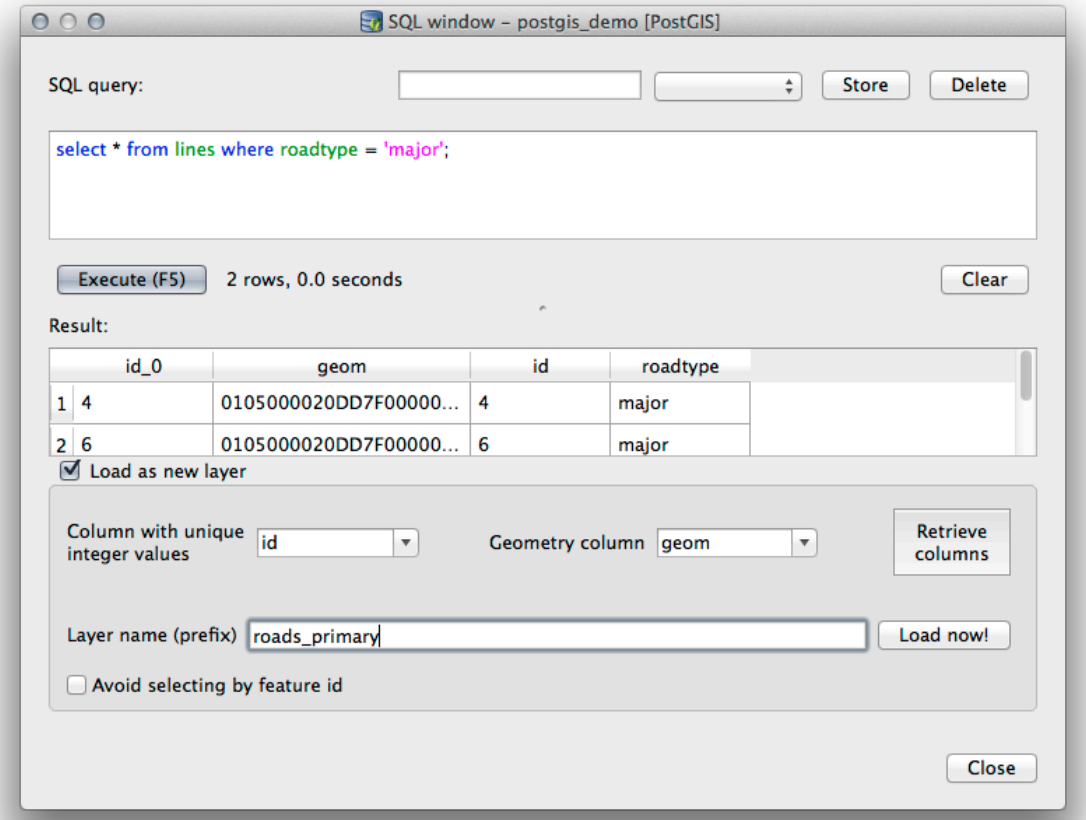
3. Compose the following *SQL query* in the space provided:

```
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.



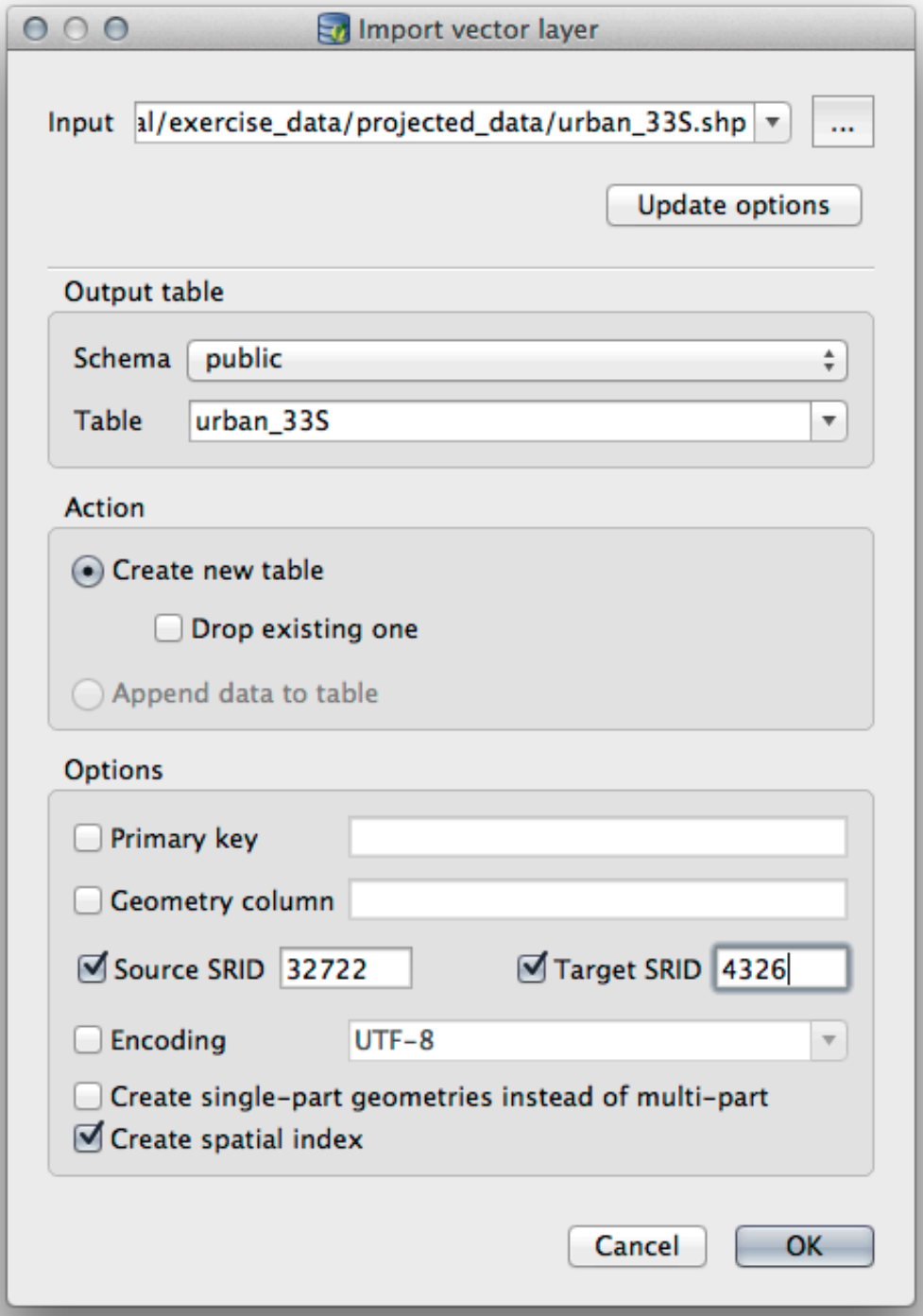
The layers that matched your query are now displayed on your map. You can of course use this query tool to execute any arbitrary SQL command including many of the ones we looked at in previous modules and sections.

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.

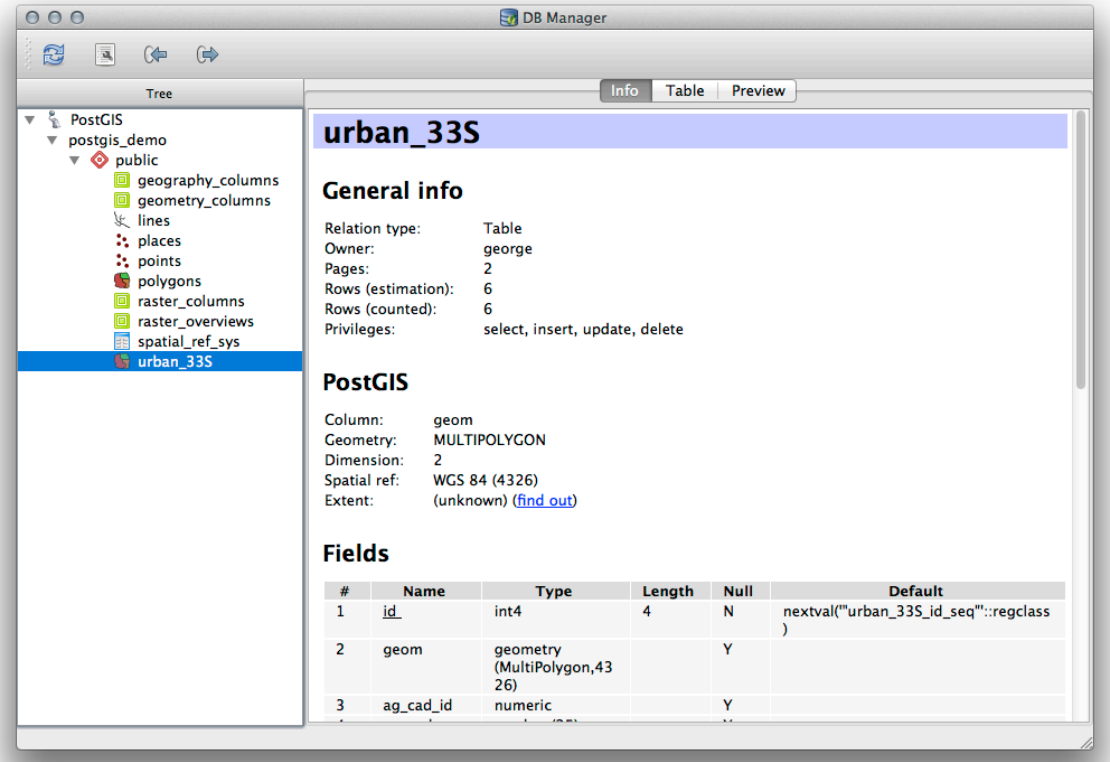
1. Click the *Import layer/file* button on the toolbar in the DB Manager dialog.

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)`.

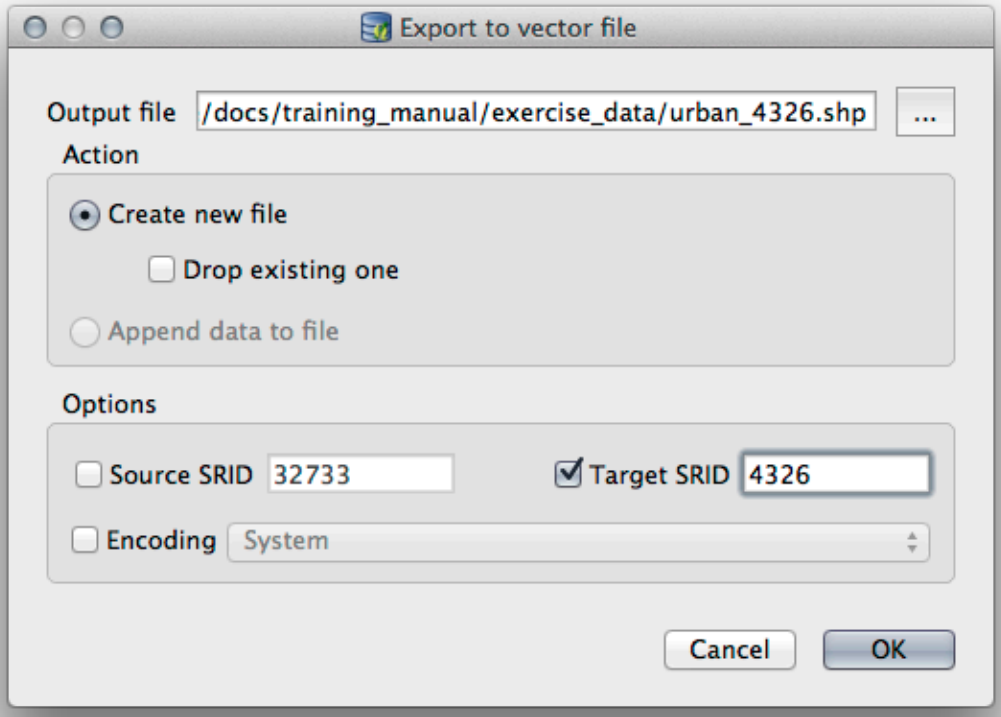


Right clicking on the table in the Tree and a selecting *Add to Canvas* will add the table as a layer in your map.

18.2.6 Exporting Data from a Database with DB Manager

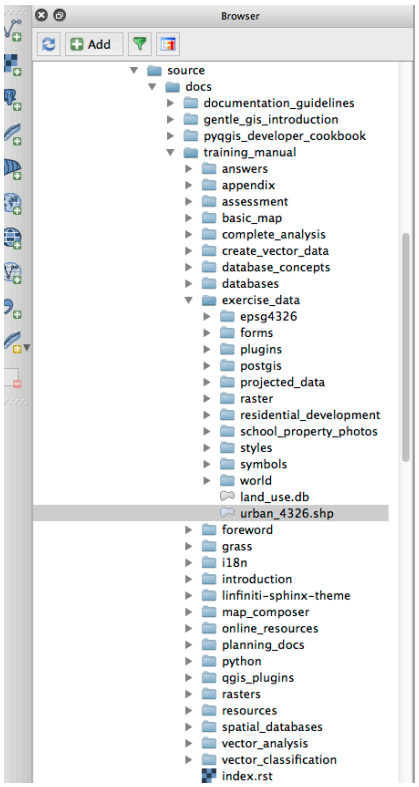
Of course DB Manager can also be used to export data from your spatial databases, so lets take a look at how that is done.

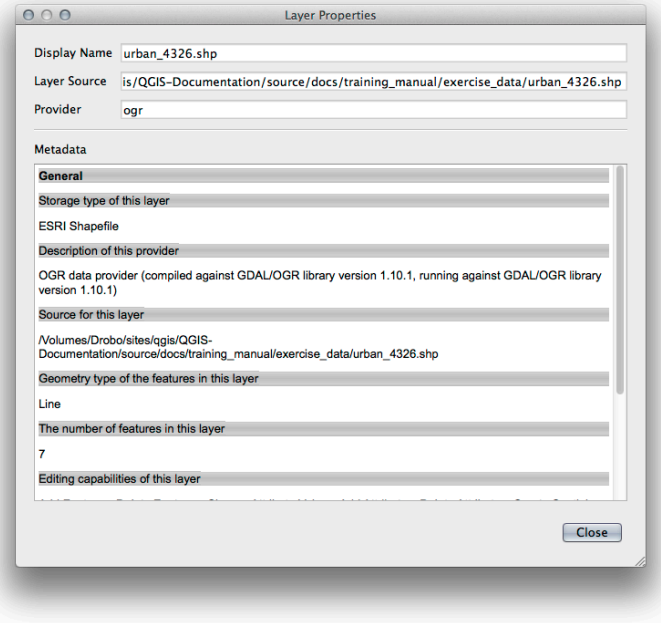
1. Select the `lines` layer in the Tree and click the *Export to File* button on the toolbar to open the *Export to vector file* dialog.
2. Click the … button to select the *Output file* and save the data to your `exercise_data` directory as `urban_4326`.
3. Set the *Target SRID* as `4326`.
4. Click *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.

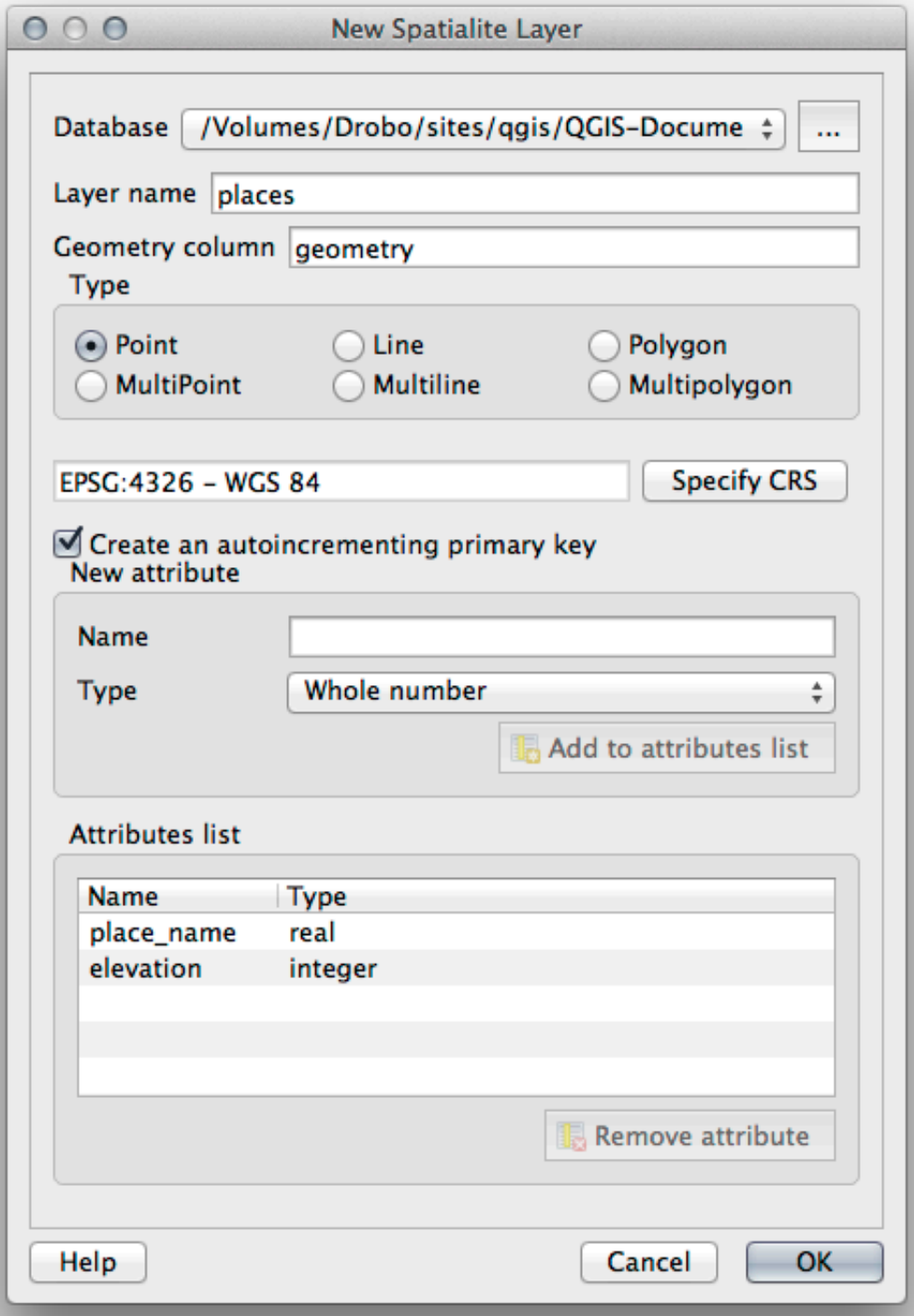
18.3.1 Follow Along: Creating a SpatiaLite database with the Browser

Using the Browser panel, we can create a new SpatiaLite database and get it setup for use in QGIS.

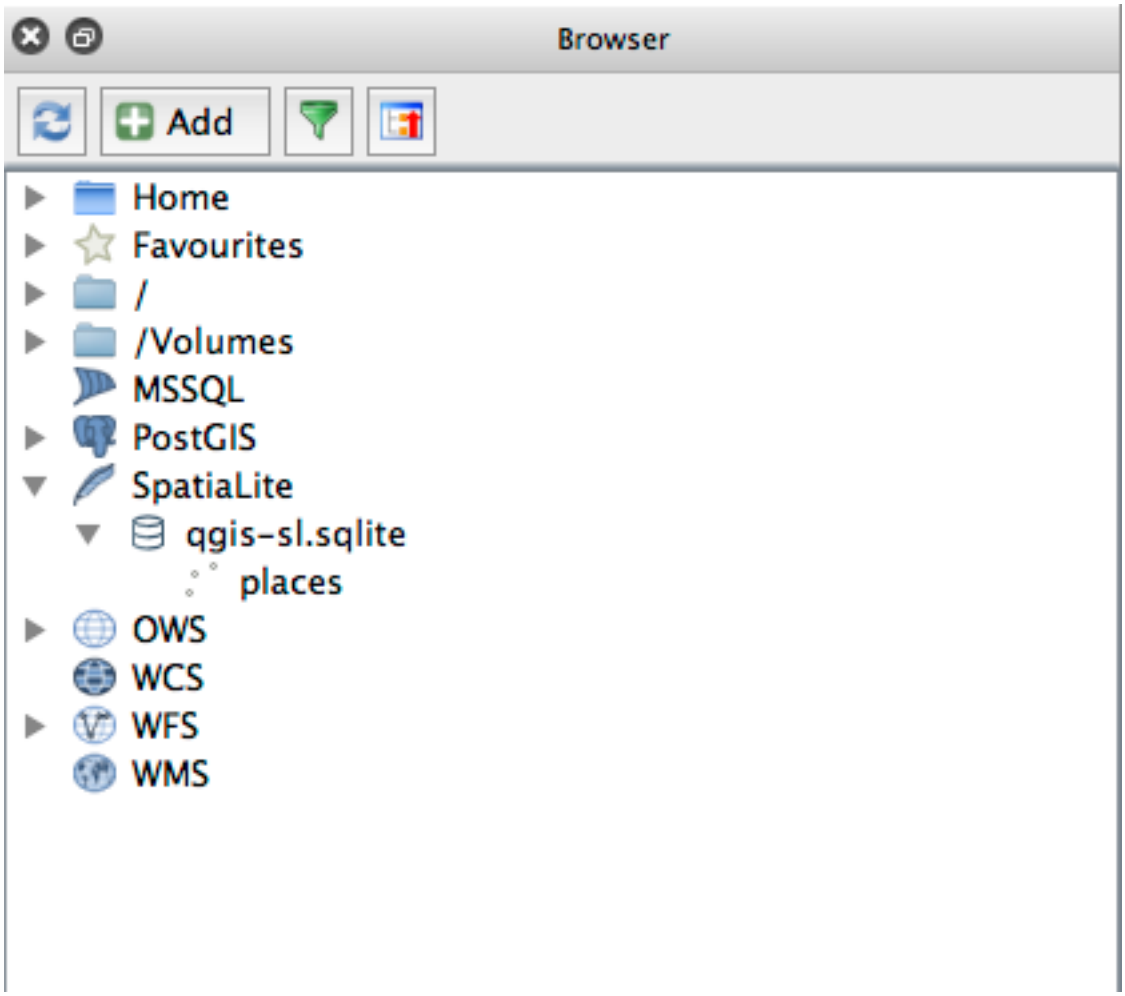
1. Right click on the *SpatiaLite* entry in the Browser tree and select *Create Database*.
2. Specify where on your filesystem you want to store the file and name it `qgis-sl.db`.
3. Again right click on the *SpatiaLite* entry in the Browser tree and now select the *New Connection* item. Find the file 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.

KAPITEL 19

Anhang: Mitwirkende an diesem Handbuch

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 Ressourcen herunterladen

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 Hinzufügen eines Moduls

- 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`.

Bemerkung: 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.

KAPITEL 20

Vorbereitung der Übungsdaten

Bemerkung: This process is intended for course conveners, or more experienced QGIS users who wish to create localised sample data sets for their course. Default data sets are provided with the Training Manual, but you may follow these instructions if you wish to replace the default data sets.

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.

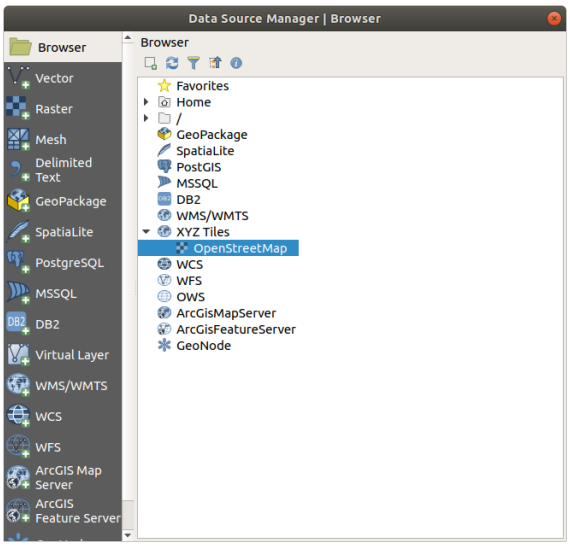
Anyone can use this dataset without difficulty, but you may prefer to use data from your own country or home town. If you choose to do so, your localised data will be used in all lessons from Module 3 to Module 7.2. Later modules use more complex data sources which may or may not be available for your region.

Bemerkung: These instructions assume you have a good knowledge of QGIS and are not intended to be used as teaching material.

20.1 Try Yourself Create OSM based vector files

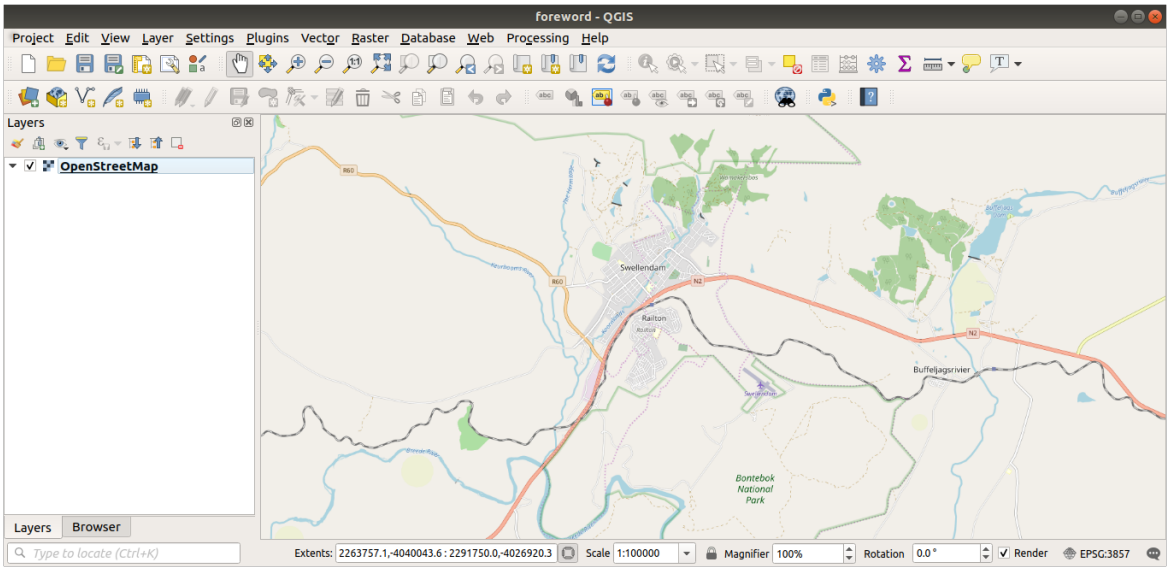
If you wish to replace the default data set with localised data for your course, this can easily be done with tools built into QGIS. The region you choose to use should have a good mix of urban and rural areas, containing roads of differing significance, area boundaries (such as nature reserves or farms) and surface water, such as streams and rivers.

1. Öffne ein neues QGIS-Projekt
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.



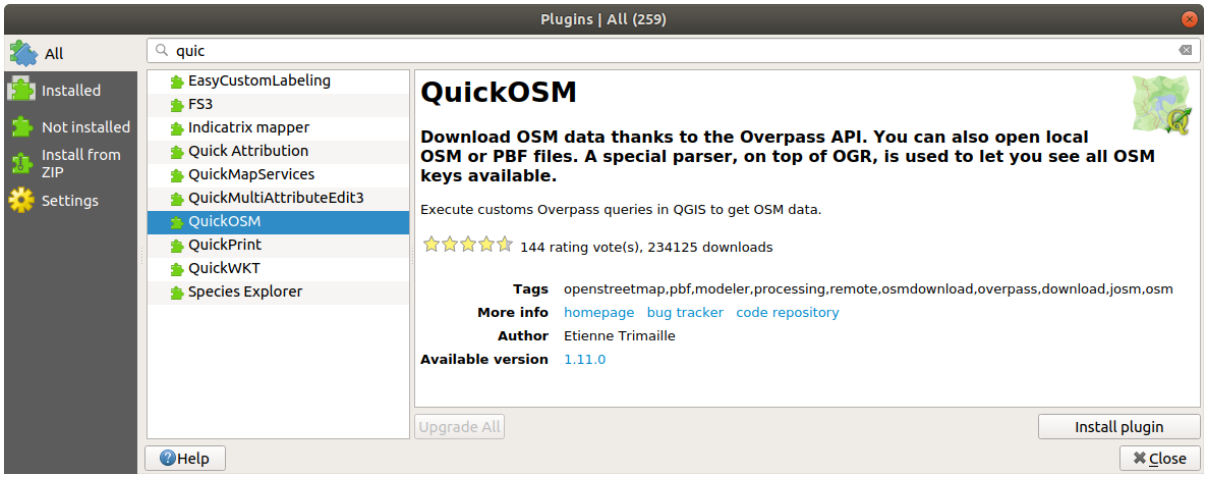
A map of the world is now visible on the map canvas.

4. Close the *Data Source Manager* dialog
5. Move to the area you'd like to use as study area

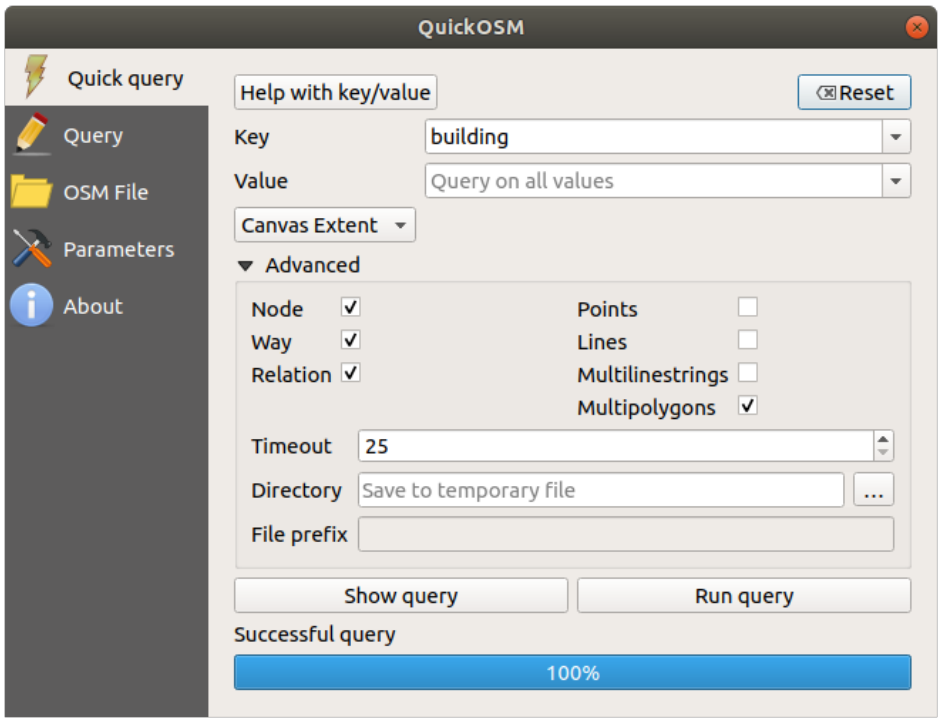


Now that we have the area we'll extract the data from, let's enable the extraction tools.

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*

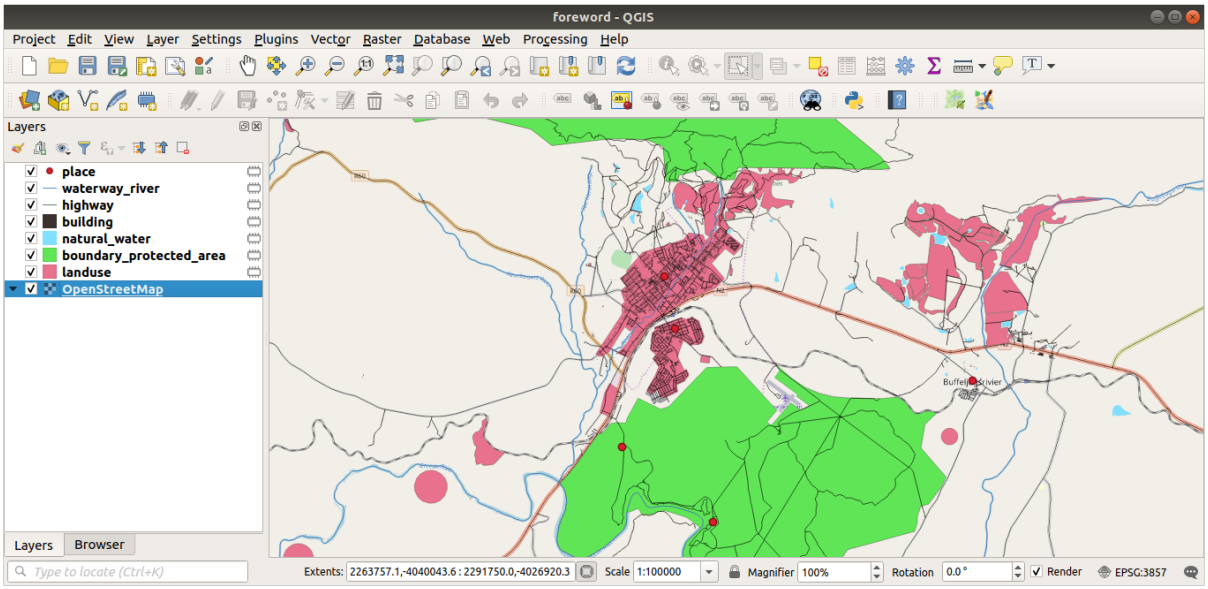


A new `building` layer is added to the *Layers* panel, showing buildings in the selected extent.

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.

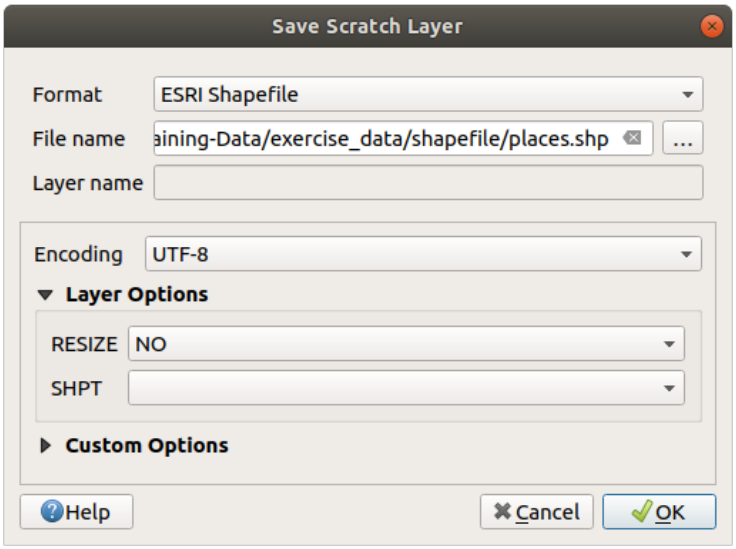
We now need to save the resulting data to use during your course. We'll be using ESRI Shapefile, GeoPackage and SpatiaLite formats depending on the data.

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.

   Bemerkung: 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. Klicken Sie auf *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.

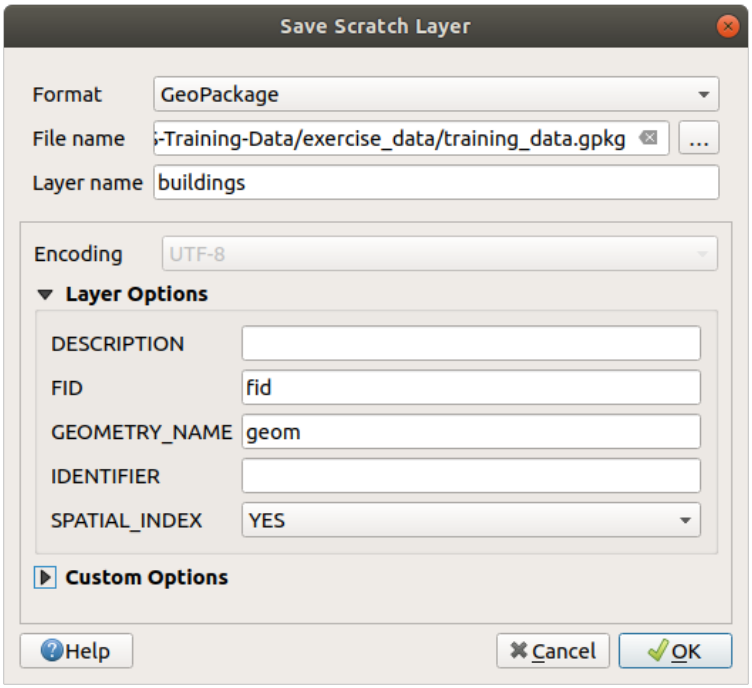
6. Repeat the process for other layers, renaming them as follows:

   - `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:

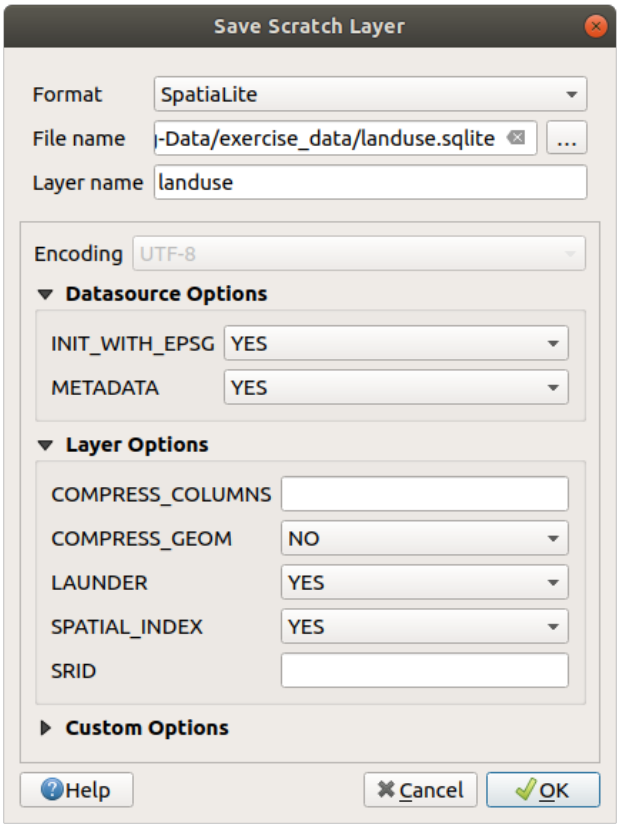
1. Click the icon next to the *building* layer
2. Select the *GeoPackage* format
3. Save the file as `training_data.gpkg` under the `exercise_data/` folder
4. By default, the *Layer name* is filled as the file name. Replace it with `buildings`.



5. Klicken Sie auf *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.

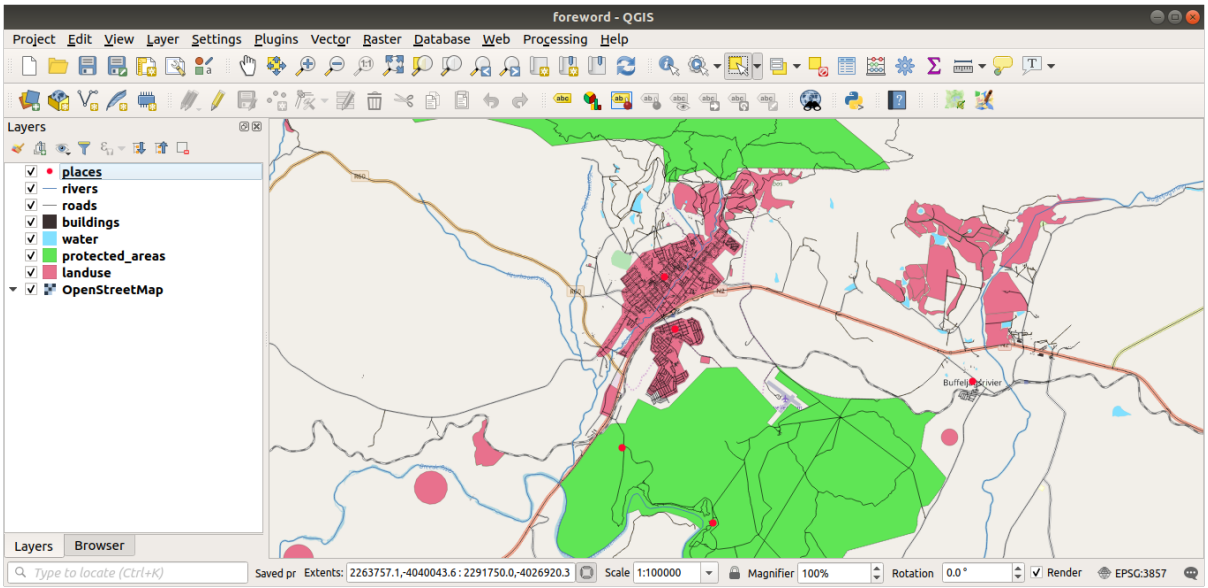
The last step is to save the remaining temporary file as a SpatiaLite file.

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. Klicken Sie auf *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 Create SRTM DEM tiff files

For modules *Module: Vektordaten erzeugen* and *Module: Raster*, 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: Vektordaten erzeugen*, *Follow Along: Datenquellen* 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 Replace tokens

Having created your localized dataset, the final step is to replace the tokens in the `substitutions.txt` file so that the appropriate names will appear in your localized version of the Training Manual.

The tokens you need to replace are as follows:

- `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.