Practical Introduction to QGIS

by Karsten Vennemann

TERRA GIS
TERRESTRIAL ENVIRONMENT REGIONAL ANALYSIS

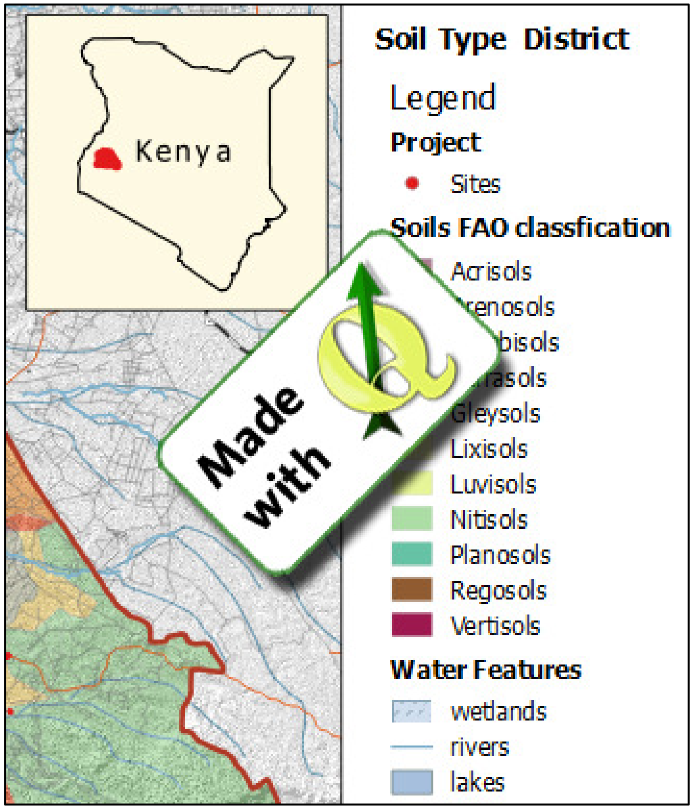
Live Online Session each day
8:30 am – 4:30 pm PST (UTC-7 h)

1 hour lunch break + two 15 minute breaks

Day 1	Tuesday	May 26th 2020
Day 2	Wednesday	May 27th 2020
Day 3	Thursday	May 28th 2020

QGIS

a very capable and flexible Desktop GIS



Karsten Vennemann

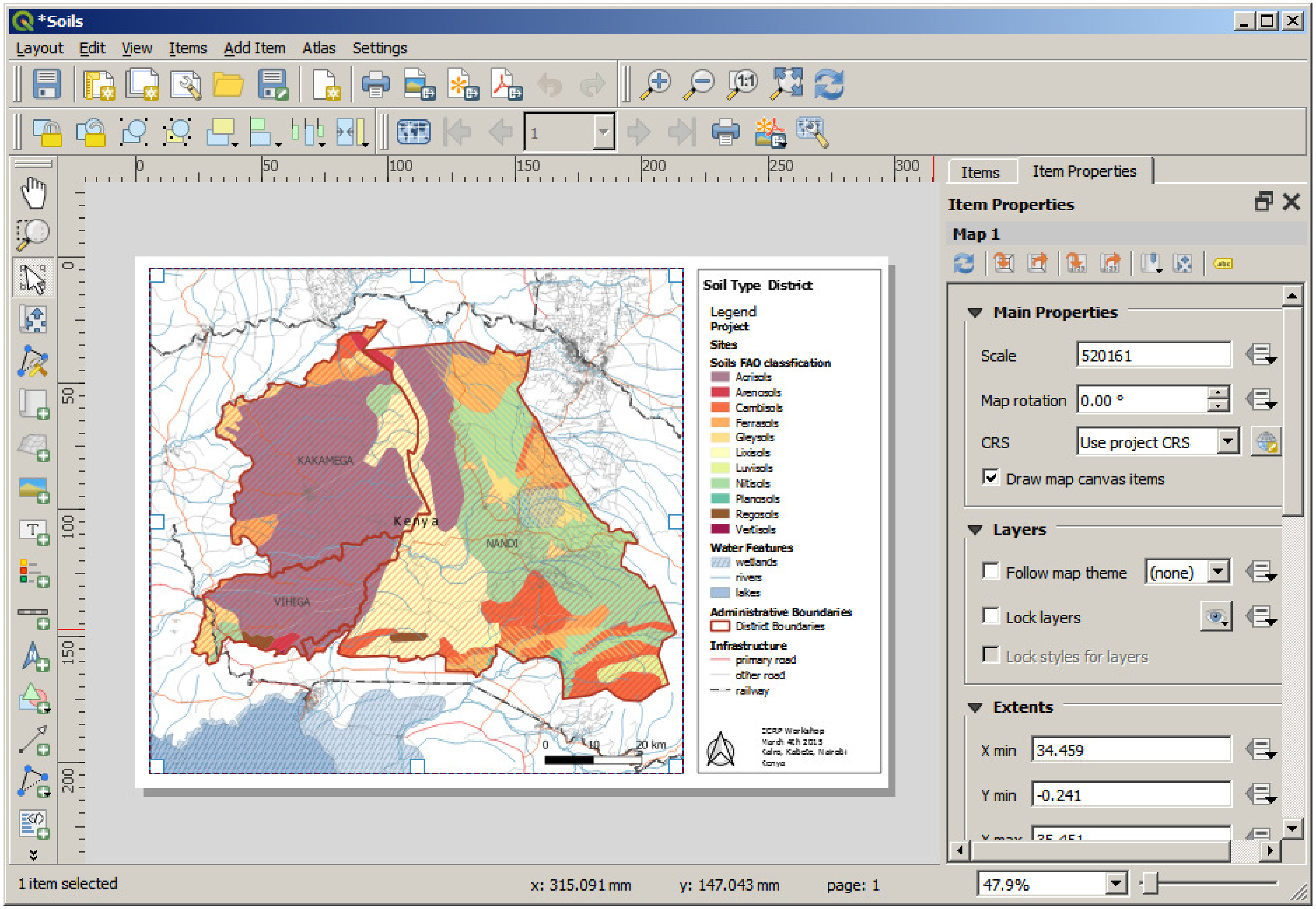
TERRA GIS
TERRESTRIAL ENVIRONMENT REGIONAL ANALYSIS

Overview and Foundations of QGIS

- The QGIS project and it's open source community

QGIS - Desktop GIS

originally a GIS viewing environment QGIS for the Linux desktop but is available for Solaris, Windows and Mac. Support for many data Formats



Item Properties

Map 1

Main Properties

Scale: 520161

Map rotation: 0.00 °

CRS: Use project CRS

☑ Draw map canvas items

Layers

☐ Follow map theme: (none)

☐ Lock layers

☐ Lock styles for layers

Extents

X min: 34.459

Y min: -0.241

X max: 35.451

1 item selected   x: 315.091 mm   y: 147.043 mm   page: 1   47.9%

QGIS Facts

Main supporter of **Quantum GIS**	Gary Sherman and others
Type	Desktop GIS Viewer
Functionality	Can be used as a UI to GRASS GIS with GRASS Plug-in, Python bindings allow for programmatic interaction
Operating systems	Multi platform
Project started	2002
Implementation	C++, Depends on QT widget
OS libraries	OGR/GDAL
PostGIS support	Yes
License	GPL

QGIS Highlights

- „Intermediate“ Desktop GIS
- all basic and intermediate GIS Functionality
- support for many input formats
- easily extensible and highly customizable
- extended comprehensive Analytic capabilities
  -> Processing Tools and modeler
- automation and custom tool development via
  - Python scripting (Python bindings and pyQT integration)
  - Enables plug-in and user interface development
- Very active User and Developer Community
  - rapid development, good community support

What is Open Source (GIS)?

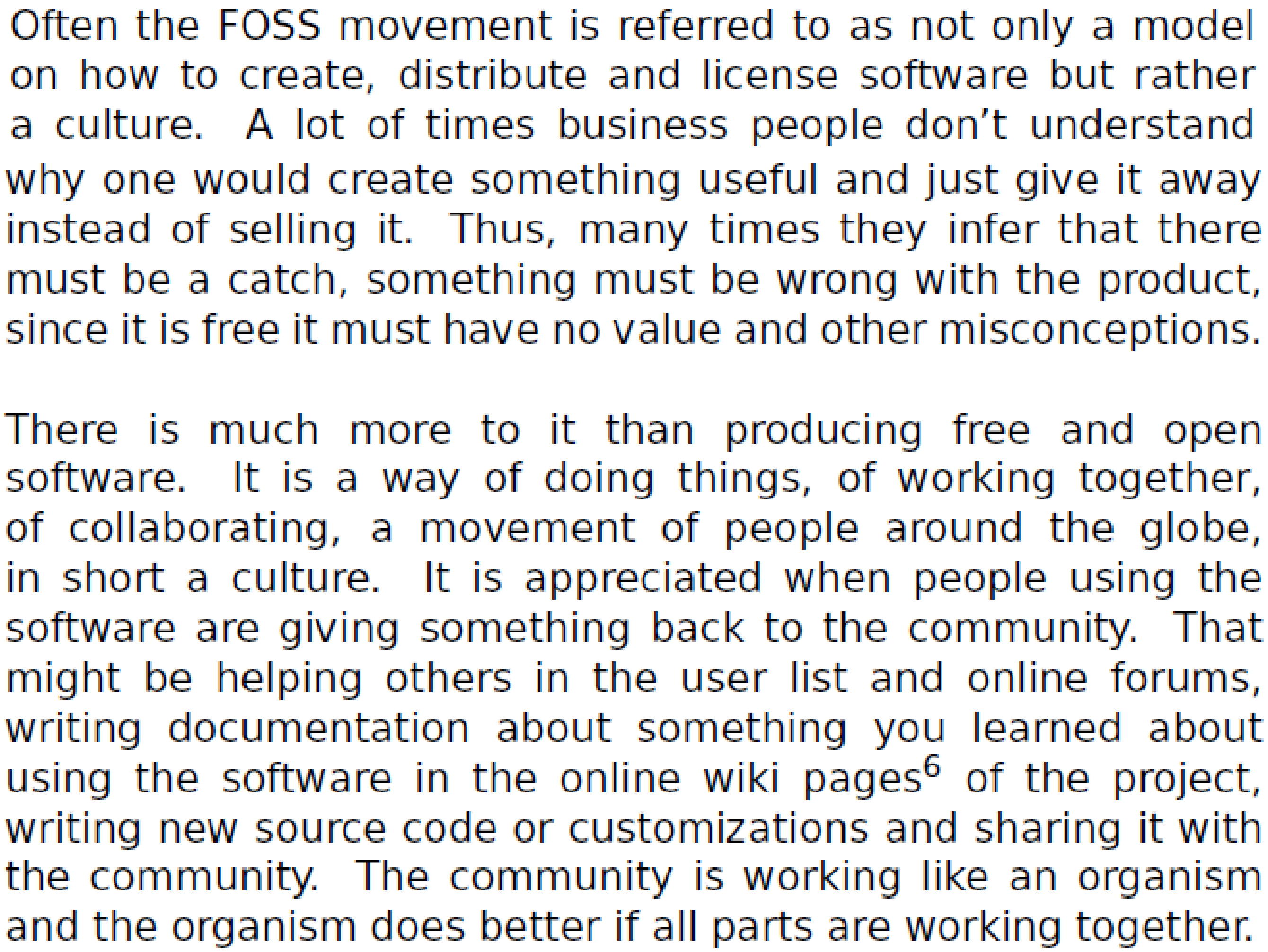
Open source means that the source code is available to the general public for use, distribution, and modification from its original design free of charge (among a long list of other requirements)

Open Source ≠ Open Standards

While most open source geospatial software is built on the standards of the Open Geospatial Consortium (OGC) the term “Open Source” it is not synonymous with Open Standards because both proprietary and open source software can be compliant with the OGC Open Standards. http://www.opengeospatial.org

OSGeo is the organization that supports the development of the highest quality open source geospatial software. http://www.osgeo.org

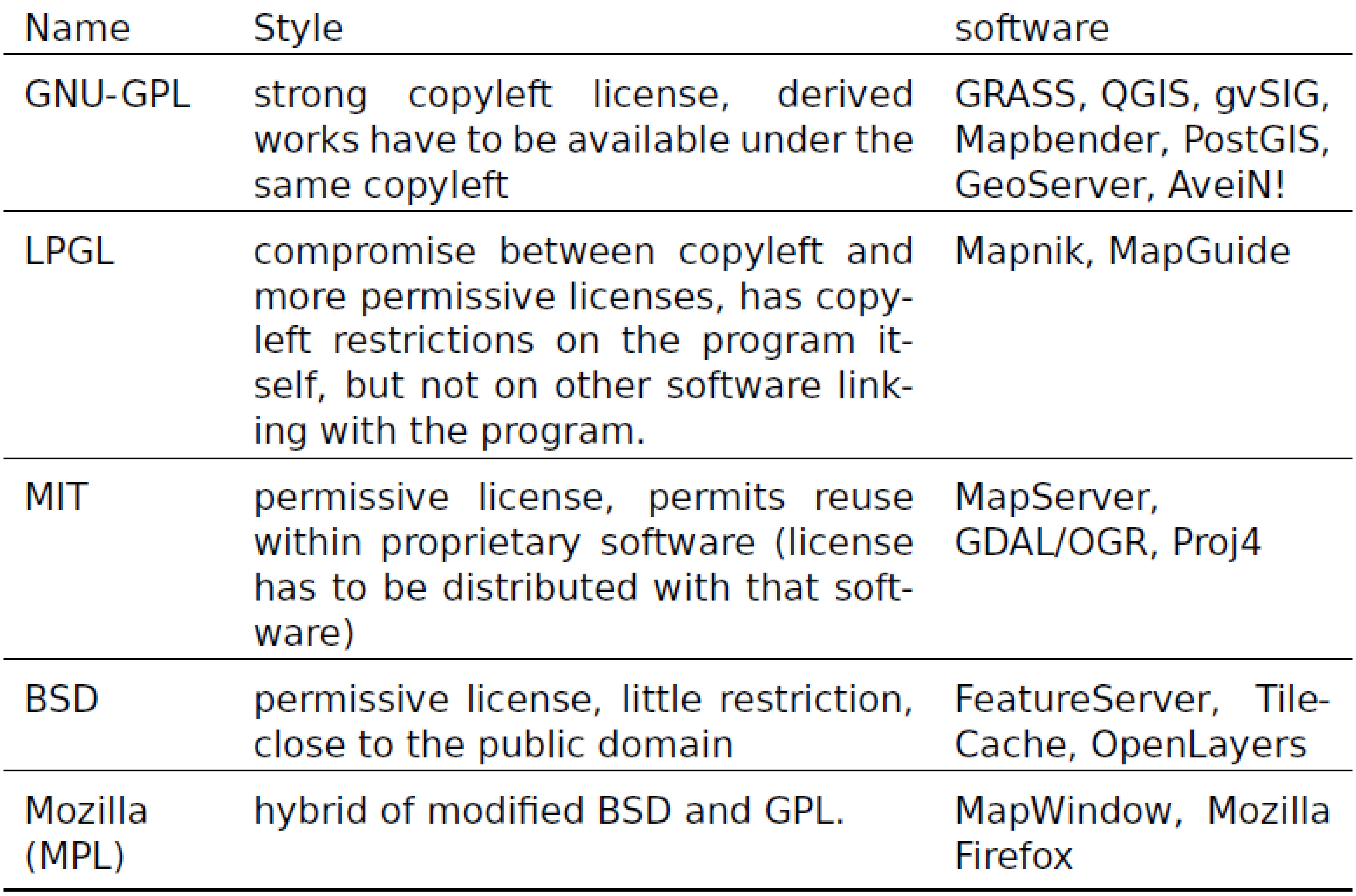
The OS Culture



Often the FOSS movement is referred to as not only a model on how to create, distribute and license software but rather a culture. A lot of times business people don't understand why one would create something useful and just give it away instead of selling it. Thus, many times they infer that there must be a catch, something must be wrong with the product, since it is free it must have no value and other misconceptions.

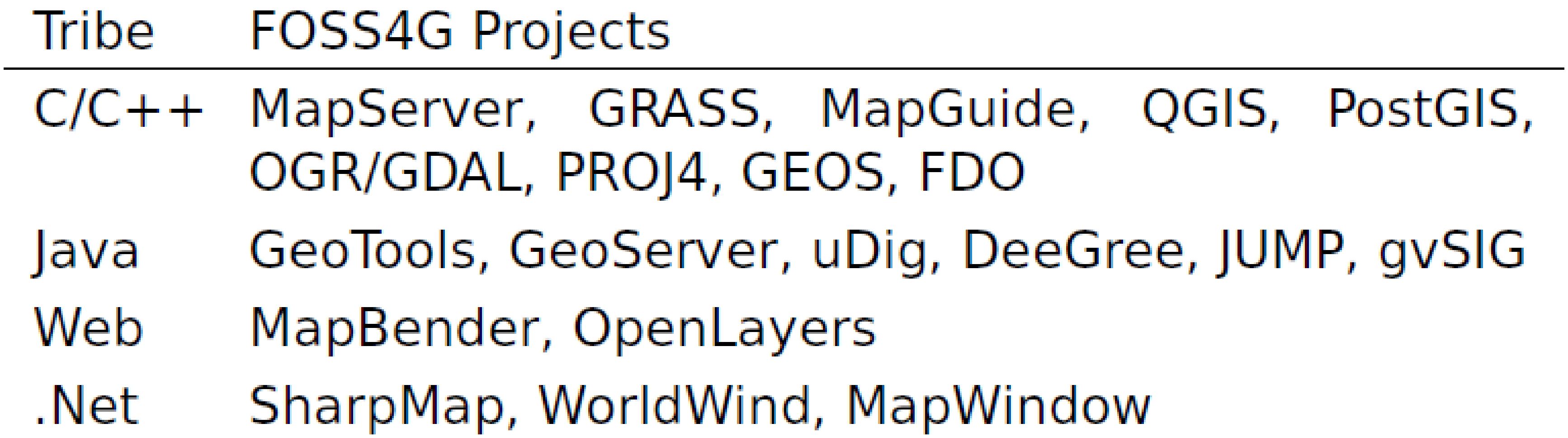
There is much more to it than producing free and open software. It is a way of doing things, of working together, of collaborating, a movement of people around the globe, in short a culture. It is appreciated when people using the software are giving something back to the community. That might be helping others in the user list and online forums, writing documentation about something you learned about using the software in the online wiki pages[6] of the project, writing new source code or customizations and sharing it with the community. The community is working like an organism and the organism does better if all parts are working together.

List of common FOSS software licenses



Name	Style	software
GNU-GPL	strong copyleft license, derived works have to be available under the same copyleft	GRASS, QGIS, gvSIG, Mapbender, PostGIS, GeoServer, AveiN!
LPGL	compromise between copyleft and more permissive licenses, has copyleft restrictions on the program itself, but not on other software linking with the program.	Mapnik, MapGuide
MIT	permissive license, permits reuse within proprietary software (license has to be distributed with that software)	MapServer, GDAL/OGR, Proj4
BSD	permissive license, little restriction, close to the public domain	FeatureServer, TileCache, OpenLayers
Mozilla (MPL)	hybrid of modified BSD and GPL.	MapWindow, Mozilla Firefox

The „Tribes“ of FOSS4G



Tribe	FOSS4G Projects
C/C++	MapServer, GRASS, MapGuide, QGIS, PostGIS, OGR/GDAL, PROJ4, GEOS, FDO
Java	GeoTools, GeoServer, uDig, DeeGree, JUMP, gvSIG
Web	MapBender, OpenLayers
.Net	SharpMap, WorldWind, MapWindow

Some of the Foundations of OS Software (Tools)

A few libraries that are the foundation of many
Open Source and commercial Geospatial Software Packages

- GDAL (Raster) and OGR (Vector)
  Geospatial Data Abstraction Library / Open GIS Simple Features Reference Implementation
  - Tools for reading, writing and processing of raster and vector data sets -> formats
  - Important base for many Desktop GIS systems e.g. ArcGIS
  - OGR extends Mapserver formats
    Oracle Spatial, ESRI Geodatabase (MDB), TIGER, MapInfo…
- PROJ4 is a library for cartographic projection routines
  - stand alone projection utility “proj”
  - libraries for more than 2500 projections (e.g. EPSG list)
- GeoTools is an open source Java GIS toolkit is a library for cartographic projection routines
  - Similar usage as OGR and GDAL for Java based projects
  - Udig and GeoServer are based on GeoTools

Examples for practical use of GDAL/OGR

- **Raster / Image processing**
  - run automatically from server side scripts on server bash shell
  - image mosaicing, reprojection
  - custom scripts to process 3 band tiff images e.g. vegetation vigor classification (Landsat 7+ 8)
  - assemble *synthetic* map images , grayscale for background + color classified raster map

Practical Introduction

The QGIS software and spatial data

- QGIS Desktop, additional software
- Spatial Data
- Data Sources Examples

volunteer efforts	Open Street Map
	Natural Earth Data
	GADM data
data portals e.g.	USGS Earth Explorer

Practical Introduction

The map document

open the Kenya Map

data/kenya/kenya_exercises.qgz

Using workshop maps and data

- Tour of the Interface
  including but not limited to
  - Adding data
  - Changing cartographic style/ representation
  - Displaying and arranging map layers
  - Attribute tables and indentifying attributes
  - Feature selections and filters
  - Data conversions / export
  - Using tool bars

Quick Tour

- **Menu Bar** provides access to all the main functions and plugins
- **Toolbars** provide one-click common functions, and task-specific functions
- **Layer List** shows all data layers currently added to the project
- **Status Bar** provides some vital information about the current project settings
- **Python Console** allows you to script Python code within QGIS
- **Map View** provides a dynamic visualization of the *active* data layers that can be mapped

QGIS Basics and Interface Overview

- Supported Data Formats
- Exploring and using vector and raster data
- Layer + map properties
- Symbology / Cartography

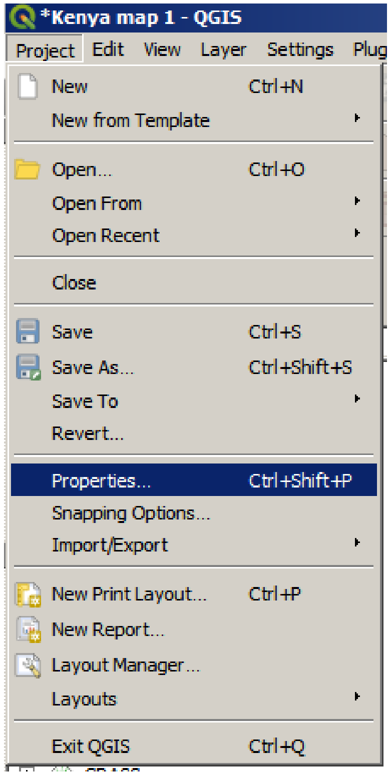
Exercises - Basic Functionalities

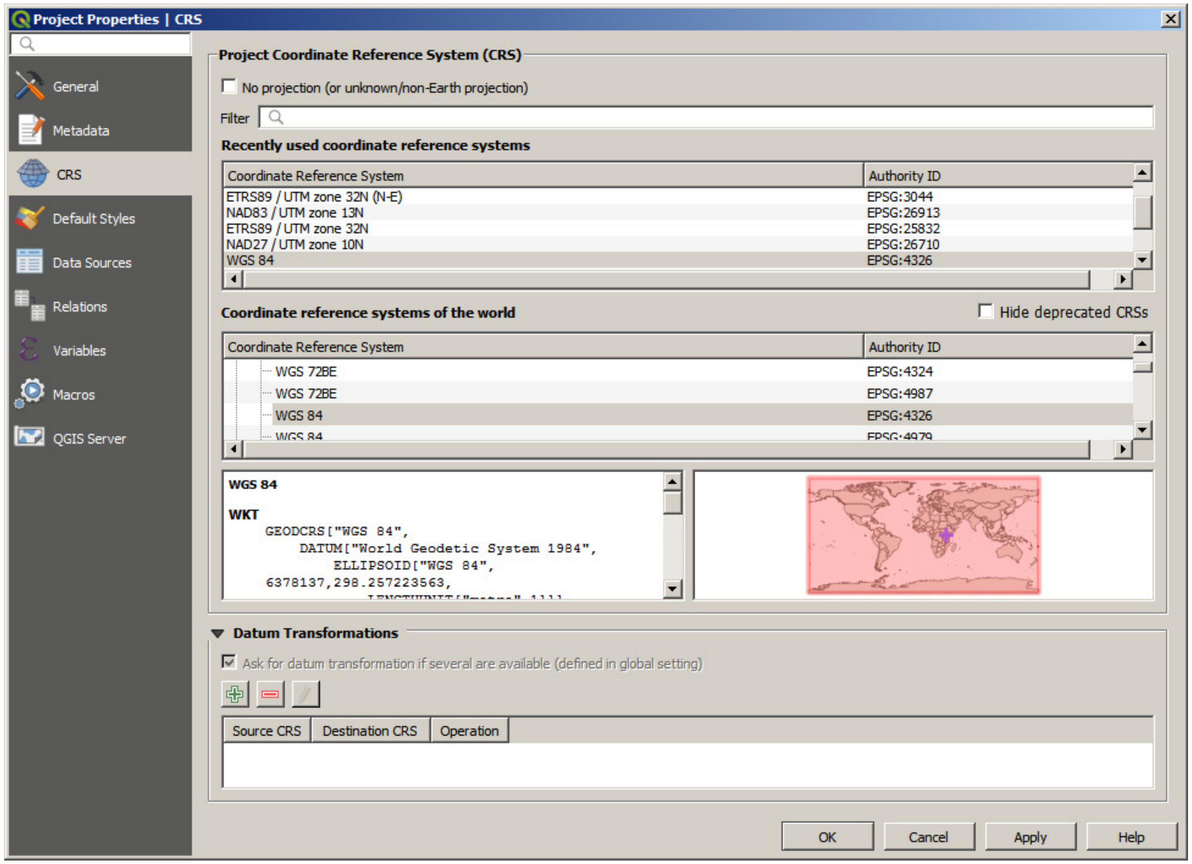
Using workshop maps and data

Settings
Properties and Options - user interface (Task 1)

CRS (Task 1.1)

Menu -> Project -> Properties





□ No projection (or unknown/non-Earth projection)

Coordinate Reference System	Authority ID
ETRS89 / UTM zone 32N (N-E)	EPSG:3044
NAD83 / UTM zone 13N	EPSG:26913
ETRS89 / UTM zone 32N	EPSG:25832
NAD27 / UTM zone 10N	EPSG:26710
WGS 84	EPSG:4326

□ Hide deprecated CRSs

Coordinate Reference System	Authority ID
WGS 72BE	EPSG:4324
WGS 72BE	EPSG:4987
WGS 84	EPSG:4326
WGS 84	EPSG:4979

WGS 84

WKT

```
    GEODCRS["WGS 84",
        DATUM["World Geodetic System 1984",
            ELLIPSOID["WGS 84",
    6378137,298.257223563,
```

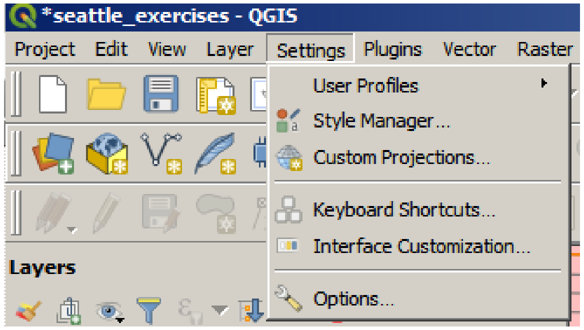
☑ Ask for datum transformation if several are available (defined in global setting)

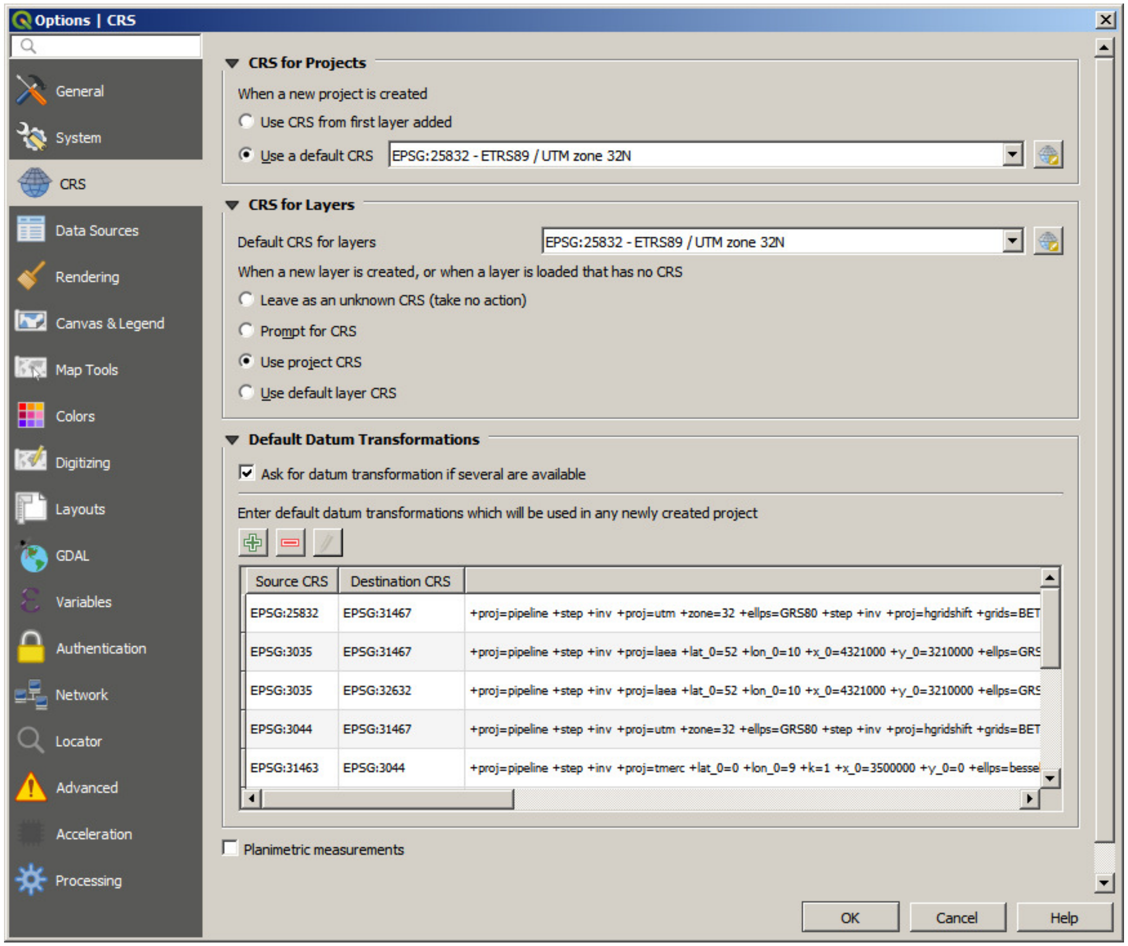
Source CRS	Destination CRS	Operation

Settings
Properties and Options - user interface (Task 1)

- Review Options (Task 1.2)

Menu -> Settings -> Options





CRS for Projects

When a new project is created

○ Use CRS from first layer added

● Use a default CRS: EPSG:25832 - ETRS89 / UTM zone 32N

CRS for Layers

Default CRS for layers: EPSG:25832 - ETRS89 / UTM zone 32N

When a new layer is created, or when a layer is loaded that has no CRS

○ Leave as an unknown CRS (take no action)

○ Prompt for CRS

● Use project CRS

○ Use default layer CRS

Default Datum Transformations

☑ Ask for datum transformation if several are available

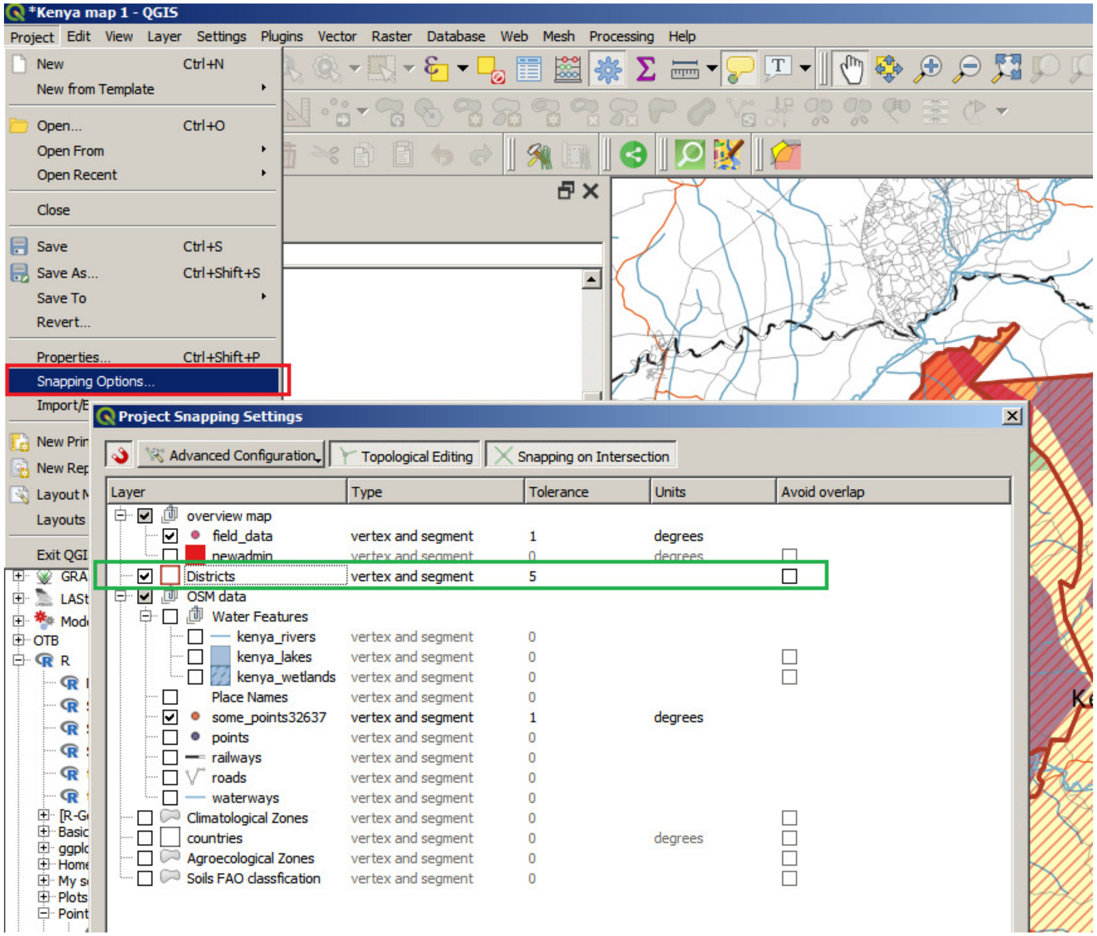
Enter default datum transformations which will be used in any newly created project

Source CRS	Destination CRS	
EPSG:25832	EPSG:31467	+proj=pipeline +step +inv +proj=utm +zone=32 +ellps=GRS80 +step +inv +proj=hgridshift +grids=BET
EPSG:3035	EPSG:31467	+proj=pipeline +step +inv +proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS
EPSG:3035	EPSG:32632	+proj=pipeline +step +inv +proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS
EPSG:3044	EPSG:31467	+proj=pipeline +step +inv +proj=utm +zone=32 +ellps=GRS80 +step +inv +proj=hgridshift +grids=BET
EPSG:31463	EPSG:3044	+proj=pipeline +step +inv +proj=tmerc +lat_0=0 +lon_0=9 +k=1 +x_0=3500000 +y_0=0 +ellps=bessel

□ Planimetric measurements

Settings - Options - user interface (Task 1)

- Digitizing (Task 1.3)
  - enable and define snapping behavior

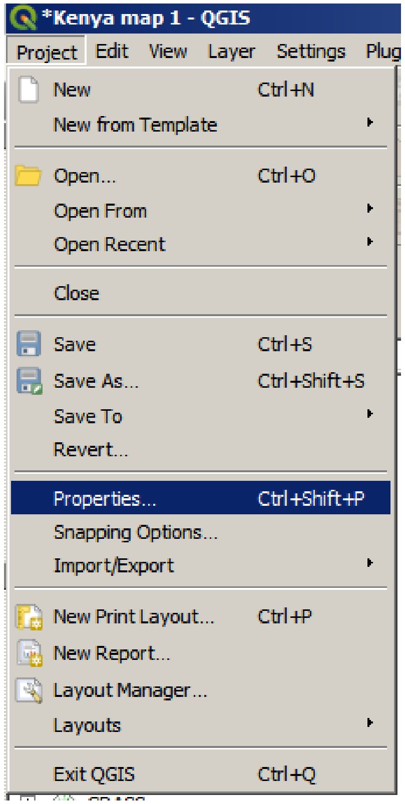


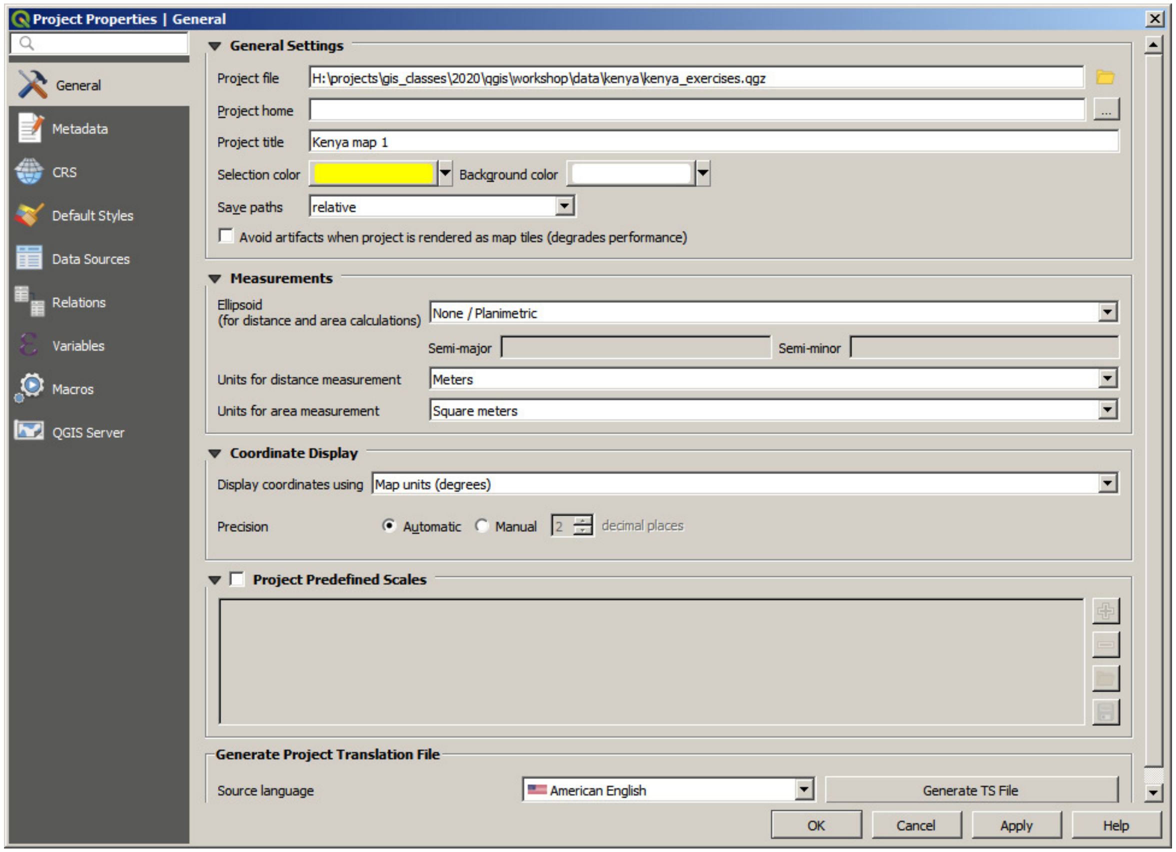
Project Snapping Settings

Layer	Type	Tolerance	Units	Avoid overlap
☑ overview map				
☑ field_data	vertex and segment	1	degrees	
☐ newadmin	vertex and segment	0	degrees	☐
☑ Districts	vertex and segment	5		☐
☑ OSM data				
☐ Water Features				
☐ kenya_rivers	vertex and segment	0		
☐ kenya_lakes	vertex and segment	0		☐
☐ kenya_wetlands	vertex and segment	0		☐
☐ Place Names	vertex and segment	0		
☑ some_points32637	vertex and segment	1	degrees	
☐ points	vertex and segment	0		
☐ railways	vertex and segment	0		
☐ roads	vertex and segment	0		
☐ waterways	vertex and segment	0		
☐ Climatological Zones	vertex and segment	0		☐
☐ countries	vertex and segment	0	degrees	☐
☐ Agroecological Zones	vertex and segment	0		☐
☐ Soils FAO classfication	vertex and segment	0		☐

Settings - Options - user interface (Task 1)

- Canvas and Legend (Task 1.4)
  - Set “Save path” to relative
  - change selection color and more





Project file: H:\projects\gis_classes\2020\qgis\workshop\data\kenya\kenya_exercises.qgz

Project home:

Project title: Kenya map 1

Selection color: 🟨 Background color: □

Save paths: relative

□ Avoid artifacts when project is rendered as map tiles (degrades performance)

Ellipsoid (for distance and area calculations): None / Planimetric

Semi-major: Semi-minor:

Units for distance measurement: Meters

Units for area measurement: Square meters

Display coordinates using: Map units (degrees)

Precision: ● Automatic ○ Manual 2 decimal places

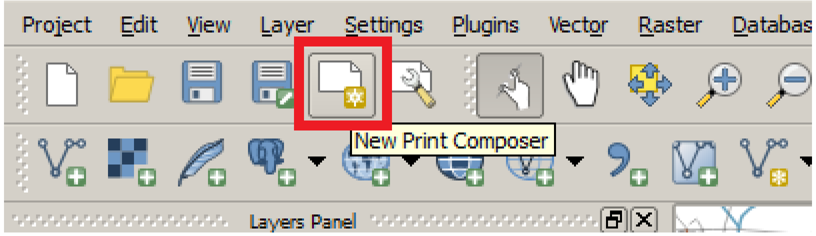
□ Project Predefined Scales

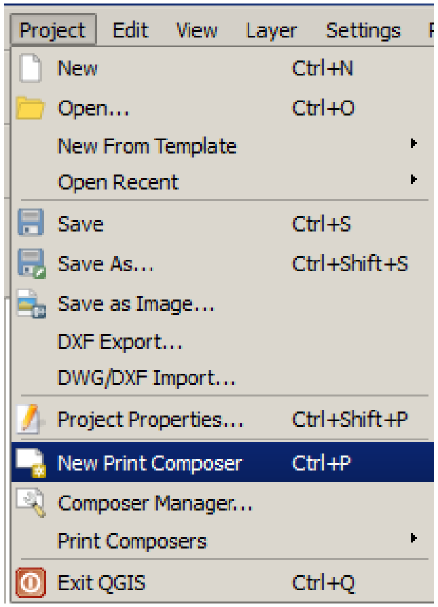
Source language: American English

Exercises - Creating Maps

Creating Maps (Task 2)

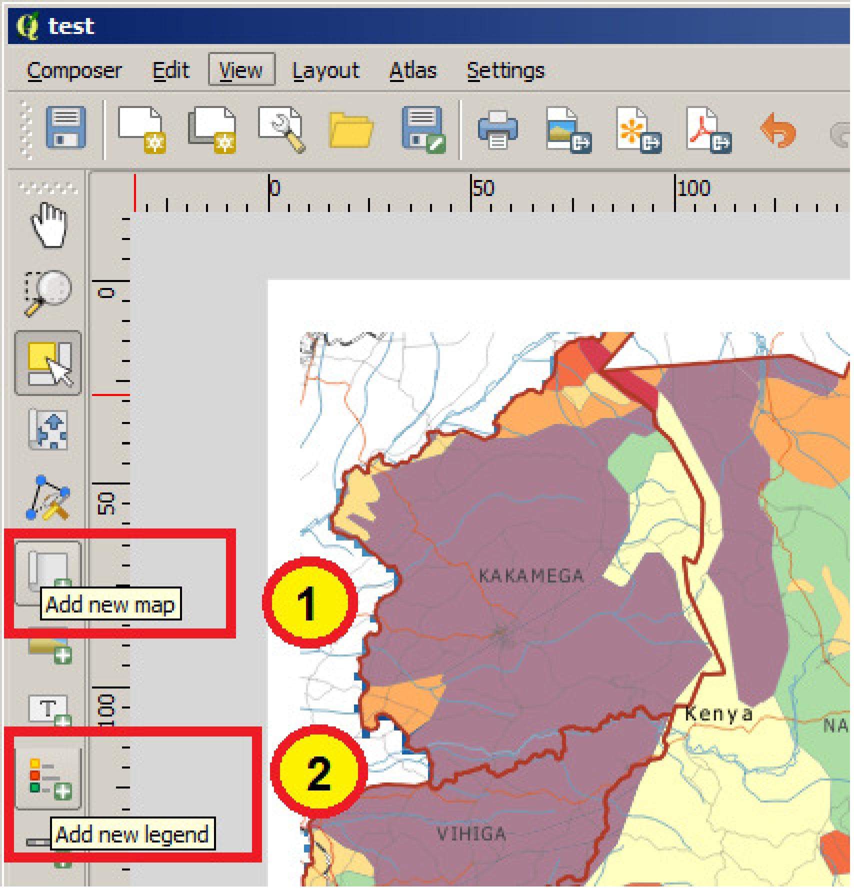
- create a new, empty project (Task 2.1)
- add at least three vector layers (Task 2.2)
  - add two Seattle data set layers
  - open layer properties – style
  - change color and outline (use “single symbol” for one layer and “categorized” for the other layers as options)
  - note the advanced option of rule based classification
- create a new “print composer” (layout) (Task 2.3)





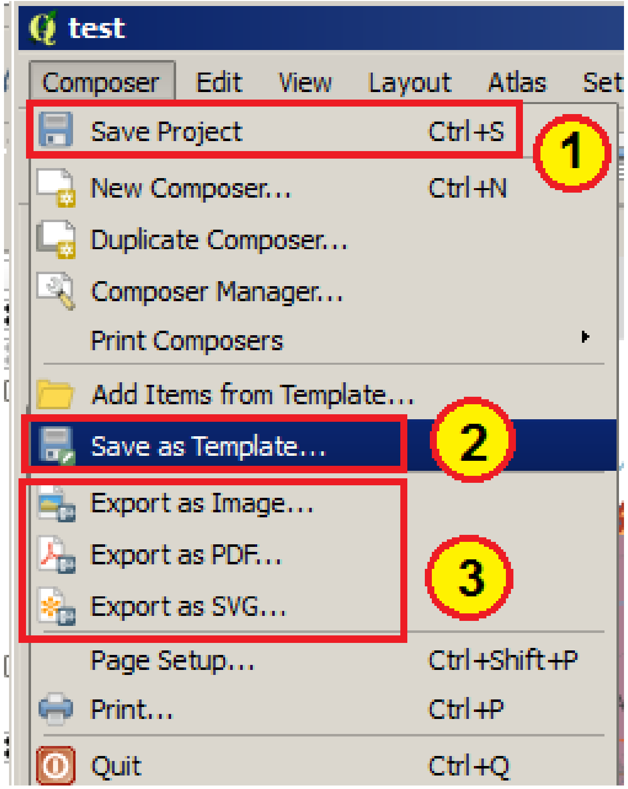
Creating Maps (Task 2.4)

- add map item (Task 2.4.1)
- add legend item (Task 2.4.2)



Creating Maps (Task 2.4)

- review options in print composer (Task 2.4.3)
  - add scale bar map item
  - add north arrow
- save project (*.qgz) (Task 2.4.4)
- save work as template (*.qgt) (Task 2.4.5)
- export map as image (Task 2.4.6)
- close print composer and create new project from your template (Task 2.4.7)

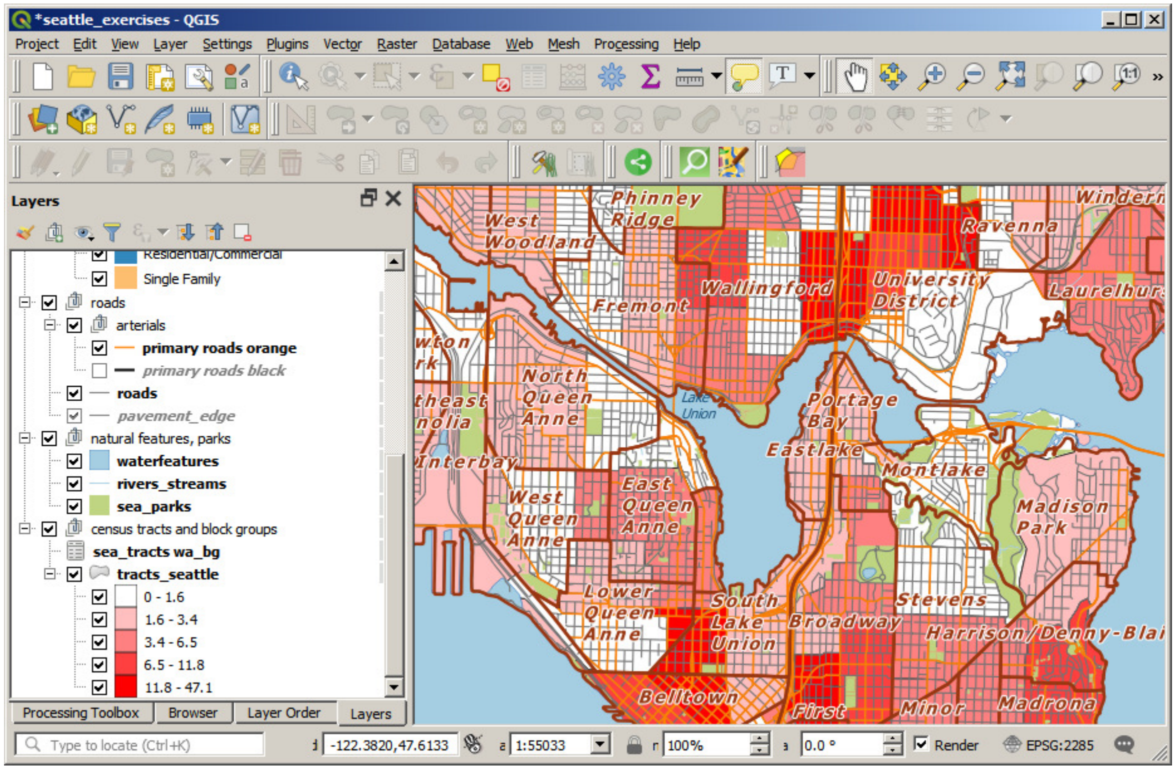


Exercises
Working with tables and layers

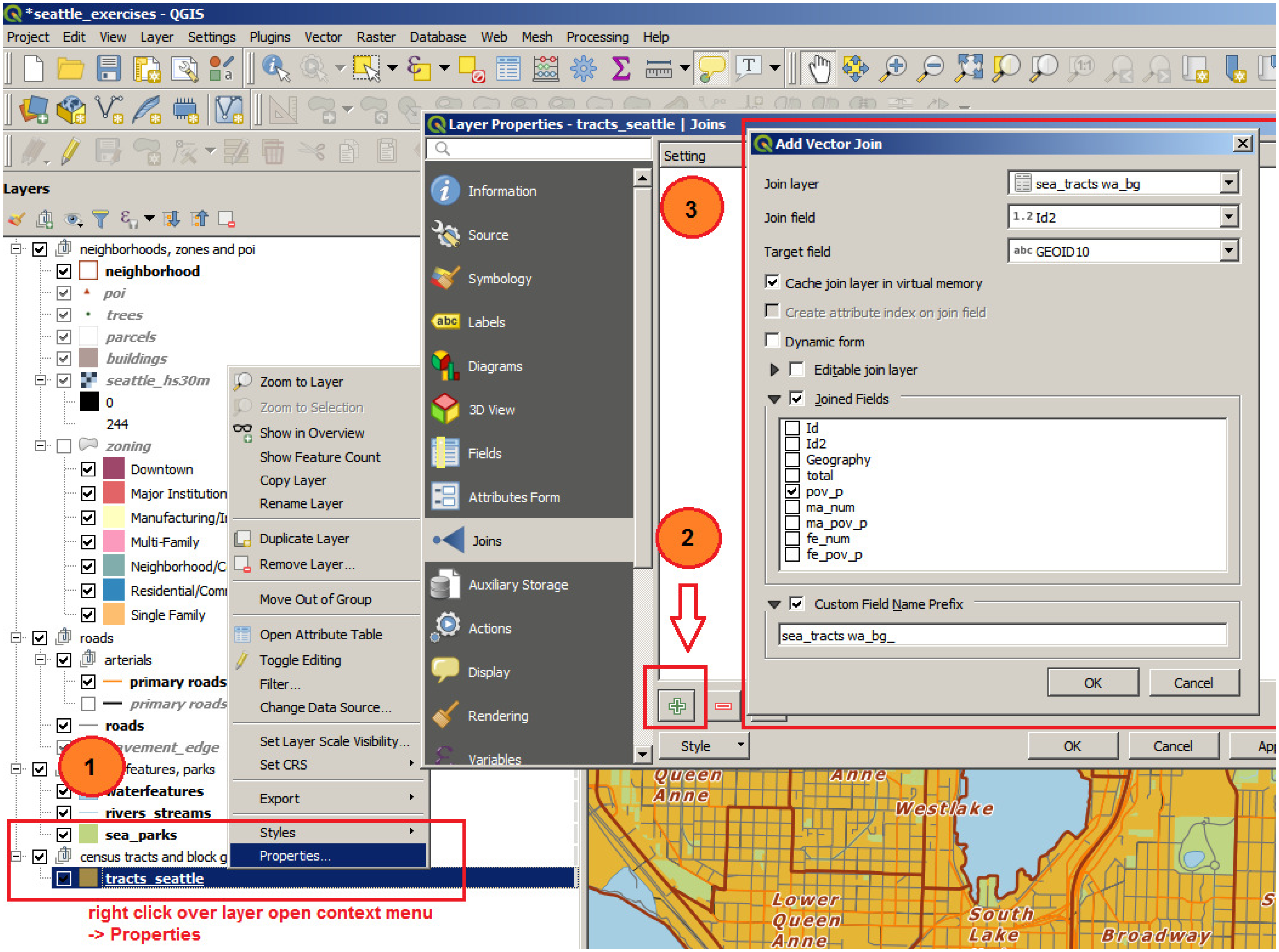
TERRA GIS
TERRESTRIAL ENVIRONMENT REGIONAL ANALYSIS

Working with tables and layers (Task 3)

- open existing QGIS map document data/seattle/seattle_exercises.qgz
- add census tracts layer data/seattle/data/tracts_seattle.shp and /seattle/census/sea_tracts.csv **+** review attribute data tables
- join information for people in poverty using fields geoid10 and Id2
- classify layer „graduated"
- needs more work because field is joined as a „text" field
- calculate virtual field using decimal number format



Working with tables and layers (Task 3)



Add Vector Join

Join layer: sea_tracts wa_bg

Join field: 1.2 Id2

Target field: abc GEOID10

☑ Cache join layer in virtual memory

☐ Create attribute index on join field

☐ Dynamic form

☐ Editable join layer

☑ Joined Fields

☐ Id
☐ Id2
☐ Geography
☐ total
☑ pov_p
☐ ma_num
☐ ma_pov_p
☐ fe_num
☐ fe_pov_p

☑ Custom Field Name Prefix

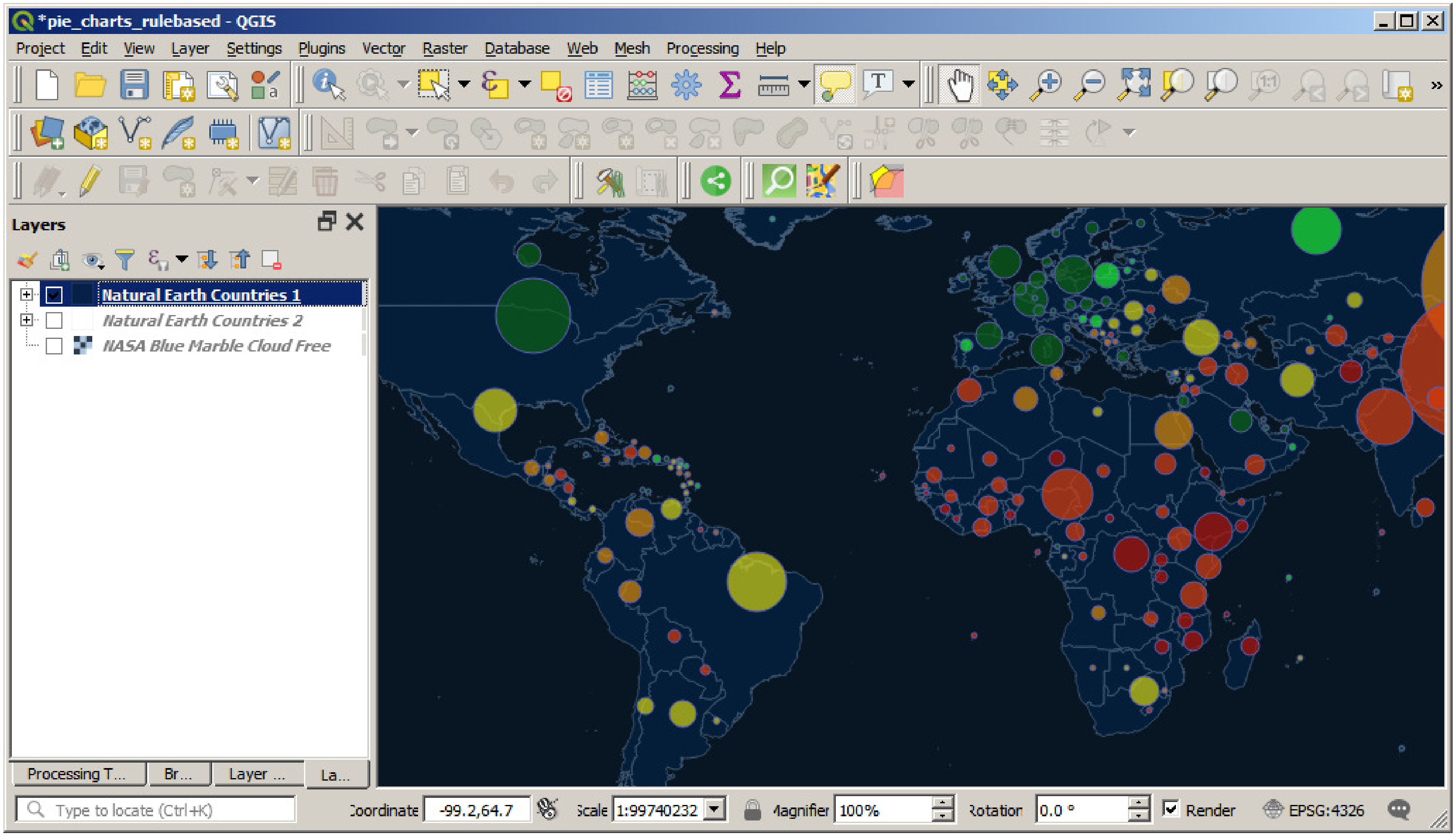
sea_tracts wa_bg_

OK Cancel

right click over layer open context menu
-> Properties

Example Rule based Rendering with Pie Charts

Project \examples\maps\pie_charts_rulebased.qgz



Coordinate -99.2,64.7 Scale 1:99740232 Magnifier 100% Rotation 0.0 ° ☑ Render EPSG:4326

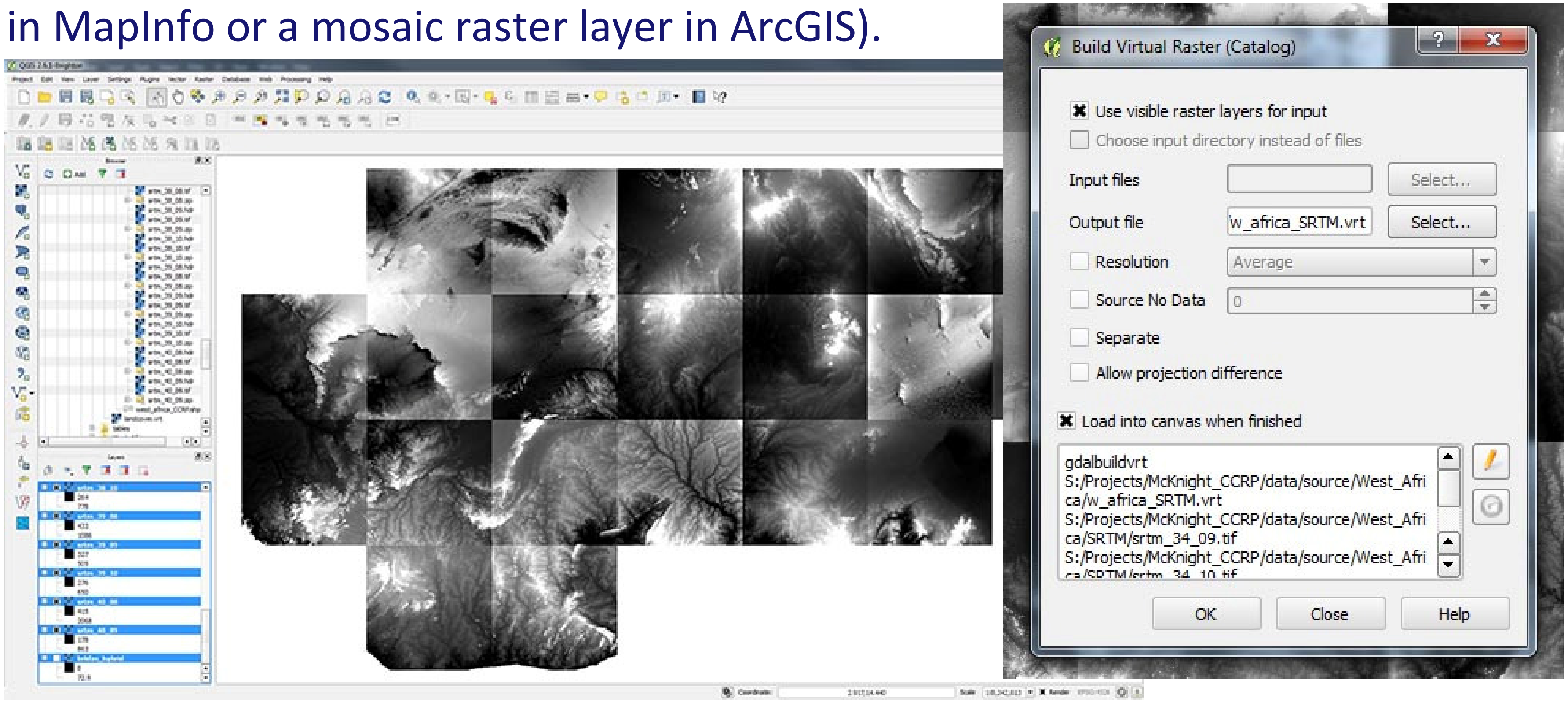
Workshop Day 2

QGIS - Managing Raster Data

QGIS - Managing Raster Data

Virtual Raster

You can deal with multiple files like they are one file by creating a virtual raster. This can be done by selecting the Raster > Miscellaneous > Build Virtual Raster (Catalog) menu option. This creates a mosaic of the images (like a seamless layer in MapInfo or a mosaic raster layer in ArcGIS).



Build Virtual Raster (Catalog)

☒ Use visible raster layers for input

☐ Choose input directory instead of files

Input files:

Output file: w_africa_SRTM.vrt

☐ Resolution: Average

☐ Source No Data: 0

☐ Separate

☐ Allow projection difference

☒ Load into canvas when finished

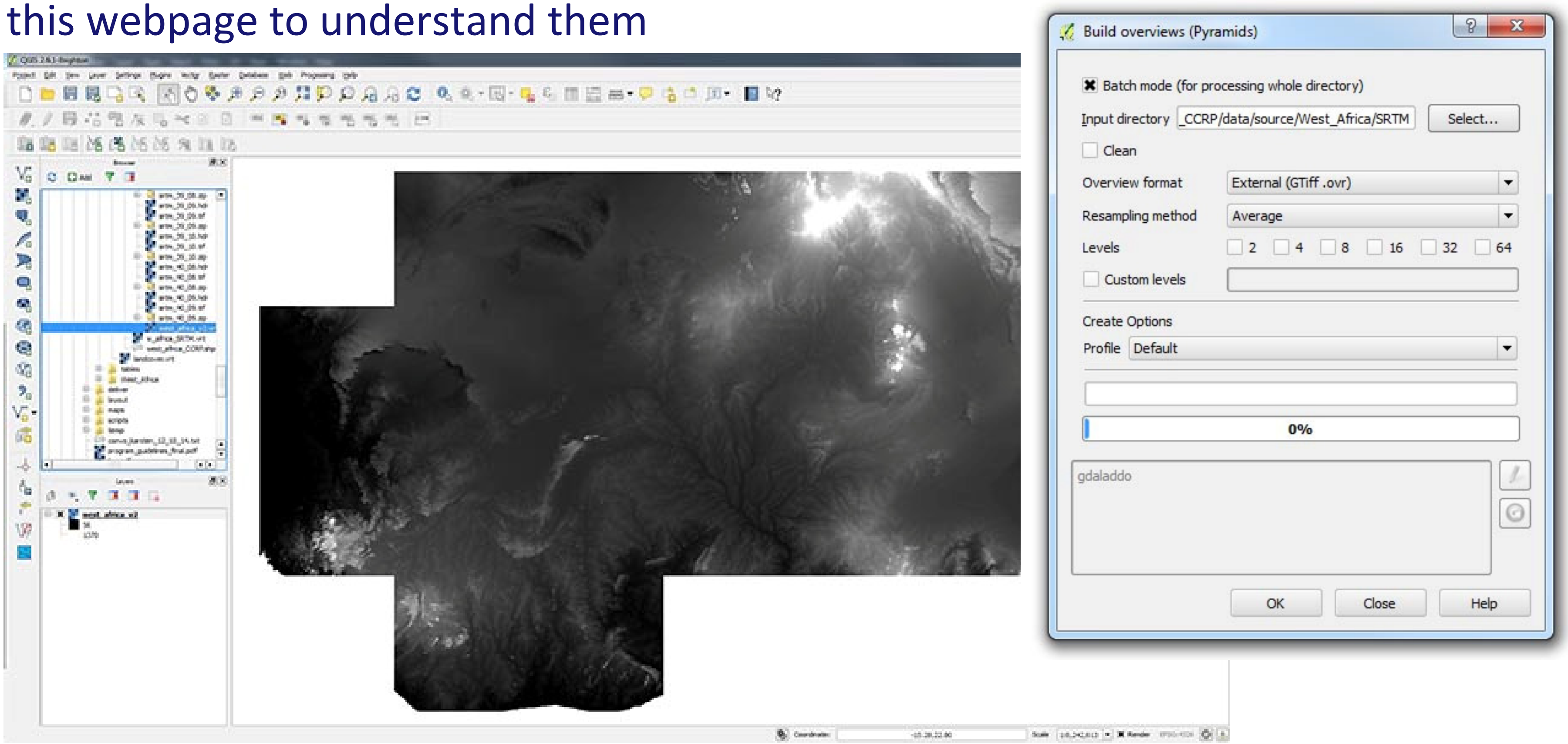
```
gdalbuildvrt
S:/Projects/McKnight_CCRP/data/source/West_Afri
ca/w_africa_SRTM.vrt
S:/Projects/McKnight_CCRP/data/source/West_Afri
ca/SRTM/srtm_34_09.tif
S:/Projects/McKnight_CCRP/data/source/West_Afri
ca/SRTM/srtm_34_10.tif
```

QGIS - Managing Raster Data

- Build Overviews (Pyramids in ArcGIS)

You can also create pyramids on multiple datasets in one go by using the Raster > Miscellaneous > Build Overviews menu option.

This allows for a batch mode. It has more advanced options, and its best to read this webpage to understand them



Build overviews (Pyramids)

☒ Batch mode (for processing whole directory)

Input directory: _CCRP/data/source/West_Africa/SRTM [Select...]

□ Clean

Overview format: External (GTiff .ovr)

Resampling method: Average

Levels: □ 2 □ 4 □ 8 □ 16 □ 32 □ 64

□ Custom levels

Create Options

Profile: Default

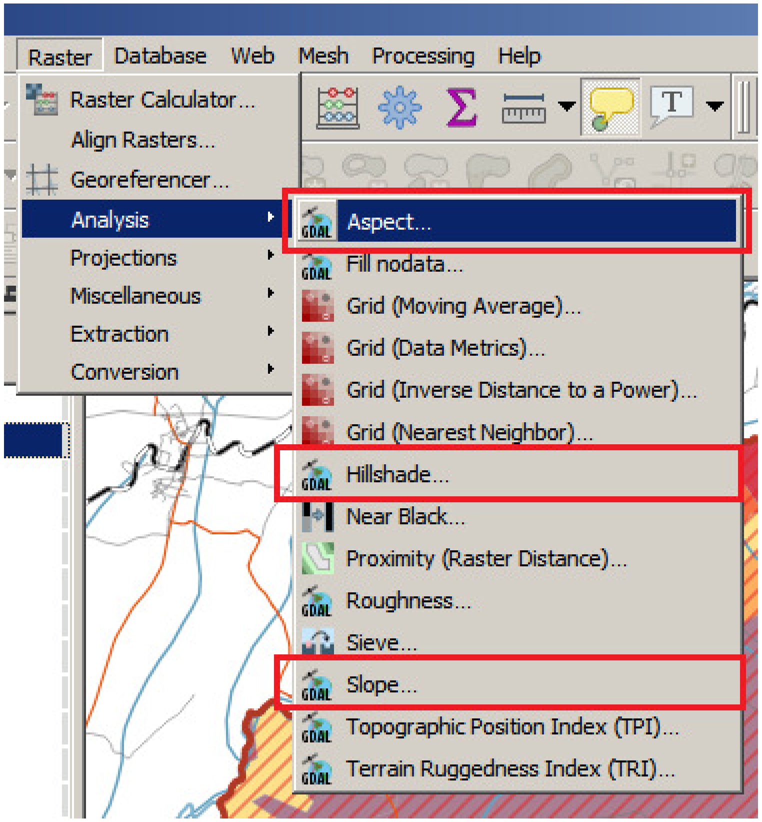
0%

gdaladdo

Exercises – Raster data

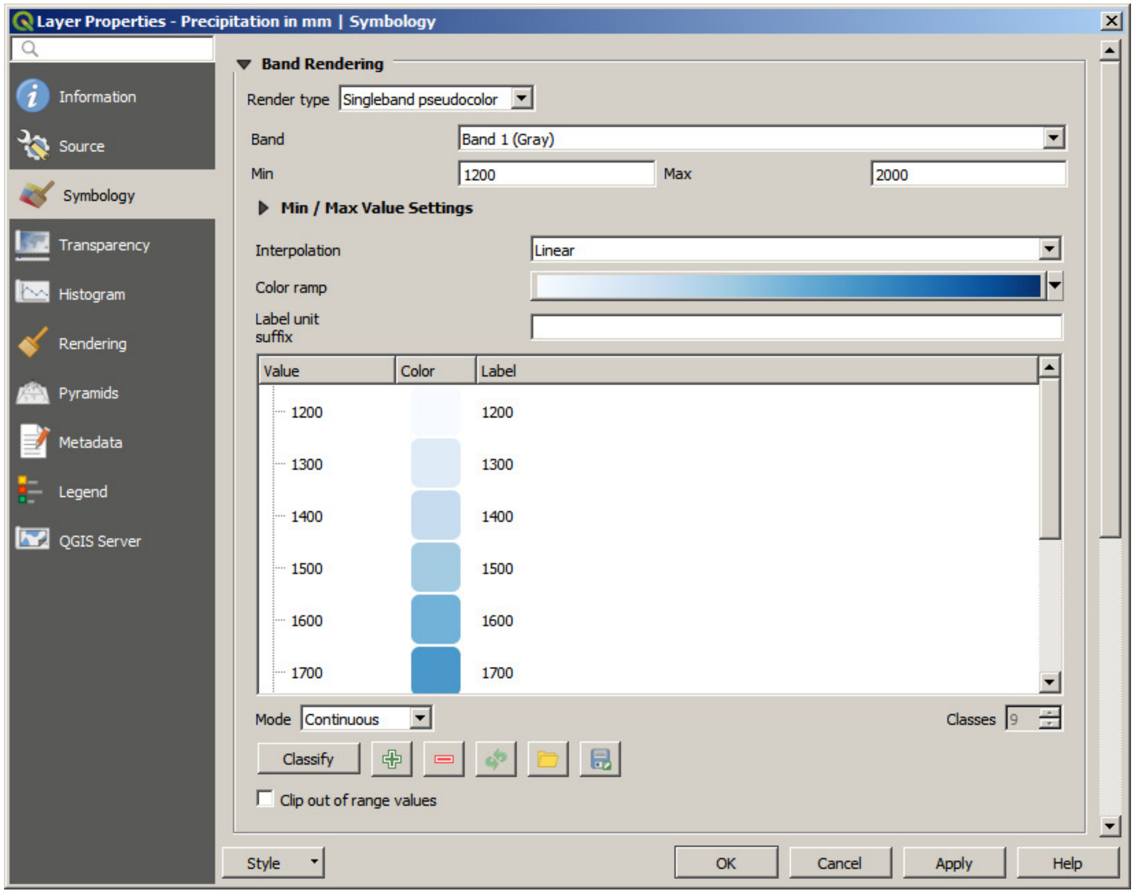
Managing Raster Data (Task 4)

- open existing QGIS map document data/kenya/seattle_exercises.qgz
- load 4 Kenya DEM raster layers *.tif files in /data/kenya/dem/ (Task 4.1)
- create a vrt file from the four dem layers (Task 4.2)
- create hill shade,
  slope,
  and aspect layers
  from DEM (Task 4.3)



Managing Raster Data (Task 5)

- in existing QGIS map document **/data/kenya/kenya_exercises.qgz**
- open layer properties for "Precipitation in mm" /data/kenya/newrain.tif
- review the existing Rendering type and the other types available
- review the existing classification
- change to show only 5 classes



Layer Properties - Precipitation in mm | Symbology

▼ Band Rendering

Render type: Singleband pseudocolor

Band: Band 1 (Gray)

Min: 1200 Max: 2000

▶ Min / Max Value Settings

Interpolation: Linear

Color ramp

Label unit suffix

Value	Color	Label
1200	■	1200
1300	■	1300
1400	■	1400
1500	■	1500
1600	■	1600
1700	■	1700

Mode: Continuous Classes: 9

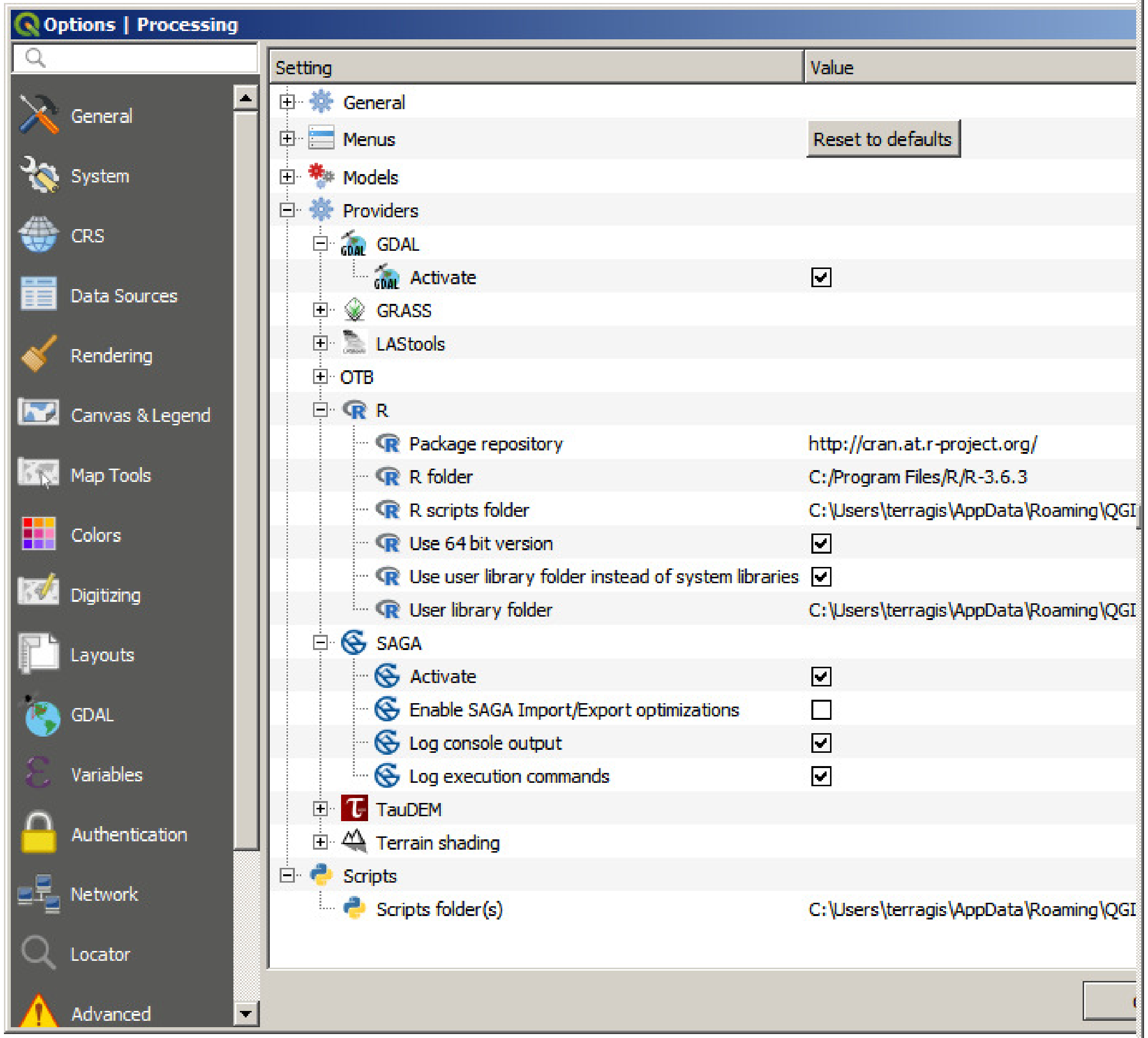
☐ Clip out of range values

Spatial Analysis and Tools

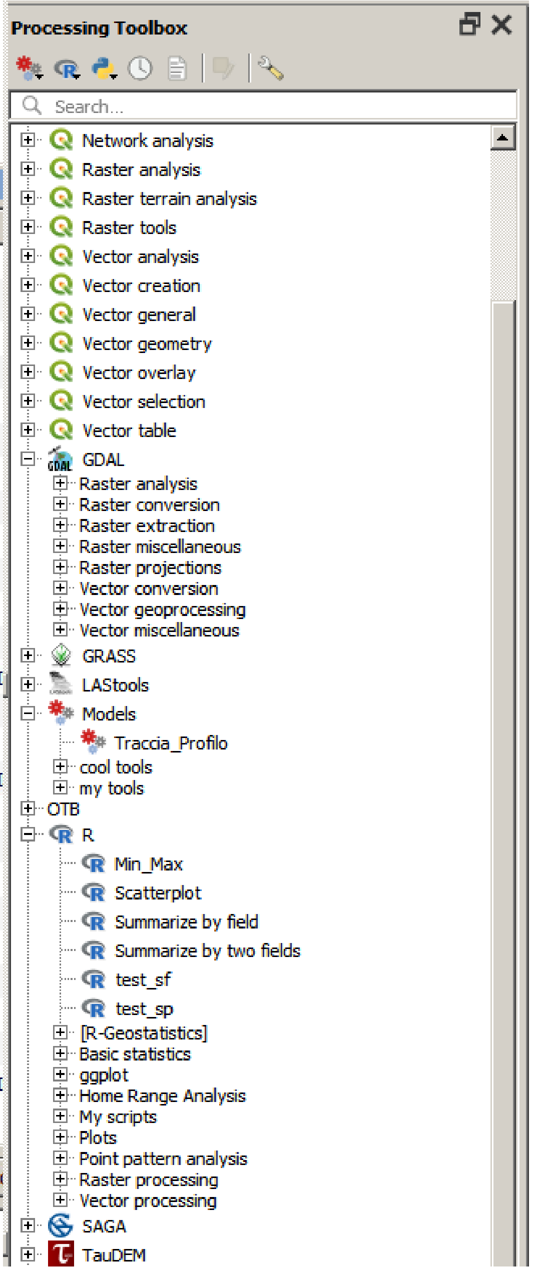
Geospatial Processing in QGIS - Exercises

- Using the geo-processing tools and graphical model builder
- Exercises
  e.g. counting trees
  in Seattle neighborhoods

QGIS – Processing Toolbox



Setting	Value
General	
Menus	Reset to defaults
Models	
Providers	
GDAL	
Activate	☑
GRASS	
LAStools	
OTB	
R	
Package repository	http://cran.at.r-project.org/
R folder	C:/Program Files/R/R-3.6.3
R scripts folder	C:\Users\terragis\AppData\Roaming\QGI
Use 64 bit version	☑
Use user library folder instead of system libraries	☑
User library folder	C:\Users\terragis\AppData\Roaming\QGI
SAGA	
Activate	☑
Enable SAGA Import/Export optimizations	☐
Log console output	☑
Log execution commands	☑
TauDEM	
Terrain shading	
Scripts	
Scripts folder(s)	C:\Users\terragis\AppData\Roaming\QGI



- Network analysis
- Raster analysis
- Raster terrain analysis
- Raster tools
- Vector analysis
- Vector creation
- Vector general
- Vector geometry
- Vector overlay
- Vector selection
- Vector table
- GDAL
  - Raster analysis
  - Raster conversion
  - Raster extraction
  - Raster miscellaneous
  - Raster projections
  - Vector conversion
  - Vector geoprocessing
  - Vector miscellaneous
- GRASS
- LAStools
- Models
  - Traccia_Profilo
  - cool tools
  - my tools
- OTB
- R
  - Min_Max
  - Scatterplot
  - Summarize by field
  - Summarize by two fields
  - test_sf
  - test_sp
  - [R-Geostatistics]
  - Basic statistics
  - ggplot
  - Home Range Analysis
  - My scripts
  - Plots
  - Point pattern analysis
  - Raster processing
  - Vector processing
- SAGA
- TauDEM

Exercises - Analysis

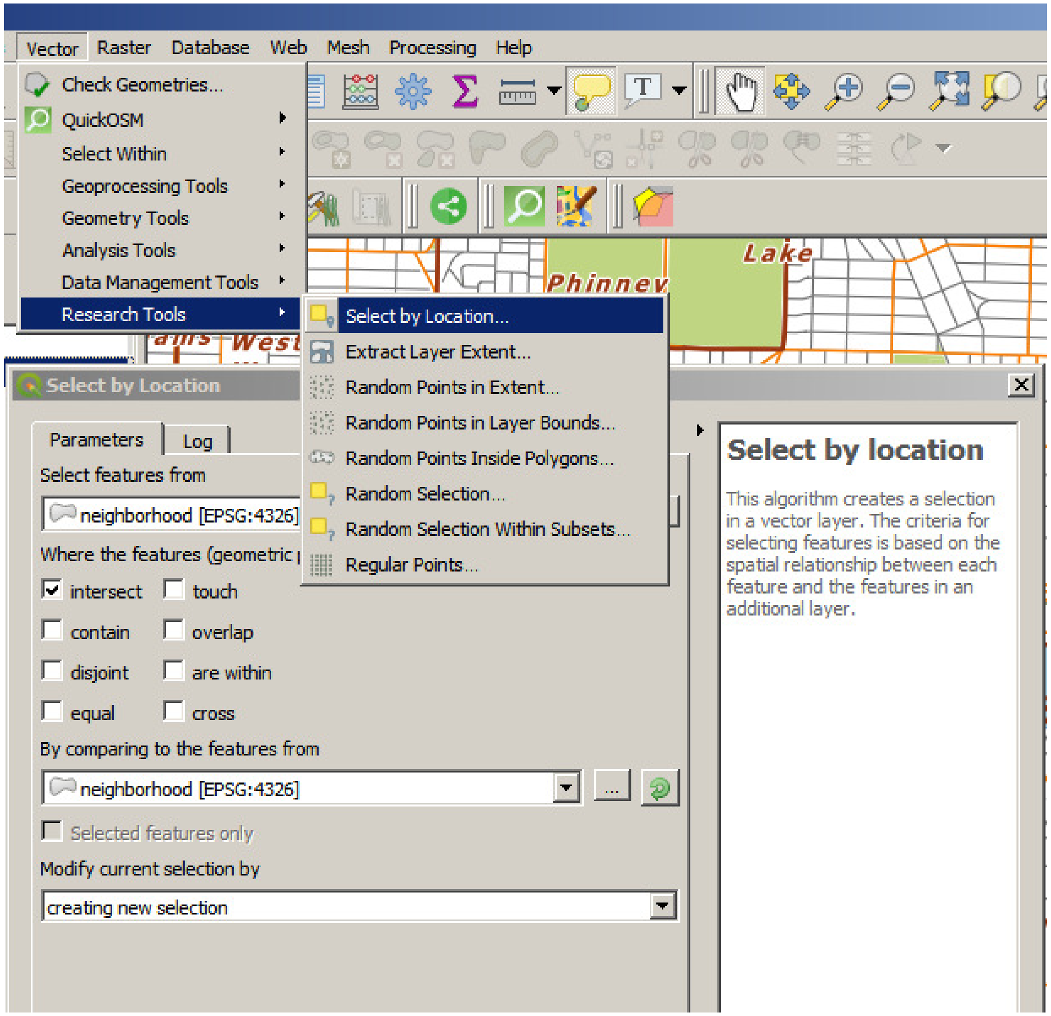
Analysis (Task 6)

- open existing QGIS map document data/seattle/seattle_exercises.qgs
- make a map of the neighborhoods colored based on the number of trees in each neighborhood (Task 6.1)
- label the neighborhoods including the total count of trees (Task 6.2)

Hint: You'll need a spatial join.

Analysis (Task 6)

Creating a selection



Select features from

neighborhood [EPSG:4326]

Where the features (geometric

☑ intersect ☐ touch

☐ contain ☐ overlap

☐ disjoint ☐ are within

☐ equal ☐ cross

By comparing to the features from

neighborhood [EPSG:4326]

☐ Selected features only

Modify current selection by

creating new selection

Select by location

This algorithm creates a selection in a vector layer. The criteria for selecting features is based on the spatial relationship between each feature and the features in an additional layer.

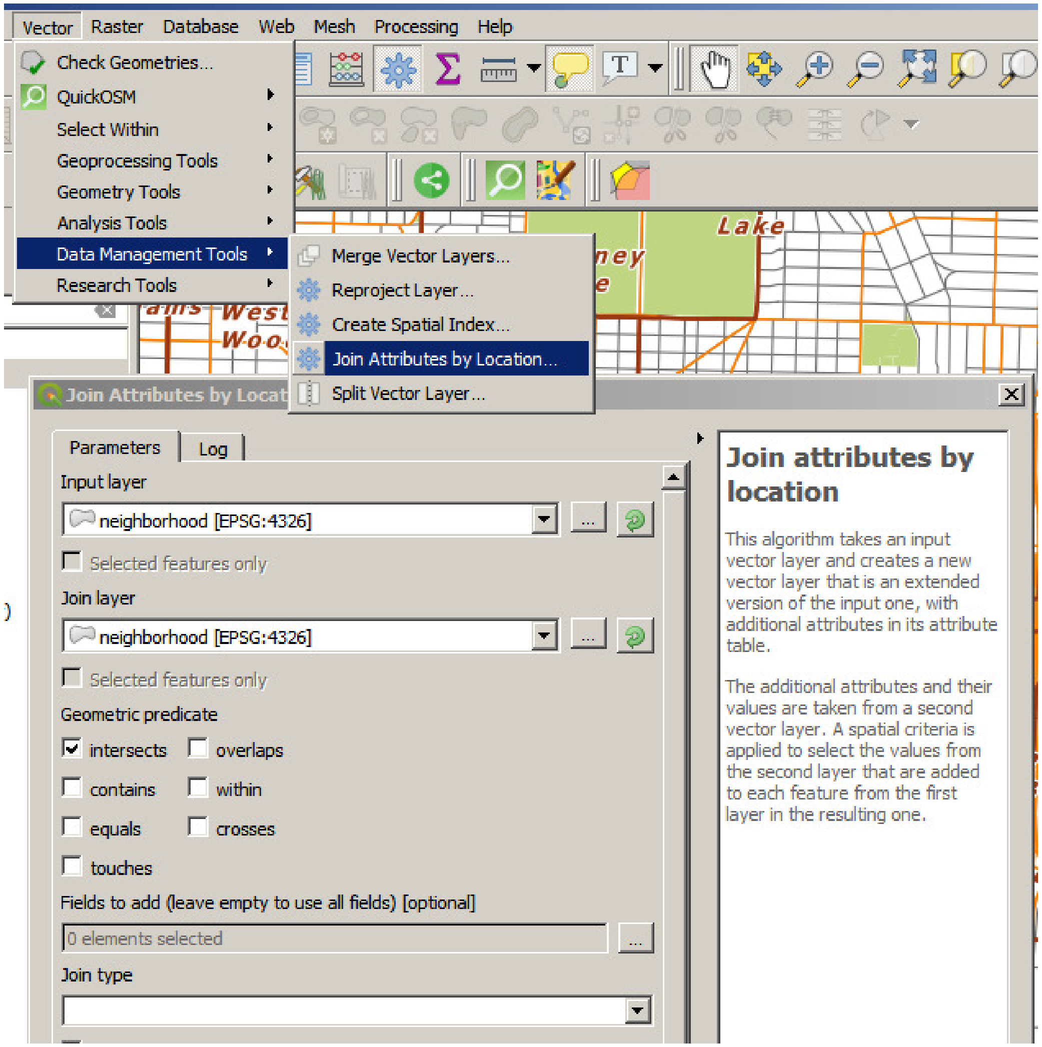
Analysis (Task 6)

Calculating numbers

“Count points in polygon”

Or

„Join attributes by location“



Input layer
neighborhood [EPSG:4326]

□ Selected features only

Join layer
neighborhood [EPSG:4326]

□ Selected features only

Geometric predicate

☑ intersects □ overlaps

□ contains □ within

□ equals □ crosses

□ touches

Fields to add (leave empty to use all fields) [optional]
0 elements selected

Join type

Join attributes by location

This algorithm takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. A spatial criteria is applied to select the values from the second layer that are added to each feature from the first layer in the resulting one.

Analysis (Task 7)

– Make a map of the number of trees per capita (Task 7.1)

Hint: You'll need population data from tracts and a field calculation.

Analysis (Task 7)

- Use the data provided and geoprocessing tools to locate areas where the city should plant more trees (Task 7.2)
- The manner in which you define/determine this is up to you.

Hints:

  - You could set a threshold on the number of trees per area (or capita)
  - You could use “Rule based” classification

Spatial Analysis and Tools

- **QGIS Plug-ins**
  - Default Plug-ins: e.g. DB Manager
  - Resource Sharing
  - OpenLayers (Google, Bing, Openstreetmap base maps)
  - Georeferencer
  - Semi Automatic Classification Plug-in (Remote sensing)
  - Others

Resources

Documents

PDF under workshop/docs/books folder (version 3.4)

QGIS User Guide

QGIS Training Manual

A Gentle Introduction to GIS

PyQGIS Developer Cookbook

QGIS Map Design - (can be bought from the Locate Press web site)

Learning QGIS (third edition) – (can be bought at the Packt web site)

The PyQGIS Programmer's Guide. Extending QGIS 3 with Python 3

Other resources for learning QGIS

OSGEO Foundation

- Conferences
- Email lists
- User groups
- OSGEO Planet blog aggregator

Geo For All is dedicated to promote the adoption of free and open source software for geospatial technology through education, research and public awareness.

- GeoAcademy
- Class videos on YouTube
- Class materials download

Workshop Day 3

Advanced functionalities with QGIS

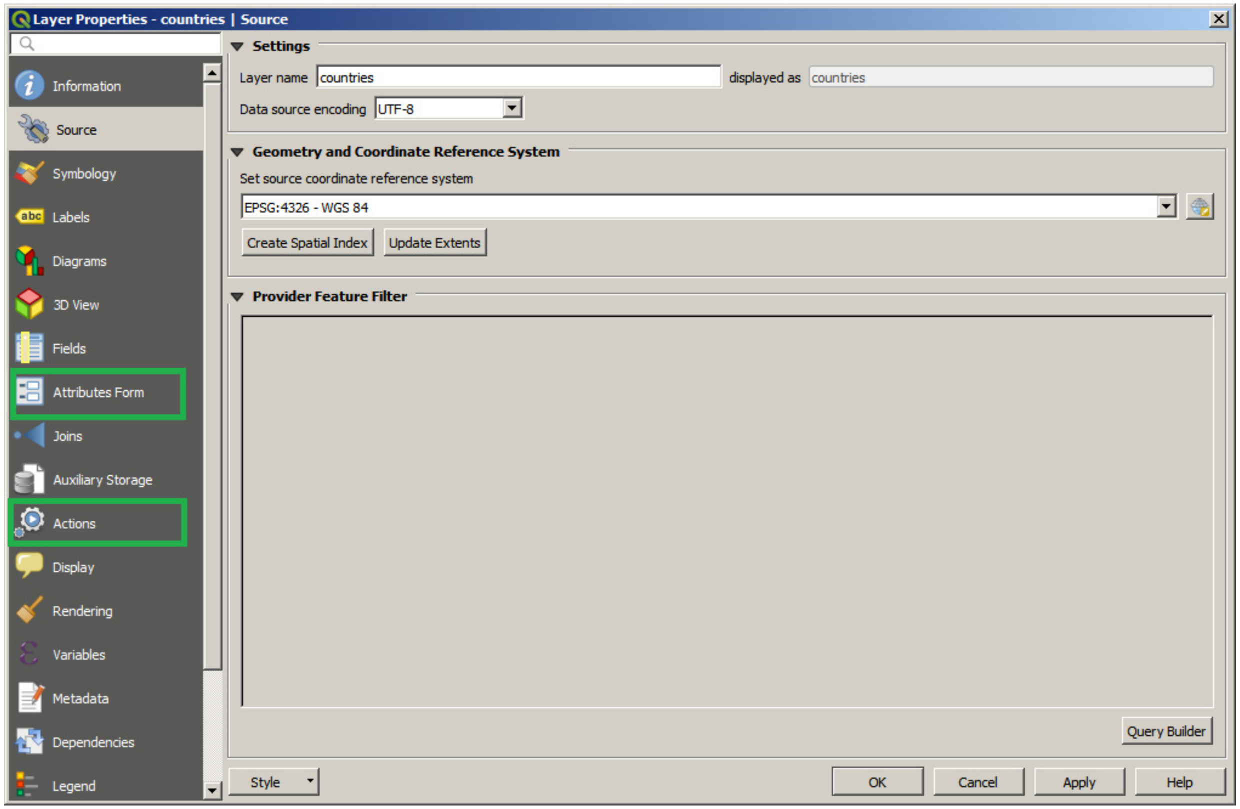
Introduction

Layer Properties

Lots of settings

interesting options including

- Flexible UI Attribute tables
- Attribute Forms e.g. using defined value drop down menus
- Custom Actions Python code



Layer Properties - countries | Source

Settings

Layer name: countries — displayed as: countries

Data source encoding: UTF-8

Geometry and Coordinate Reference System

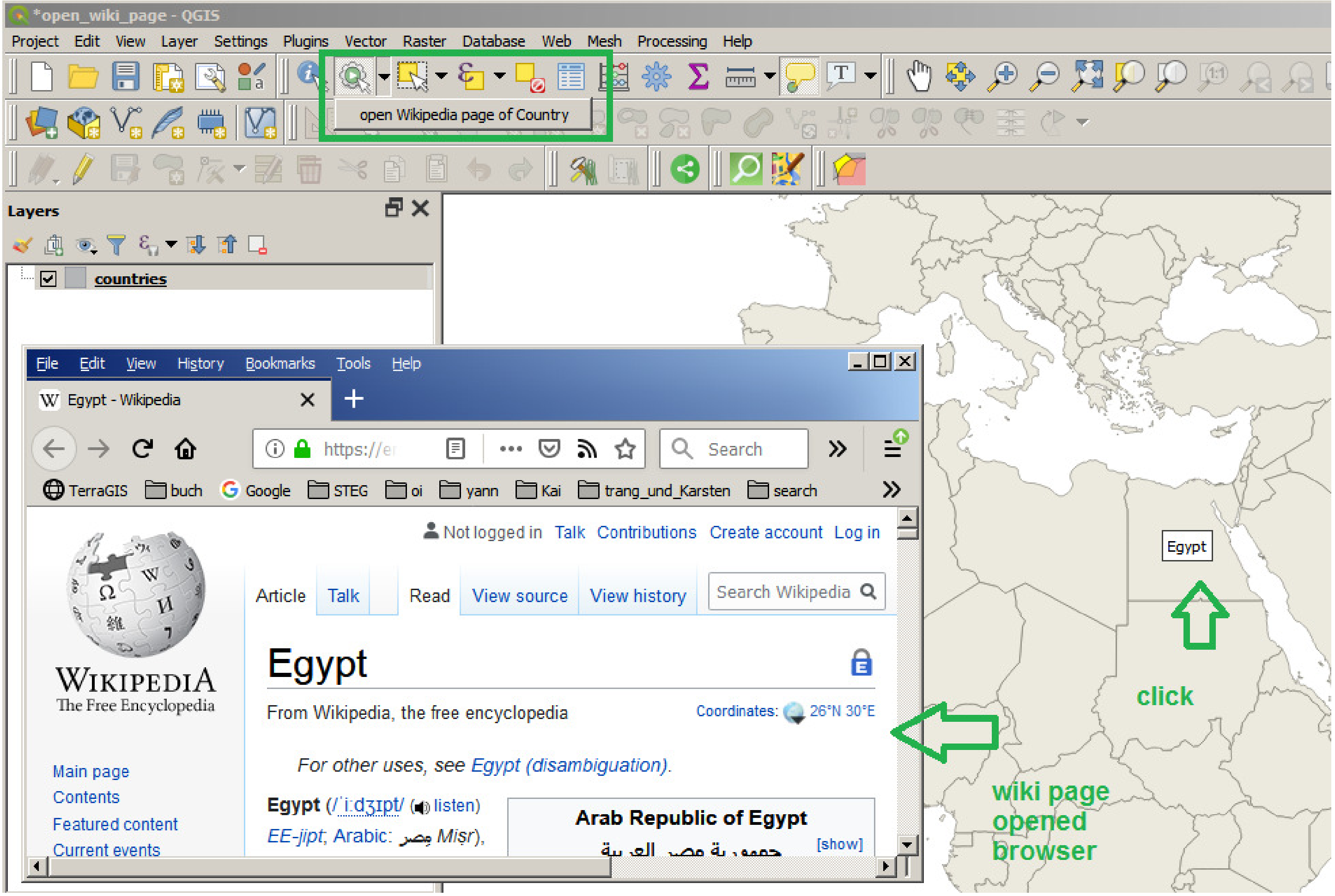
Set source coordinate reference system: EPSG:4326 - WGS 84

Provider Feature Filter

Using Layer ‘Actions’ to automate things

Layer Actions

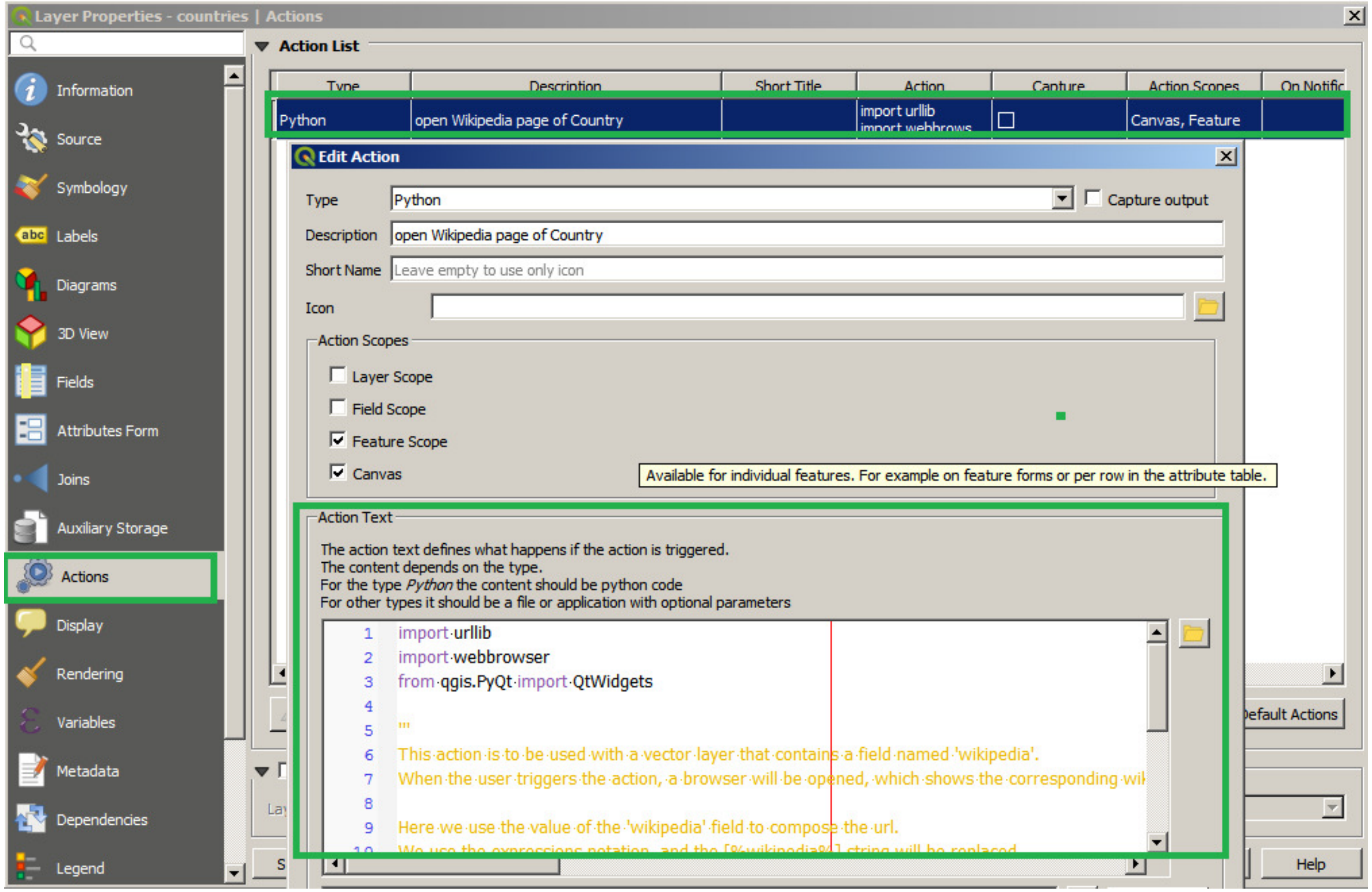
Open Wikipedia web page on click on a country on the map


*open_wiki_page - QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
open Wikipedia page of Country
Layers
countries
File Edit View History Bookmarks Tools Help
Egypt - Wikipedia
https://e
Search
TerraGIS buch Google STEG oi yann Kai trang_und_Karsten search
Not logged in Talk Contributions Create account Log in
Article Talk Read View source View history
Search Wikipedia
Wikipedia
The Free Encyclopedia
Egypt
From Wikipedia, the free encyclopedia
Coordinates: 26°N 30°E
For other uses, see Egypt (disambiguation).
Egypt (/ˈiːdʒɪpt/ listen)
EE-jipt; Arabic: مصر Miṣr),
Arab Republic of Egypt
جمهورية مصر العربية
[show]
Main page
Contents
Featured content
Current events
Egypt
click
wiki page
opened
browser

Layer Actions

Example: \examples\maps\open_wiki_page.qgz

Open Wikipedia web page on click on a country on the map



Type	Description	Short Title	Action	Capture	Action Scopes	On Notific

Python	open Wikipedia page of Country		import urllib
			import webbrows
				□	Canvas, Feature	

Type: Python ☐ Capture output

Description: open Wikipedia page of Country

Short Name: Leave empty to use only icon

Icon:

Action Scopes

☐ Layer Scope

☐ Field Scope

☑ Feature Scope

☑ Canvas

Available for individual features. For example on feature forms or per row in the attribute table.

Action Text

The action text defines what happens if the action is triggered.
The content depends on the type.
For the type *Python* the content should be python code
For other types it should be a file or application with optional parameters

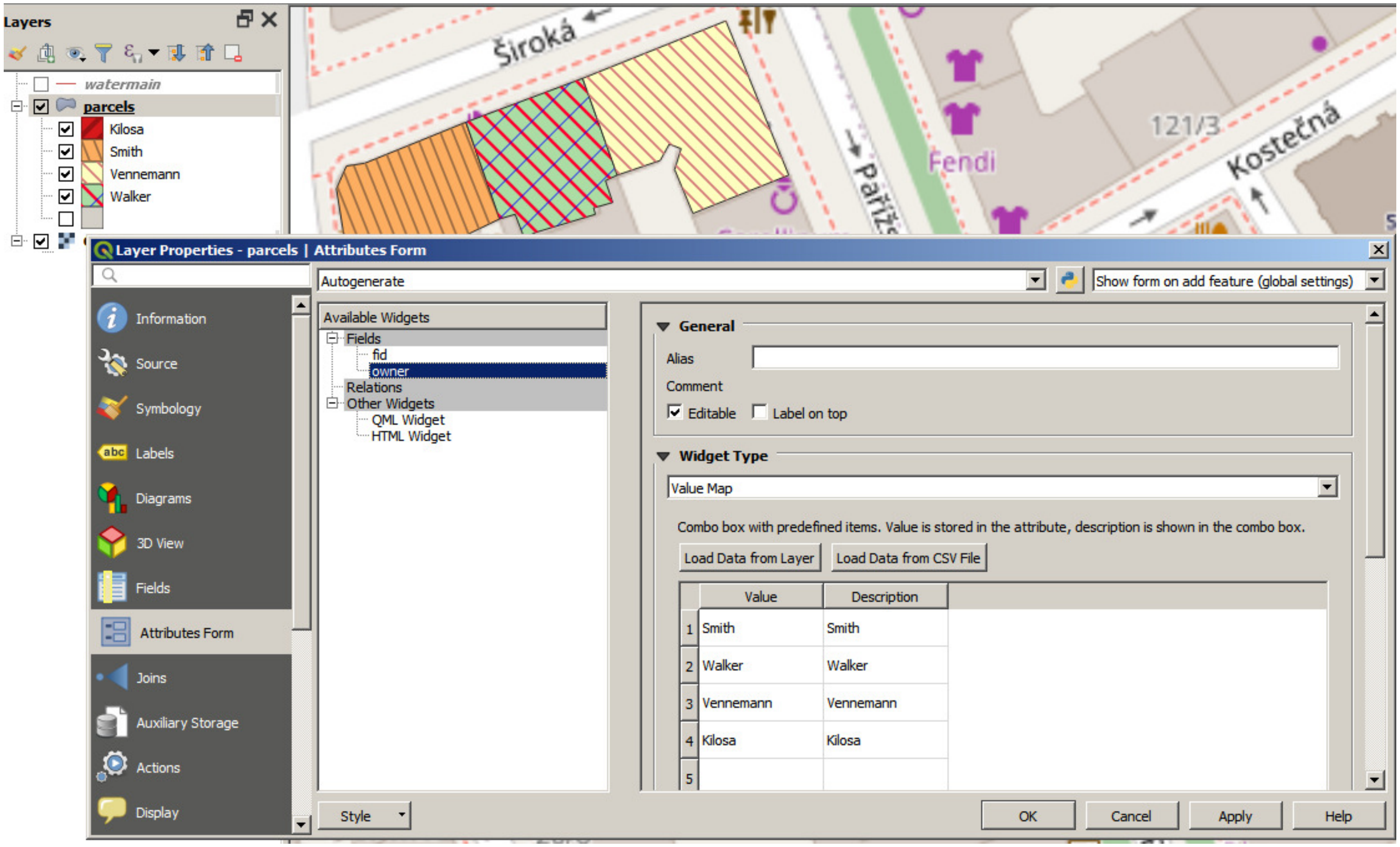
```
import urllib
import webbrowser
from qgis.PyQt import QtWidgets

'''
This action is to be used with a vector layer that contains a field named 'wikipedia'.
When the user triggers the action, a browser will be opened, which shows the corresponding wik

Here we use the value of the 'wikipedia' field to compose the url.
We use the expressions notation, and the [%wikipedia%] string will be replaced
```

Editing Layers and custom interfaces

Editing Layers

Using defined value drop down menus

Example: \examples\maps\editing.qgz



Autogenerate

Show form on add feature (global settings)

Alias:

Comment

☑ Editable ☐ Label on top

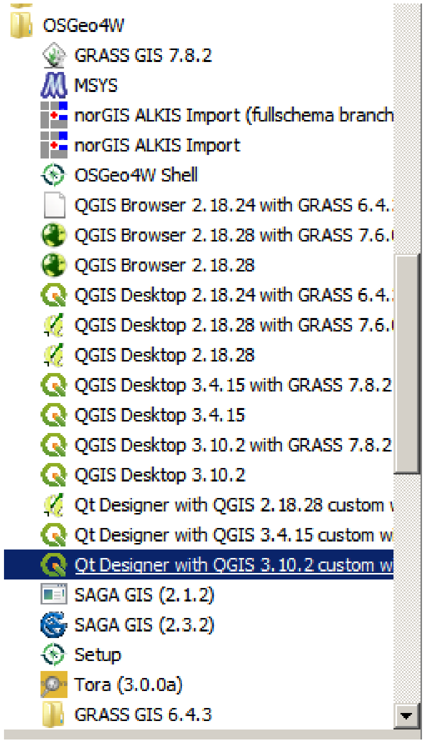
Widget Type: Value Map

Combo box with predefined items. Value is stored in the attribute, description is shown in the combo box.

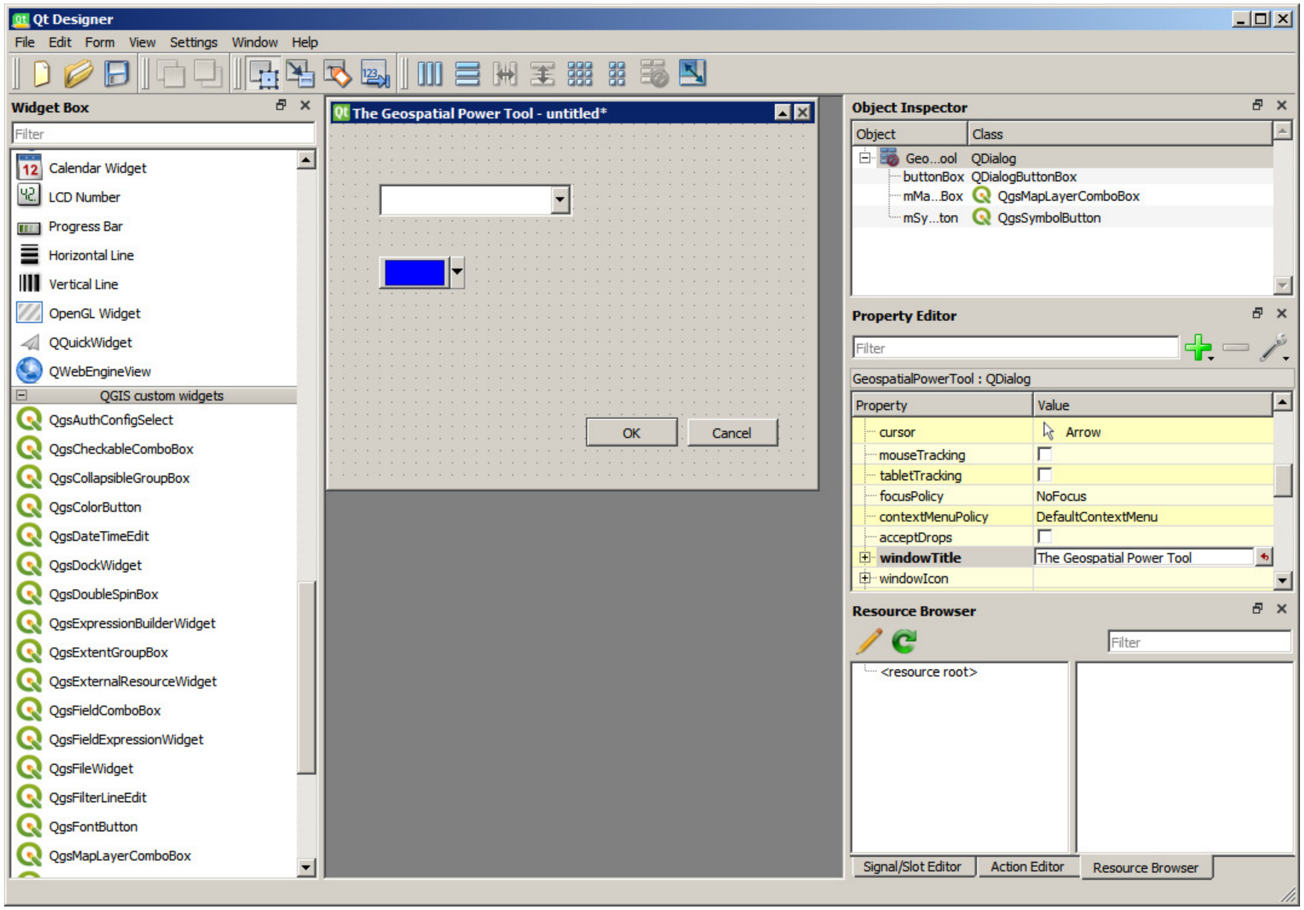
	Value	Description
1	Smith	Smith
2	Walker	Walker
3	Vennemann	Vennemann
4	Kilosa	Kilosa
5		

Interfaces with QT Designer

Can be installed via OSGeo4w, includes custom QGIS widgets, can be saved as *.ui file, a tutorial is here



OSGeo4W
GRASS GIS 7.8.2
MSYS
norGIS ALKIS Import (fullschema branch
norGIS ALKIS Import
OSGeo4W Shell
QGIS Browser 2.18.24 with GRASS 6.4.
QGIS Browser 2.18.28 with GRASS 7.6.
QGIS Browser 2.18.28
QGIS Desktop 2.18.24 with GRASS 6.4.
QGIS Desktop 2.18.28 with GRASS 7.6.
QGIS Desktop 2.18.28
QGIS Desktop 3.4.15 with GRASS 7.8.2
QGIS Desktop 3.4.15
QGIS Desktop 3.10.2 with GRASS 7.8.2
QGIS Desktop 3.10.2
Qt Designer with QGIS 2.18.28 custom w
Qt Designer with QGIS 3.4.15 custom w
Qt Designer with QGIS 3.10.2 custom w
SAGA GIS (2.1.2)
SAGA GIS (2.3.2)
Setup
Tora (3.0.0a)
GRASS GIS 6.4.3



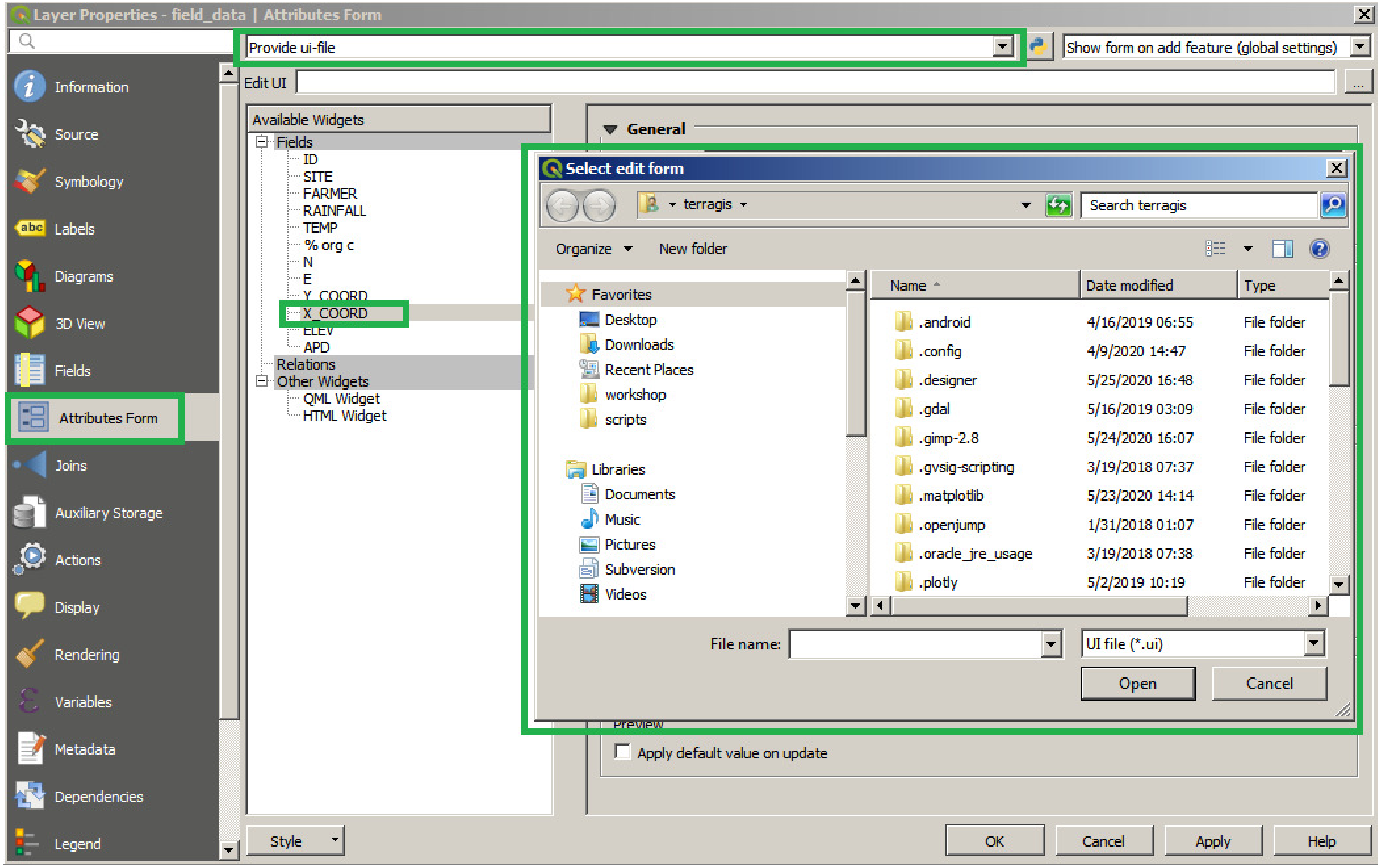
Object	Class
Geo...ool	QDialog
buttonBox	QDialogButtonBox
mMa...Box	QgsMapLayerComboBox
mSy...ton	QgsSymbolButton

GeospatialPowerTool : QDialog

Property	Value
cursor	Arrow
mouseTracking	□
tabletTracking	□
focusPolicy	NoFocus
contextMenuPolicy	DefaultContextMenu
acceptDrops	□
windowTitle	The Geospatial Power Tool
windowIcon	

Adding a custom *.ui file

Layer Properties -> Attribute Form



Name	Date modified	Type
.android	4/16/2019 06:55	File folder
.config	4/9/2020 14:47	File folder
.designer	5/25/2020 16:48	File folder
.gdal	5/16/2019 03:09	File folder
.gimp-2.8	5/24/2020 16:07	File folder
.gvsig-scripting	3/19/2018 07:37	File folder
.matplotlib	5/23/2020 14:14	File folder
.openjump	1/31/2018 01:07	File folder
.oracle_jre_usage	3/19/2018 07:38	File folder
.plotly	5/2/2019 10:19	File folder

QGIS and Spatial Databases

Extending GIS Capabilities

file based vs. server based

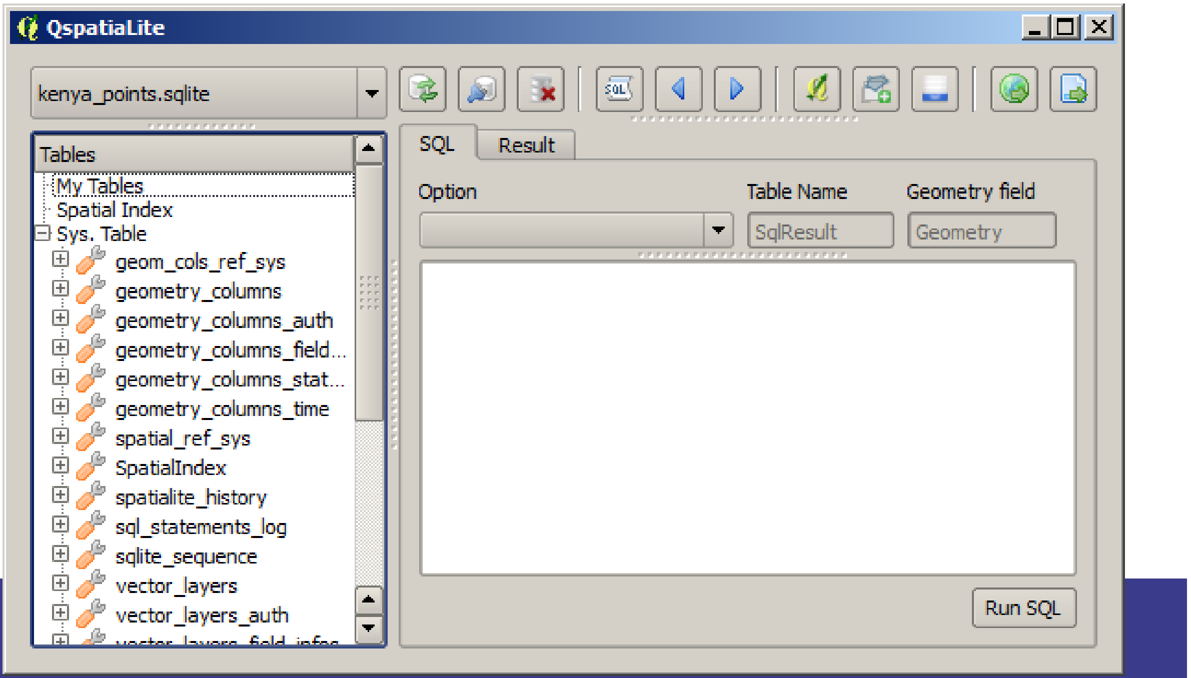
Spatialite – file based Spatial Database

SpatiaLite is a spatial DBMS built on top of **SQLite** . Both formats are file based and thus are light weight and portable. The spatial components depend on the PROJ and GEOS libraries. Related tools include the **RasterLite** library to handle Raster data and **spatialite-gis** (a minimalistic GIS tool). SpatiaLite has the potential to replace shapefiles as a simple data exchange format. Starting with version 1.1 QGIS can read the format, support by OGR/GDAL was included since version 1.7.0.

file based
DBMS
light weight
portable

Spatialite GUI

https://www.gaia-gis.it/fossil/libspatialite/index



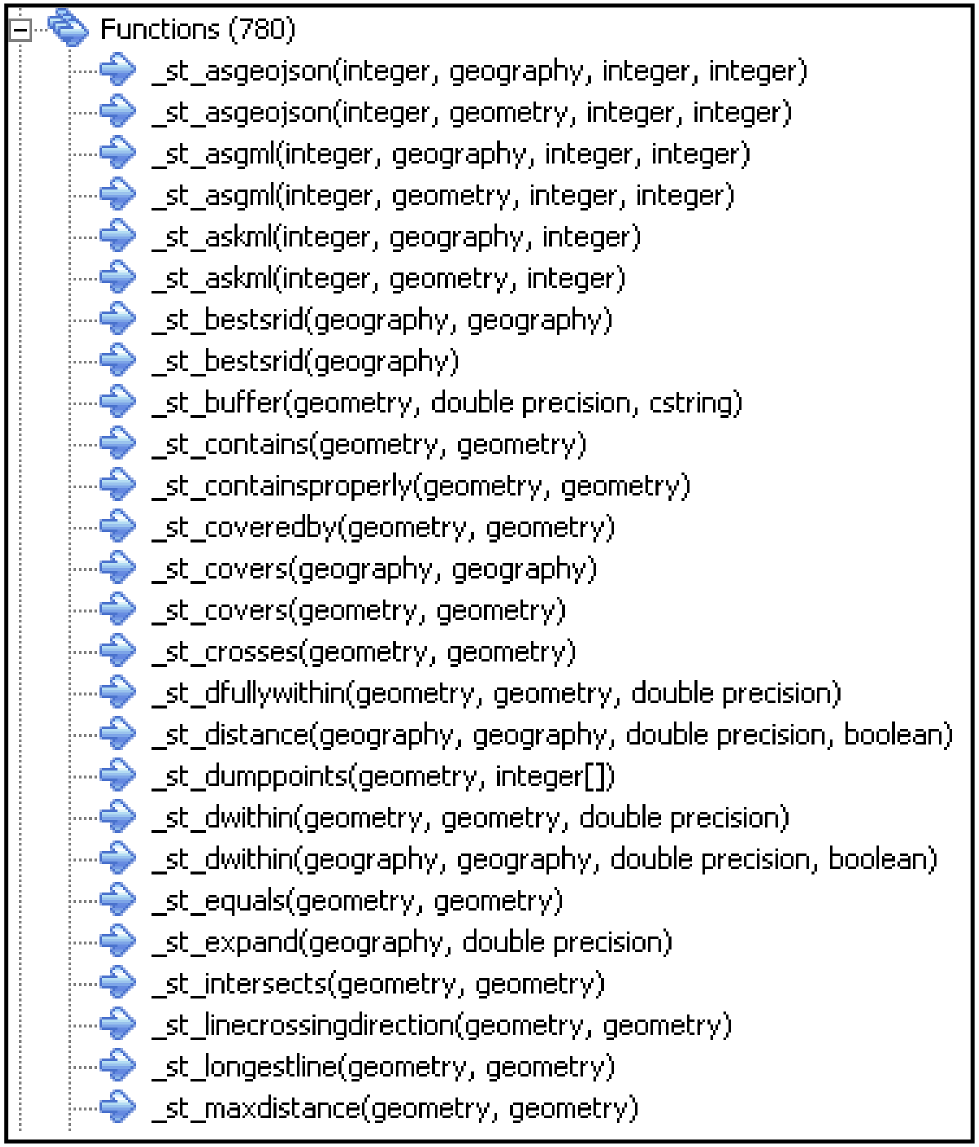
Tables
- My Tables
- Spatial Index
- Sys. Table
  - geom_cols_ref_sys
  - geometry_columns
  - geometry_columns_auth
  - geometry_columns_field...
  - geometry_columns_stat...
  - geometry_columns_time
  - spatial_ref_sys
  - SpatialIndex
  - spatialite_history
  - sql_statements_log
  - sqlite_sequence
  - vector_layers
  - vector_layers_auth
  - vector_layers_field_infos

PostGIS – Spatial Database

- PostGIS is an extension for PostgreSQL
- adds support for geographic objects to PostgreSQL
- enables PostgreSQL server to be used as a backend spatial database for GIS
- Spatial operations and analysis simply mean running a (spatial) SQL query in the database
- Similar functions to ESRI Arc SDE but also much more ….

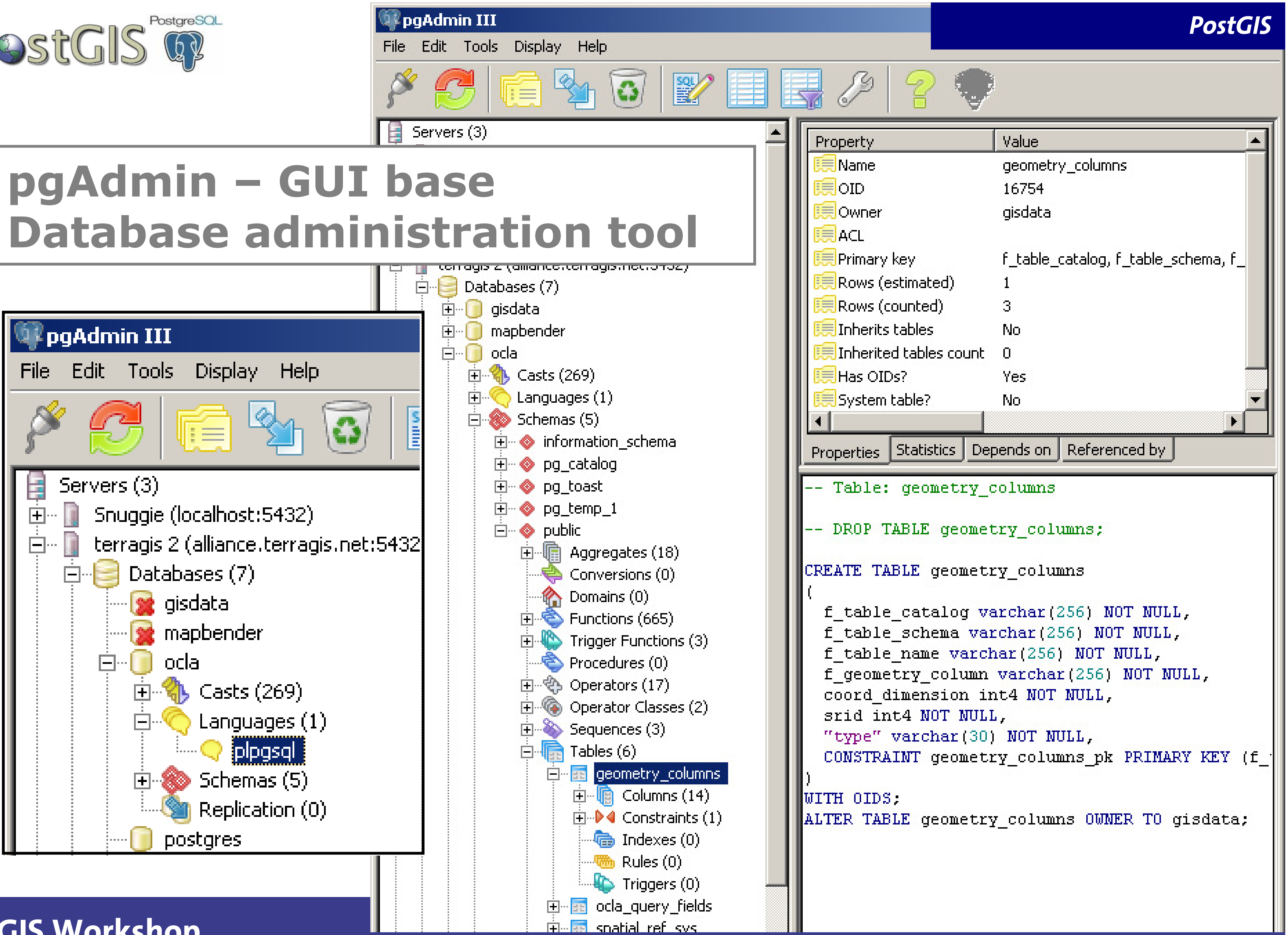
PostGIS Functions

- Spatial SQL



```
Functions (780)
    _st_asgeojson(integer, geography, integer, integer)
    _st_asgeojson(integer, geometry, integer, integer)
    _st_asgml(integer, geography, integer, integer)
    _st_asgml(integer, geometry, integer, integer)
    _st_askml(integer, geography, integer)
    _st_askml(integer, geometry, integer)
    _st_bestsrid(geography, geography)
    _st_bestsrid(geography)
    _st_buffer(geometry, double precision, cstring)
    _st_contains(geometry, geometry)
    _st_containsproperly(geometry, geometry)
    _st_coveredby(geometry, geometry)
    _st_covers(geography, geography)
    _st_covers(geometry, geometry)
    _st_crosses(geometry, geometry)
    _st_dfullywithin(geometry, geometry, double precision)
    _st_distance(geography, geography, double precision, boolean)
    _st_dumppoints(geometry, integer[])
    _st_dwithin(geometry, geometry, double precision)
    _st_dwithin(geography, geography, double precision, boolean)
    _st_equals(geometry, geometry)
    _st_expand(geography, double precision)
    _st_intersects(geometry, geometry)
    _st_linecrossingdirection(geometry, geometry)
    _st_longestline(geometry, geometry)
    _st_maxdistance(geometry, geometry)
```

pgAdmin – GUI base Database administration tool



```
Servers (3)
  Snuggie (localhost:5432)
  terragis 2 (alliance.terragis.net:5432
    Databases (7)
      gisdata
      mapbender
      ocla
        Casts (269)
        Languages (1)
          plpgsql
        Schemas (5)
        Replication (0)
      postgres
```

```
Servers (3)
  terragis 2 (alliance.terragis.net:5432)
    Databases (7)
      gisdata
      mapbender
      ocla
        Casts (269)
        Languages (1)
        Schemas (5)
          information_schema
          pg_catalog
          pg_toast
          pg_temp_1
          public
            Aggregates (18)
            Conversions (0)
            Domains (0)
            Functions (665)
            Trigger Functions (3)
            Procedures (0)
            Operators (17)
            Operator Classes (2)
            Sequences (3)
            Tables (6)
              geometry_columns
                Columns (14)
                Constraints (1)
                Indexes (0)
                Rules (0)
                Triggers (0)
              ocla_query_fields
              spatial_ref_sys
```

Property	Value
Name	geometry_columns
OID	16754
Owner	gisdata
ACL	
Primary key	f_table_catalog, f_table_schema, f_
Rows (estimated)	1
Rows (counted)	3
Inherits tables	No
Inherited tables count	0
Has OIDs?	Yes
System table?	No

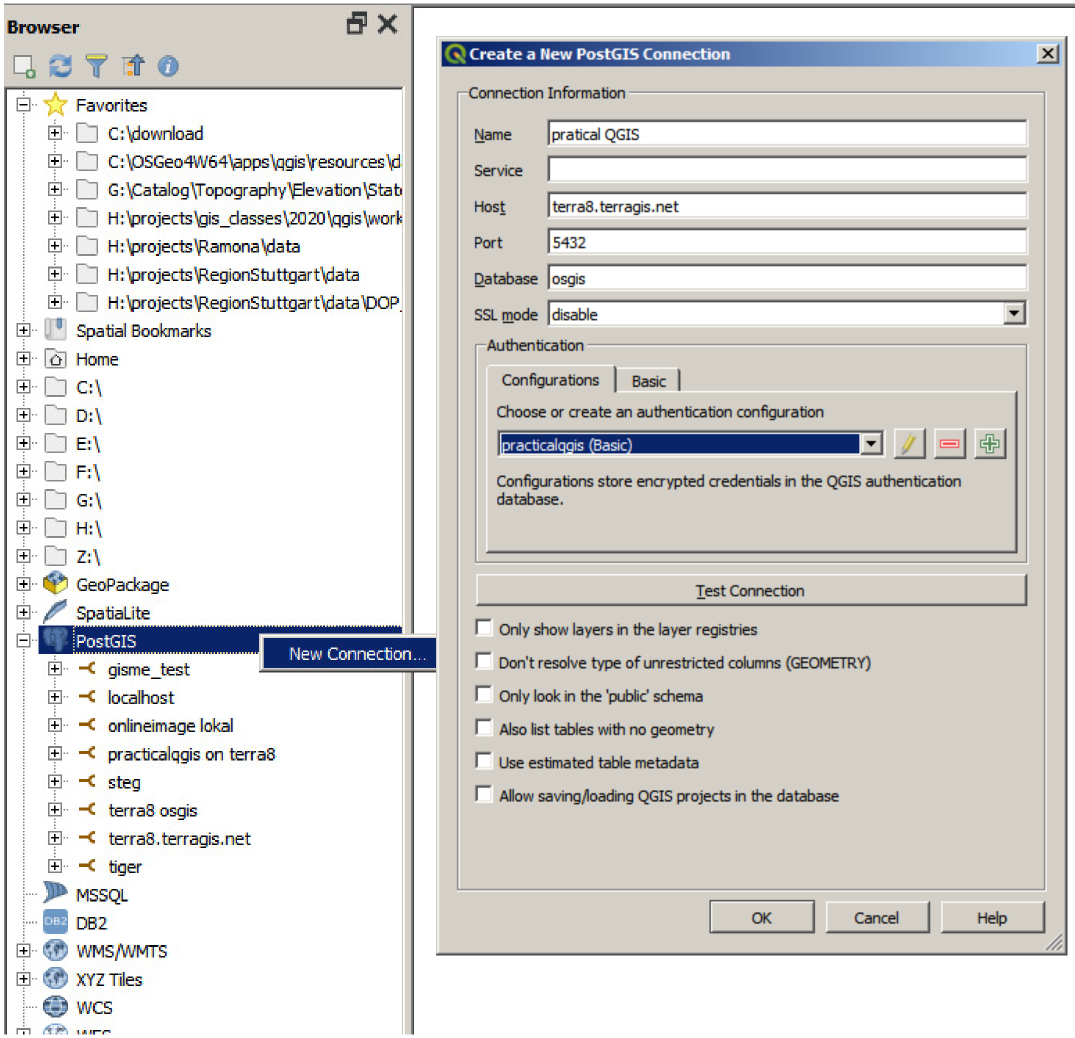
```
-- Table: geometry_columns

-- DROP TABLE geometry_columns;

CREATE TABLE geometry_columns
(
  f_table_catalog varchar(256) NOT NULL,
  f_table_schema varchar(256) NOT NULL,
  f_table_name varchar(256) NOT NULL,
  f_geometry_column varchar(256) NOT NULL,
  coord_dimension int4 NOT NULL,
  srid int4 NOT NULL,
  "type" varchar(30) NOT NULL,
  CONSTRAINT geometry_columns_pk PRIMARY KEY (f_
)
WITH OIDS;
ALTER TABLE geometry_columns OWNER TO gisdata;
```

Example for practical use of the PostGIS Database

- **Unified data storage and retrieval**
- **GIS functionalities**
  - Find nearest spatial features
    - Nearest road (reverse geocoding)
    - Nearest *conspecific* plant species (Whippet model)
  - Buffer, locate within another feature, and calculate distances (modeling)
  - Model calculations of attributes (leading to prioritization scores)
- **Extension of Web GIS capabilities**
  - Data queries for dynamic data display

Working with PostGIS data in QGIS

Create Database connections in Browser, right click …



Create a New PostGIS Connection

Connection Information

Name	pratical QGIS
Service	
Host	terra8.terragis.net
Port	5432
Database	osgis
SSL mode	disable

Authentication

Configurations | Basic

Choose or create an authentication configuration

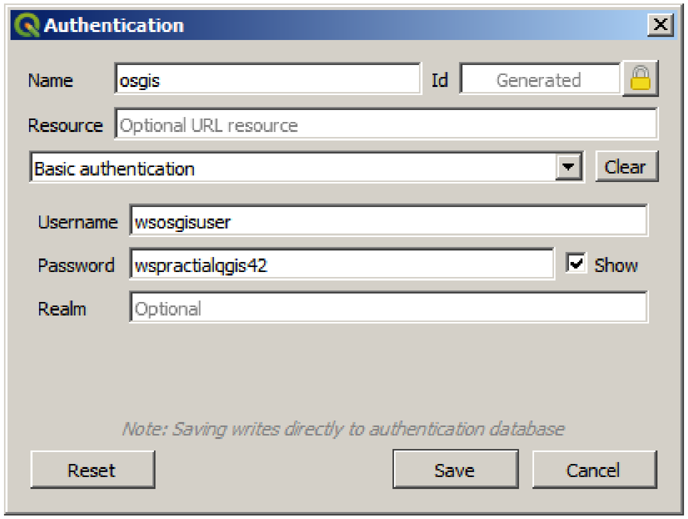
practicalqgis (Basic)

Configurations store encrypted credentials in the QGIS authentication database.

Test Connection

- ☐ Only show layers in the layer registries
- ☐ Don't resolve type of unrestricted columns (GEOMETRY)
- ☐ Only look in the 'public' schema
- ☐ Also list tables with no geometry
- ☐ Use estimated table metadata
- ☐ Allow saving/loading QGIS projects in the database

OK | Cancel | Help



Authentication

Name	osgis
Id	Generated
Resource	Optional URL resource
	Basic authentication
Username	wsosgisuser
Password	wspractialqgis42 ☑ Show
Realm	Optional

Note: Saving writes directly to authentication database

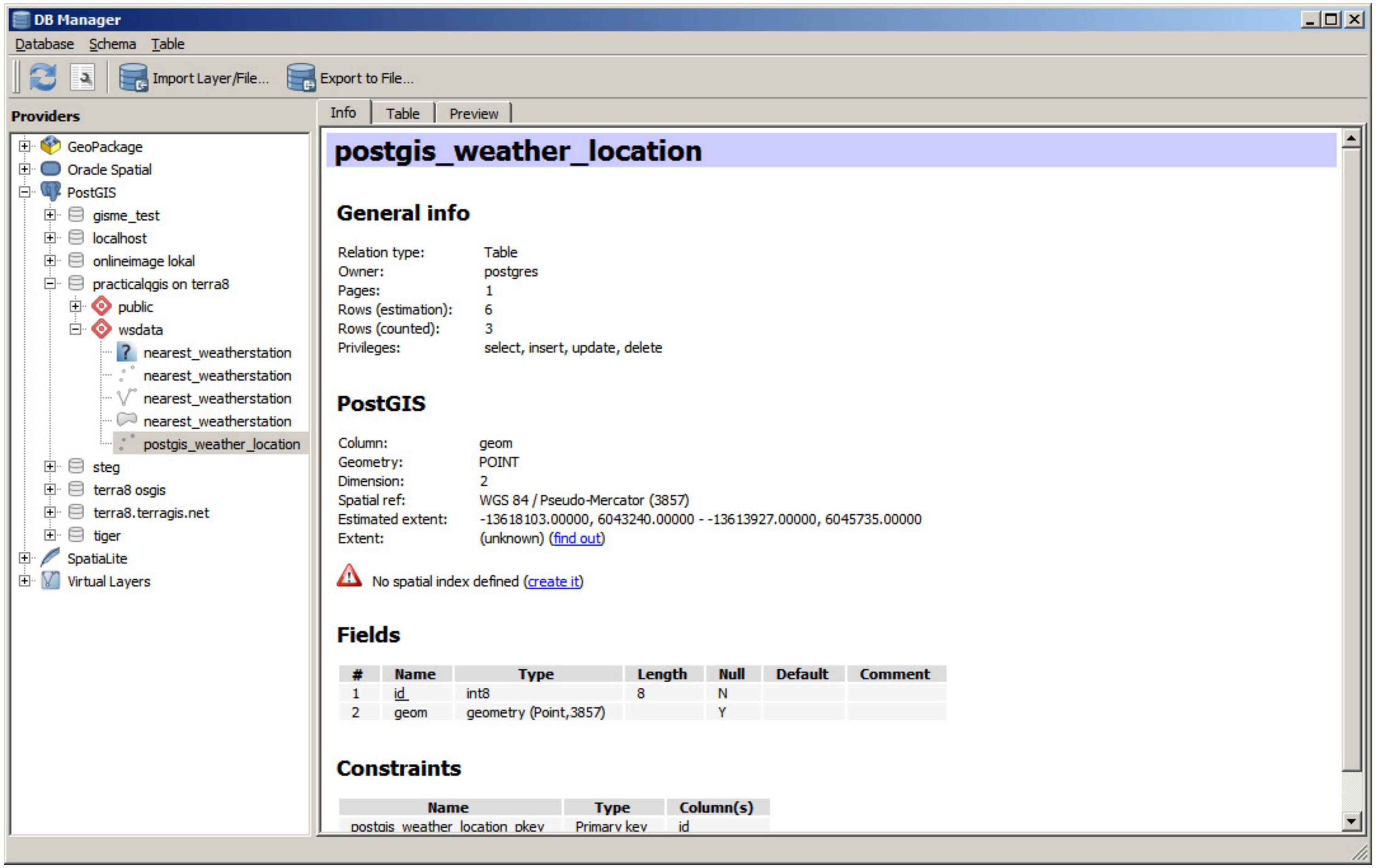
POSTGIS Connection

Host: terra8.terragis.net
Database: osgis
User: wsosgisuser
Pw: wspractialqgis42

DB Manager working with PostGIS data in QGIS

Import / export data

Dynamic query layers



postgis_weather_location

General info

Relation type:	Table
Owner:	postgres
Pages:	1
Rows (estimation):	6
Rows (counted):	3
Privileges:	select, insert, update, delete

PostGIS

Column:	geom
Geometry:	POINT
Dimension:	2
Spatial ref:	WGS 84 / Pseudo-Mercator (3857)
Estimated extent:	-13618103.00000, 6043240.00000 - -13613927.00000, 6045735.00000
Extent:	(unknown) (find out)

⚠ No spatial index defined (create it)

Fields

#	Name	Type	Length	Null	Default	Comment
1	id	int8	8	N		
2	geom	geometry (Point,3857)		Y		

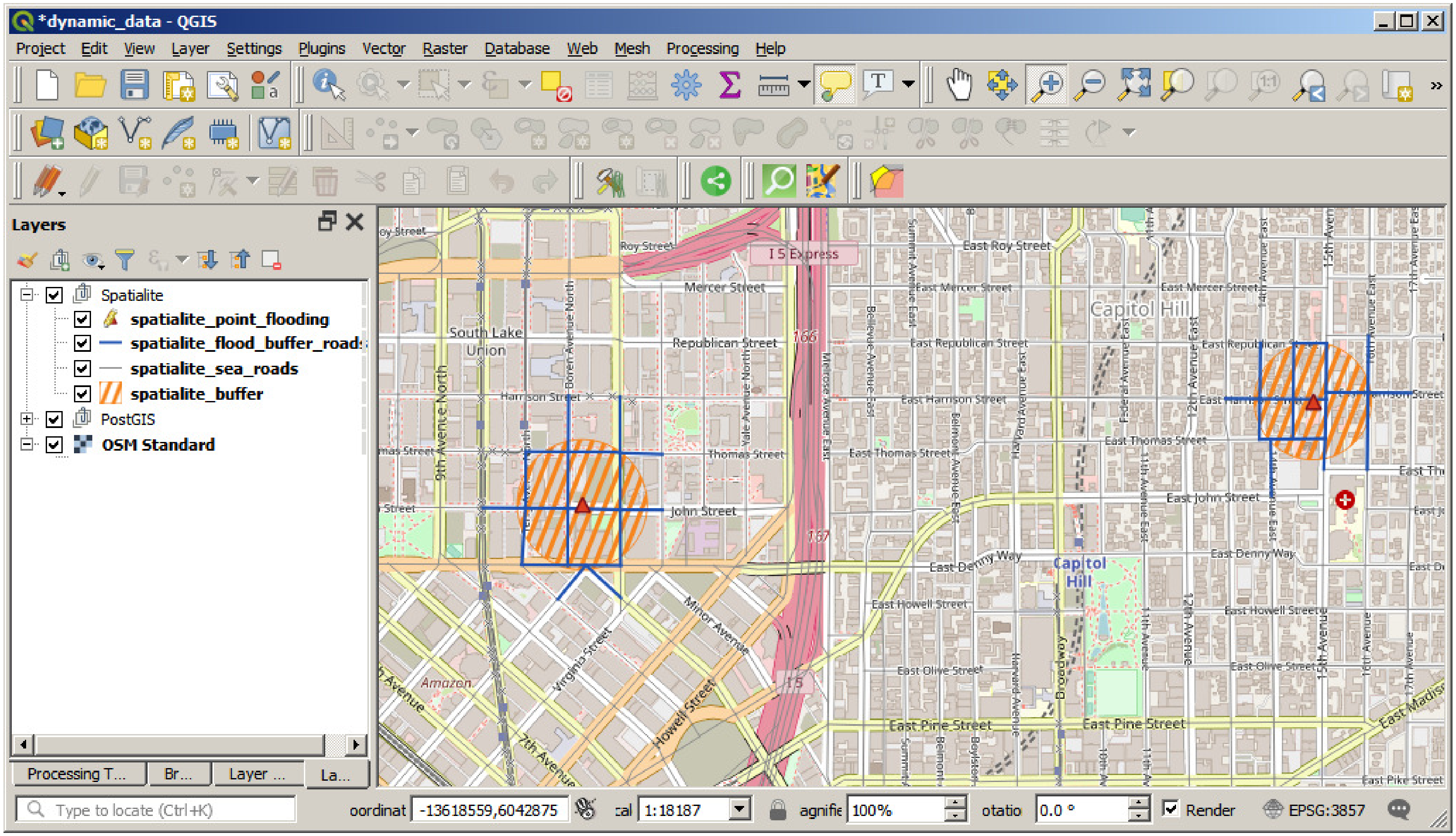
Constraints

Name	Type	Column(s)
postgis_weather_location_pkey	Primary key	id

Examples - Dynamic Layers with Spatialite or PostGIS

Project \examples\maps\dynamic_data.qgz

1. get weather data from the web and display closest weather station + measure distance



*dynamic_data - QGIS

Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help

Layers

- ☑ Spatialite
  - ☑ spatialite_point_flooding
  - ☑ spatialite_flood_buffer_road:
  - ☑ spatialite_sea_roads
  - ☑ spatialite_buffer
- ☑ PostGIS
- ☑ OSM Standard

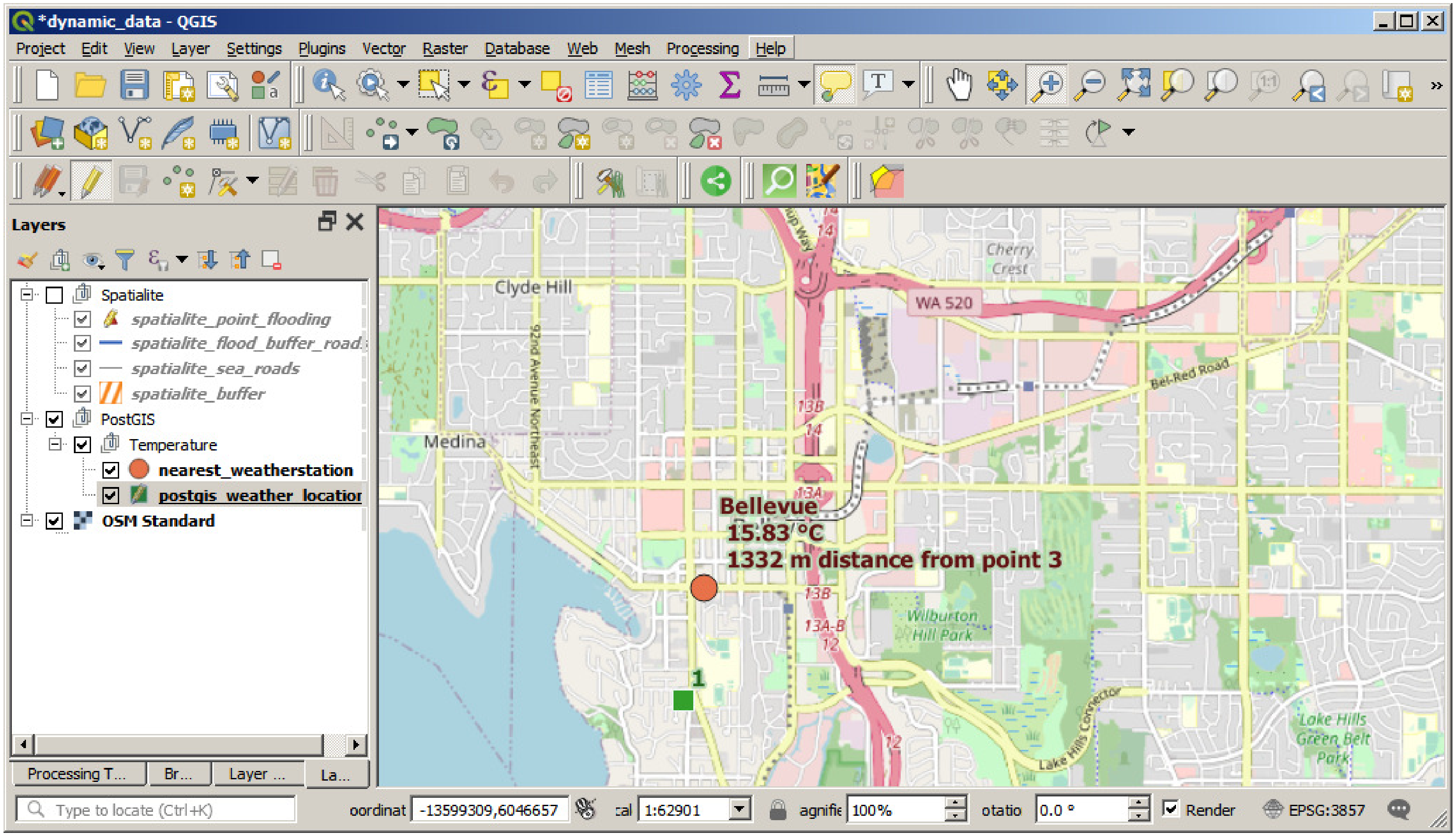
Processing T... Br... Layer ... La...

Type to locate (Ctrl+K) | oordinat -13618559,6042875 | cal 1:18187 | agnifie 100% | otatio 0.0 ° | ☑ Render | EPSG:3857

Examples - Dynamic Layers with Spatialite or PostGIS

Project \examples\maps\dynamic_data.qgz

2. Automatically buffer point to create flooded area, select and display affected roads



- □ Spatialite
  - ☑ spatialite_point_flooding
  - ☑ spatialite_flood_buffer_road
  - ☑ spatialite_sea_roads
  - ☑ spatialite_buffer
- ☑ PostGIS
  - ☑ Temperature
    - ☑ nearest_weatherstation
    - ☑ postgis_weather_locatior
- ☑ OSM Standard

Bellevue
15.83 °C
1332 m distance from point 3

Using Python and R (Statistics program) scripts

QGIS + R Package (Statistical Software)

- Official R webpage and download links
- R Studio

Articles R + Geospatial use

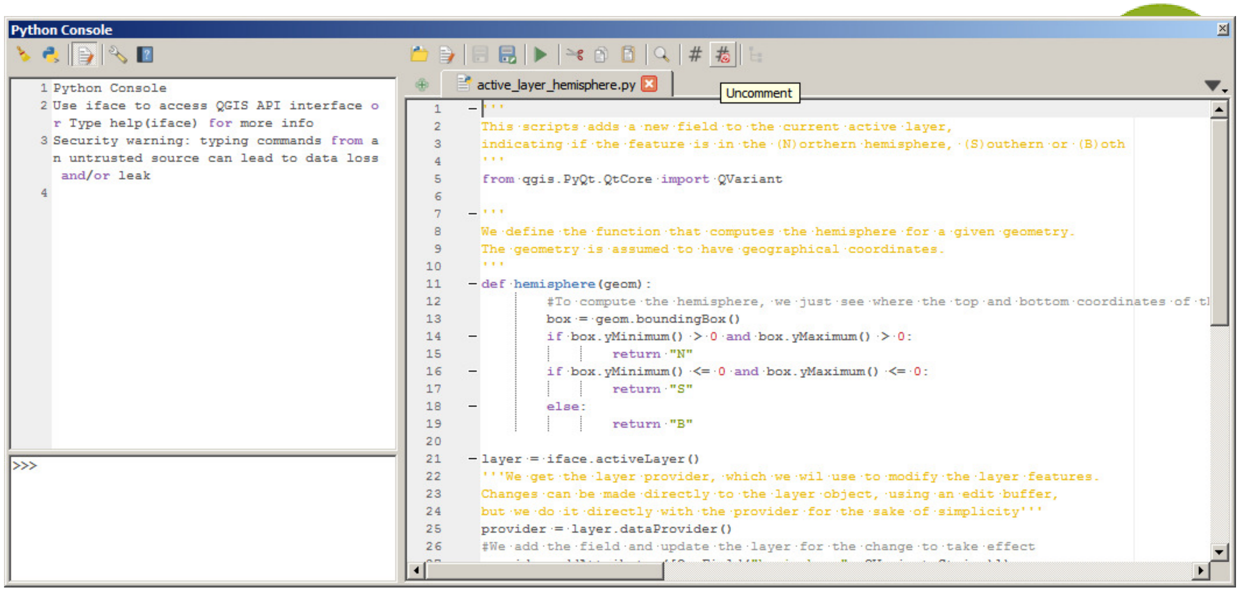
- Geospatial data in R and Beyond R, an Integrated Statistical Programming
- Environment and GIS Spatial data and R

R scripts and GIS integration

- R-scripts in Processing "Spatial data processing framework for QGIS" Plug-in (formerly Sextante plug-in)
  Needs to be enabled in Processing menu / providers:
- After R is enabled (and installed on your operating system) you can run the scripts from the Processing Toolbox – see next page
- Note that R Packages need to installed via an R interface
- QGIS will find all R packages only if present in local user folder

QGIS
Python Interface

- **Three main options**
  - Python console
  - Processing tools
  - Plugins



```
Python Console
Use iface to access QGIS API interface or Type help(iface) for more info
Security warning: typing commands from an untrusted source can lead to data loss and/or leak

>>>
```

```
'''
This scripts adds a new field to the current active layer,
indicating if the feature is in the (N)orthern hemisphere, (S)outhern or (B)oth
'''
from qgis.PyQt.QtCore import QVariant

'''
We define the function that computes the hemisphere for a given geometry.
The geometry is assumed to have geographical coordinates.
'''
def hemisphere(geom):
        #To compute the hemisphere, we just see where the top and bottom coordinates of tl
        box = geom.boundingBox()
        if box.yMinimum() > 0 and box.yMaximum() > 0:
                return "N"
        if box.yMinimum() <= 0 and box.yMaximum() <= 0:
                return "S"
        else:
                return "B"

layer = iface.activeLayer()
'''We get the layer provider, which we wil use to modify the layer features.
Changes can be made directly to the layer object, using an edit buffer,
but we do it directly with the provider for the sake of simplicity'''
provider = layer.dataProvider()
#We add the field and update the layer for the change to take effect
```

- **Good overview about Python in QGIS**

  Understanding Python in QGIS - Victor Olaya

  https://github.com/volaya/qgis-python-course

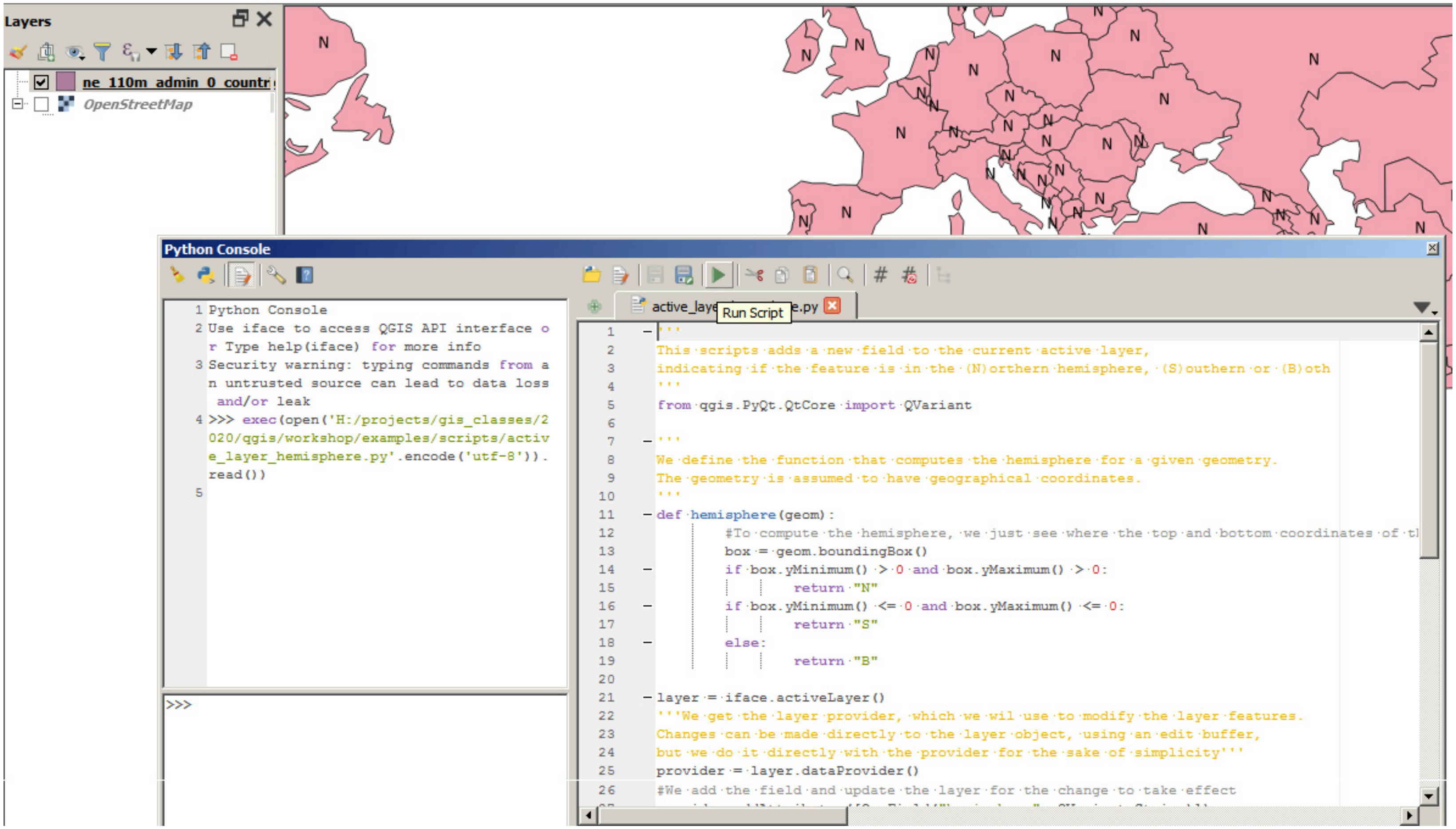
- **Tools such as**
  - WinPython (IDE + QT Designer)
  - QT Designer
  - *Plugin-Builder 3* **Plugin**
    Creates a QGIS plug-in template
    for use as a starting point in plug-in development

Example Python Console

Example to label countries with custom function

Project \examples\maps\countries_label_hemisphere.qgz

Code \examples\scripts\active_layer_hemisphere.py

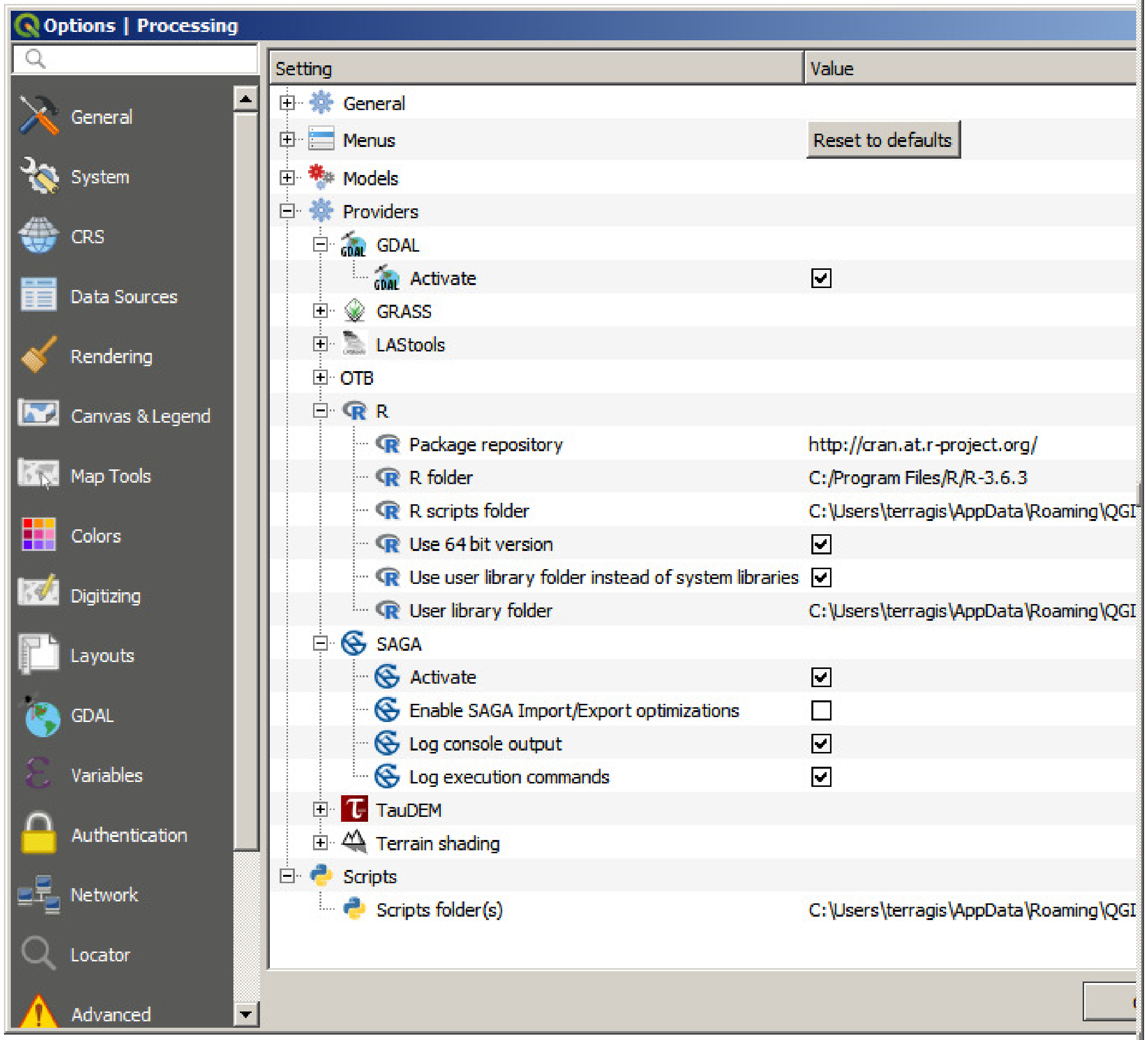


```
Python Console
Use iface to access QGIS API interface or Type help(iface) for more info
Security warning: typing commands from an untrusted source can lead to data loss and/or leak
>>> exec(open('H:/projects/gis_classes/2020/qgis/workshop/examples/scripts/active_layer_hemisphere.py'.encode('utf-8')).read())

>>>
```

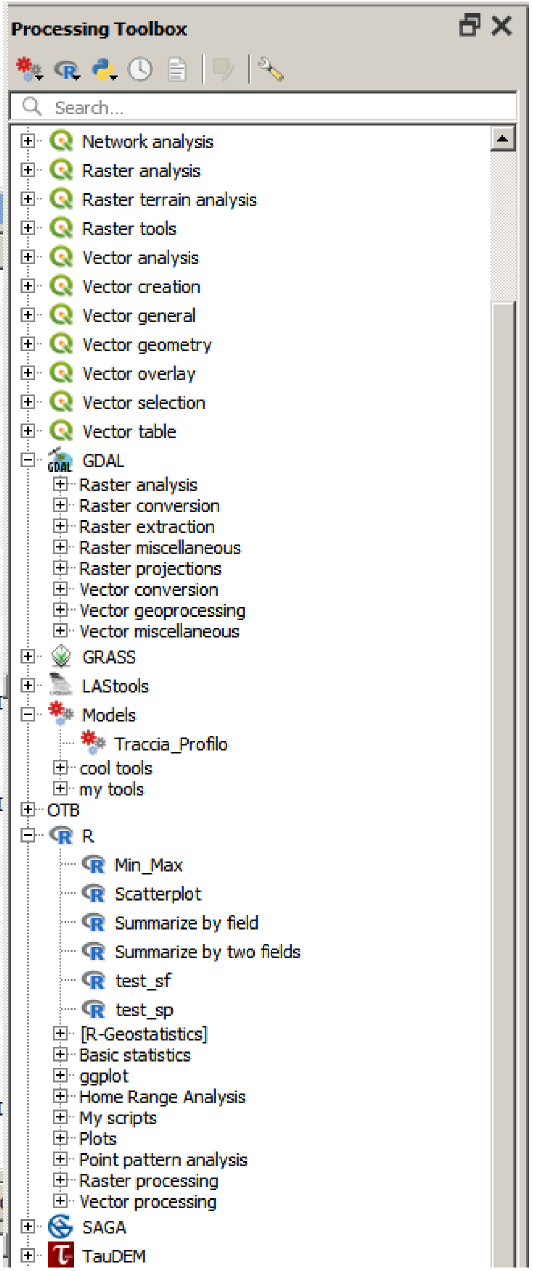
```
'''
This scripts adds a new field to the current active layer,
indicating if the feature is in the (N)orthern hemisphere, (S)outhern or (B)oth
'''
from qgis.PyQt.QtCore import QVariant

'''
We define the function that computes the hemisphere for a given geometry.
The geometry is assumed to have geographical coordinates.
'''
def hemisphere(geom):
        #To compute the hemisphere, we just see where the top and bottom coordinates of tl
        box = geom.boundingBox()
        if box.yMinimum() > 0 and box.yMaximum() > 0:
                return "N"
        if box.yMinimum() <= 0 and box.yMaximum() <= 0:
                return "S"
        else:
                return "B"

layer = iface.activeLayer()
'''We get the layer provider, which we wil use to modify the layer features.
Changes can be made directly to the layer object, using an edit buffer,
but we do it directly with the provider for the sake of simplicity'''
provider = layer.dataProvider()
#We add the field and update the layer for the change to take effect
```

Example Python and R in Processing

Enabling Providers

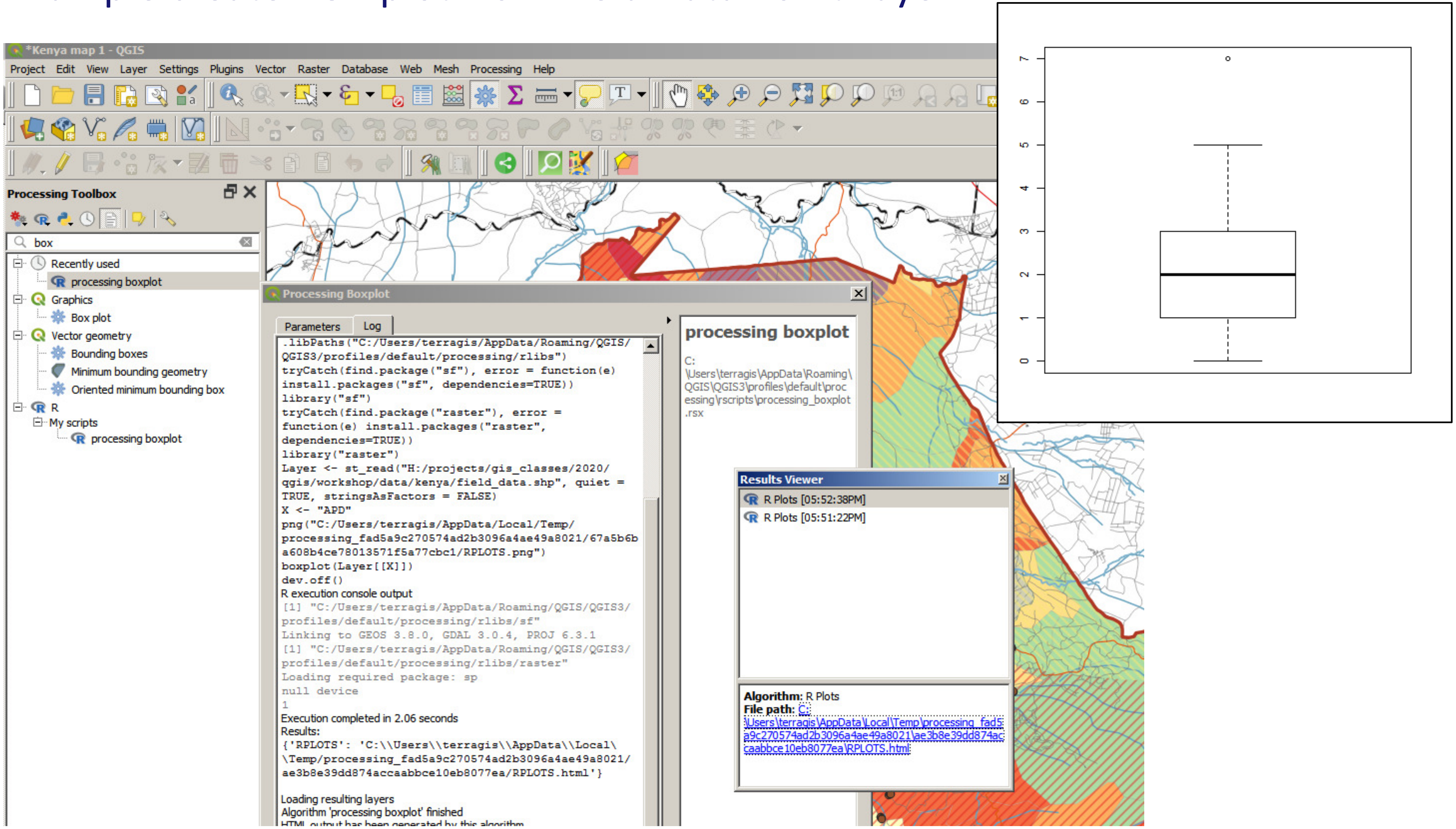


Setting	Value
General	
Menus	Reset to defaults
Models	
Providers	
GDAL	
Activate	☑
GRASS	
LAStools	
OTB	
R	
Package repository	http://cran.at.r-project.org/
R folder	C:/Program Files/R/R-3.6.3
R scripts folder	C:\Users\terragis\AppData\Roaming\QGI
Use 64 bit version	☑
Use user library folder instead of system libraries	☑
User library folder	C:\Users\terragis\AppData\Roaming\QGI
SAGA	
Activate	☑
Enable SAGA Import/Export optimizations	☐
Log console output	☑
Log execution commands	☑
TauDEM	
Terrain shading	
Scripts	
Scripts folder(s)	C:\Users\terragis\AppData\Roaming\QGI



Example R Processing

Example create BOX plot from Field Data Point Layer



```
.libPaths("C:/Users/terragis/AppData/Roaming/QGIS/
QGIS3/profiles/default/processing/rlibs")
tryCatch(find.package("sf"), error = function(e)
install.packages("sf", dependencies=TRUE))
library("sf")
tryCatch(find.package("raster"), error =
function(e) install.packages("raster",
dependencies=TRUE))
library("raster")
Layer <- st_read("H:/projects/gis_classes/2020/
qgis/workshop/data/kenya/field_data.shp", quiet =
TRUE, stringsAsFactors = FALSE)
X <- "APD"
png("C:/Users/terragis/AppData/Local/Temp/
processing_fad5a9c270574ad2b3096a4ae49a8021/67a5b6b
a608b4ce78013571f5a77cbc1/RPLOTS.png")
boxplot(Layer[[X]])
dev.off()
R execution console output
[1] "C:/Users/terragis/AppData/Roaming/QGIS/QGIS3/
profiles/default/processing/rlibs/sf"
Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
[1] "C:/Users/terragis/AppData/Roaming/QGIS/QGIS3/
profiles/default/processing/rlibs/raster"
Loading required package: sp
null device
1
Execution completed in 2.06 seconds
Results:
{'RPLOTS': 'C:\\Users\\terragis\\AppData\\Local\
\Temp/processing_fad5a9c270574ad2b3096a4ae49a8021/
ae3b8e39dd874accaabbce10eb8077ea/RPLOTS.html'}

Loading resulting layers
Algorithm 'processing boxplot' finished
HTML output has been generated by this algorithm
```

processing boxplot

C:
\Users\terragis\AppData\Roaming\QGIS\QGIS3\profiles\default\processing\rscripts\processing_boxplot.rsx

Results Viewer

R Plots [05:52:38PM]

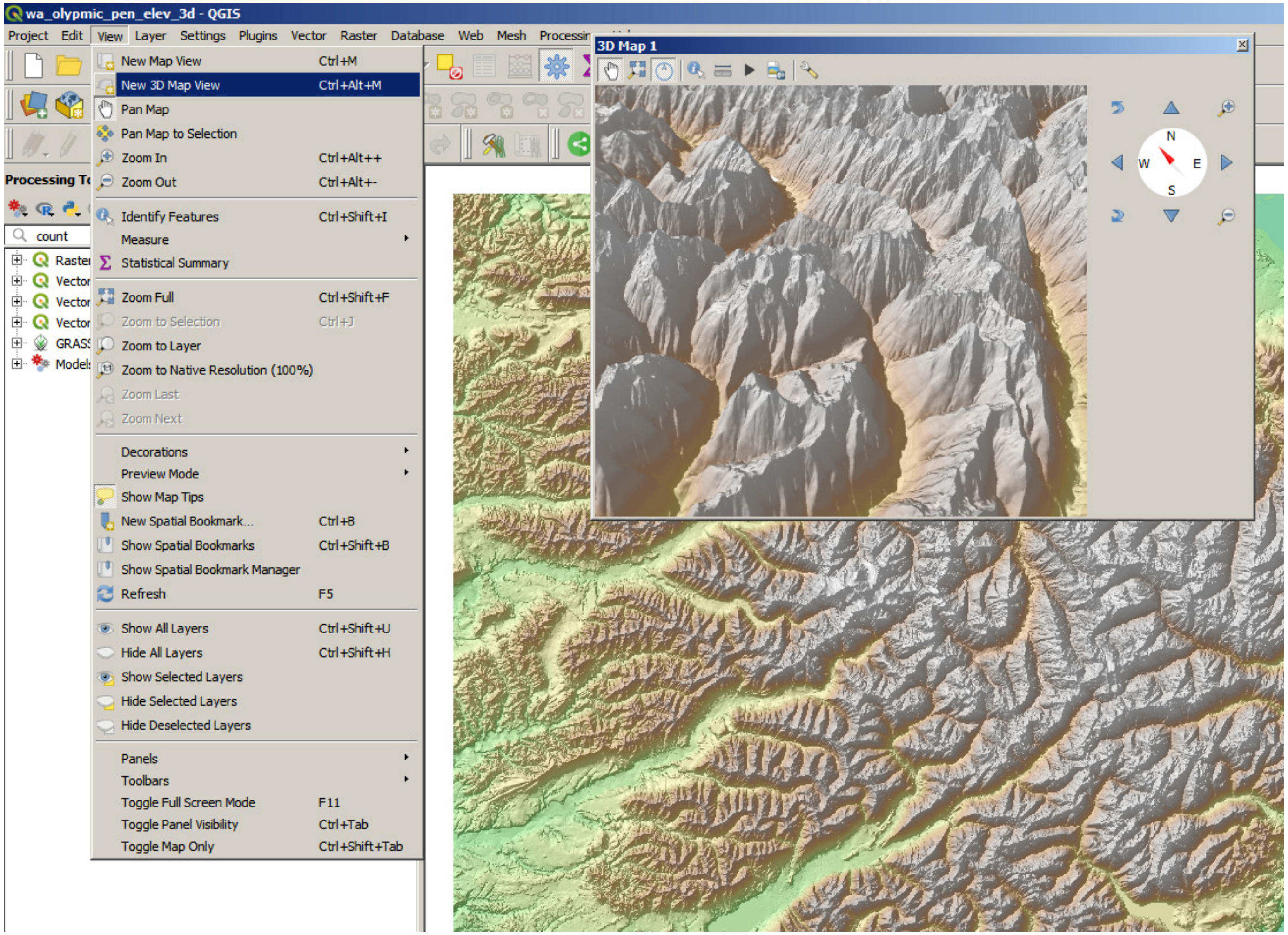
R Plots [05:51:22PM]

Algorithm: R Plots
File path: C:\Users\terragis\AppData\Local\Temp\processing_fad5a9c270574ad2b3096a4ae49a8021\ae3b8e39dd874accaabbce10eb8077ea\RPLOTS.html

Additional Notes

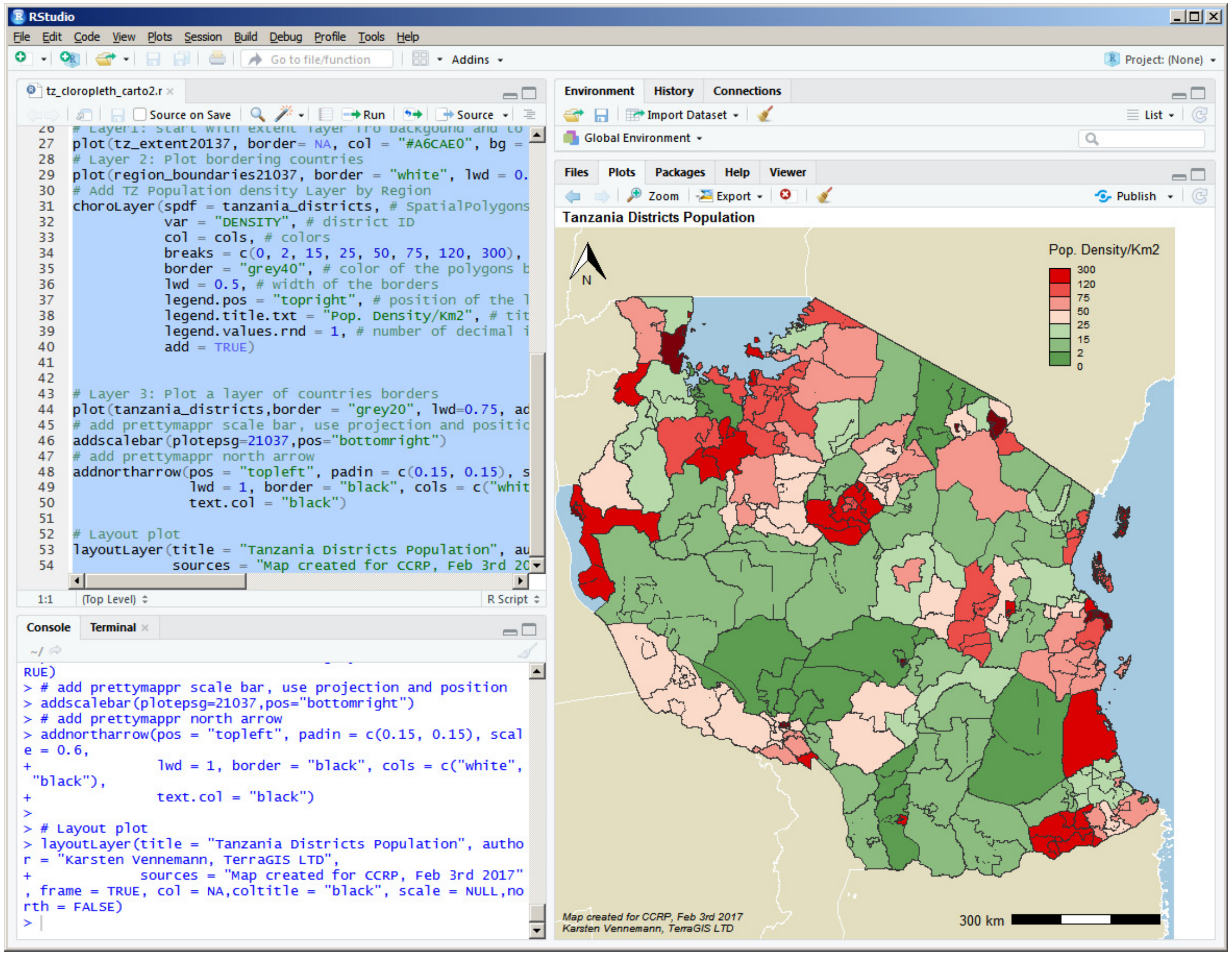
Example 3D View

Project \examples\maps\wa_olypmic_pen_elev_3d.qgz



Example Map in RStudio (no QGIS needed)

\examples\scripts\more\tz_cloropleth_carto2.r



```r
# Layer1: Start with extent layer fro backgound and to
plot(tz_extent20137, border= NA, col = "#A6CAE0", bg =
# Layer 2: Plot bordering countries
plot(region_boundaries21037, border = "white", lwd = 0.
# Add TZ Population density Layer by Region
choroLayer(spdf = tanzania_districts, # SpatialPolygons
           var = "DENSITY", # district ID
           col = cols, # colors
           breaks = c(0, 2, 15, 25, 50, 75, 120, 300),
           border = "grey40", # color of the polygons b
           lwd = 0.5, # width of the borders
           legend.pos = "topright", # position of the l
           legend.title.txt = "Pop. Density/Km2", # tit
           legend.values.rnd = 1, # number of decimal i
           add = TRUE)


# Layer 3: Plot a layer of countries borders
plot(tanzania_districts,border = "grey20", lwd=0.75, ad
# add prettymappr scale bar, use projection and positic
addscalebar(plotepsg=21037,pos="bottomright")
# add prettymappr north arrow
addnortharrow(pos = "topleft", padin = c(0.15, 0.15), s
              lwd = 1, border = "black", cols = c("whit
              text.col = "black")

# Layout plot
layoutLayer(title = "Tanzania Districts Population", au
            sources = "Map created for CCRP, Feb 3rd 20
```

```
RUE)
> # add prettymappr scale bar, use projection and position
> addscalebar(plotepsg=21037,pos="bottomright")
> # add prettymappr north arrow
> addnortharrow(pos = "topleft", padin = c(0.15, 0.15), scal
e = 0.6,
+               lwd = 1, border = "black", cols = c("white",
 "black"),
+               text.col = "black")
>
> # Layout plot
> layoutLayer(title = "Tanzania Districts Population", autho
r = "Karsten Vennemann, TerraGIS LTD",
+             sources = "Map created for CCRP, Feb 3rd 2017"
, frame = TRUE, col = NA,coltitle = "black", scale = NULL,no
rth = FALSE)
> |
```

QGIS – Metadata

viewing, editing, exporting, etc

What are our options for documenting database or shape files with metadata?

- Starting with QGIS 3.6 one can enter information about the layer via the layer properties window and save metadata as a file (QMD file) or "Save as Default“ right click layer, choose Metadata tab

QGIS - Joining Tables

- tutorial http://www.qgistutorials.com/en/docs/performing_table_joins.html

QGIS - Building a Map Atlas

- tutorial https://docs.qgis.org/3.10/en/docs/training_manual/forestry/forest_maps.html

QGIS – Topology

- step-by-step tutorial: https://docs.qgis.org/3.10/en/docs/training_manual/create_vector_data/topo_editing.html

QGIS - Grass integration - two options:

- Using the Processing plug-in (Spatial data processing framework for QGIS) is easy to use
- Using the Grass plug-in - has more functions but is more difficult to use installation via OSGeo4W "advanced" mode