



# Procesamiento de series de tiempo en **GRASS GIS**

## Aplicaciones en Ecología y Ambiente

Dra. Verónica Andreo  
CONICET - INMeT

Río Cuarto, 2018



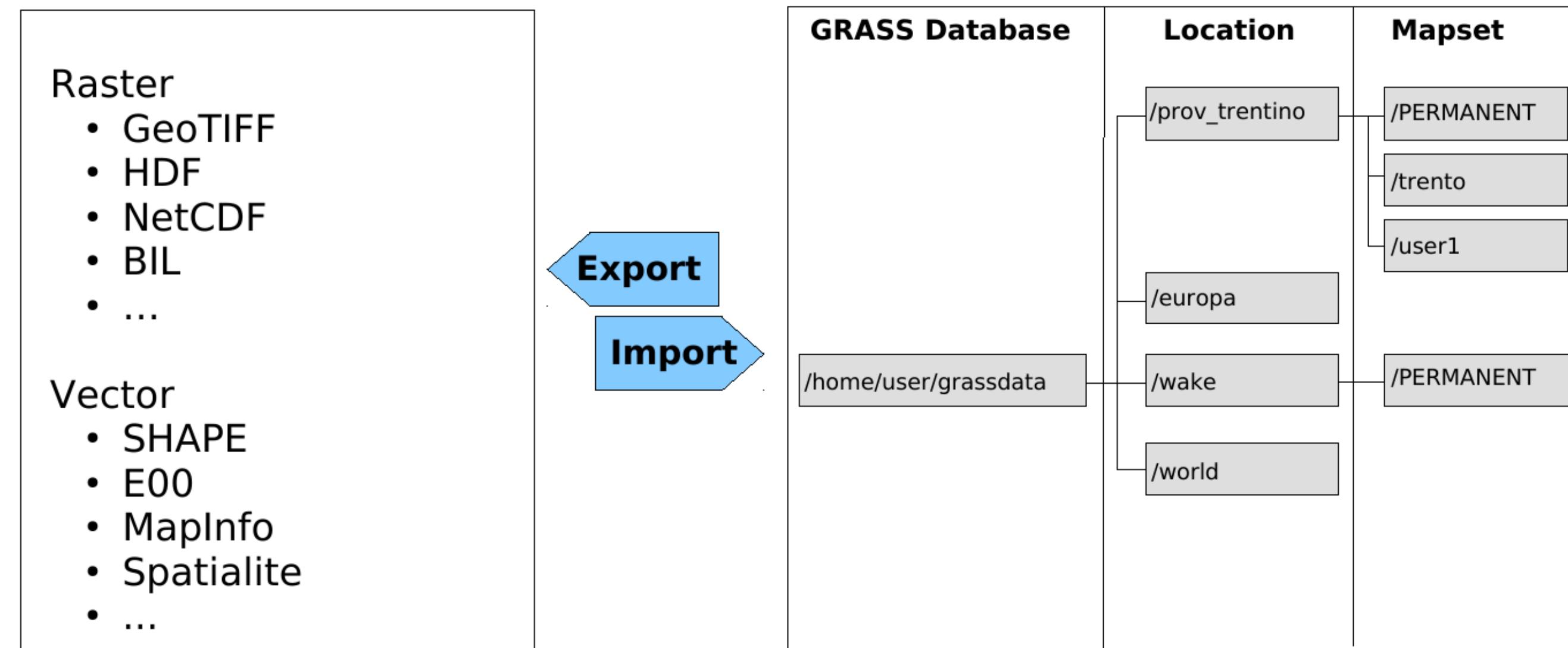
# GRASS GIS: Overview of general capabilities





# Interoperability





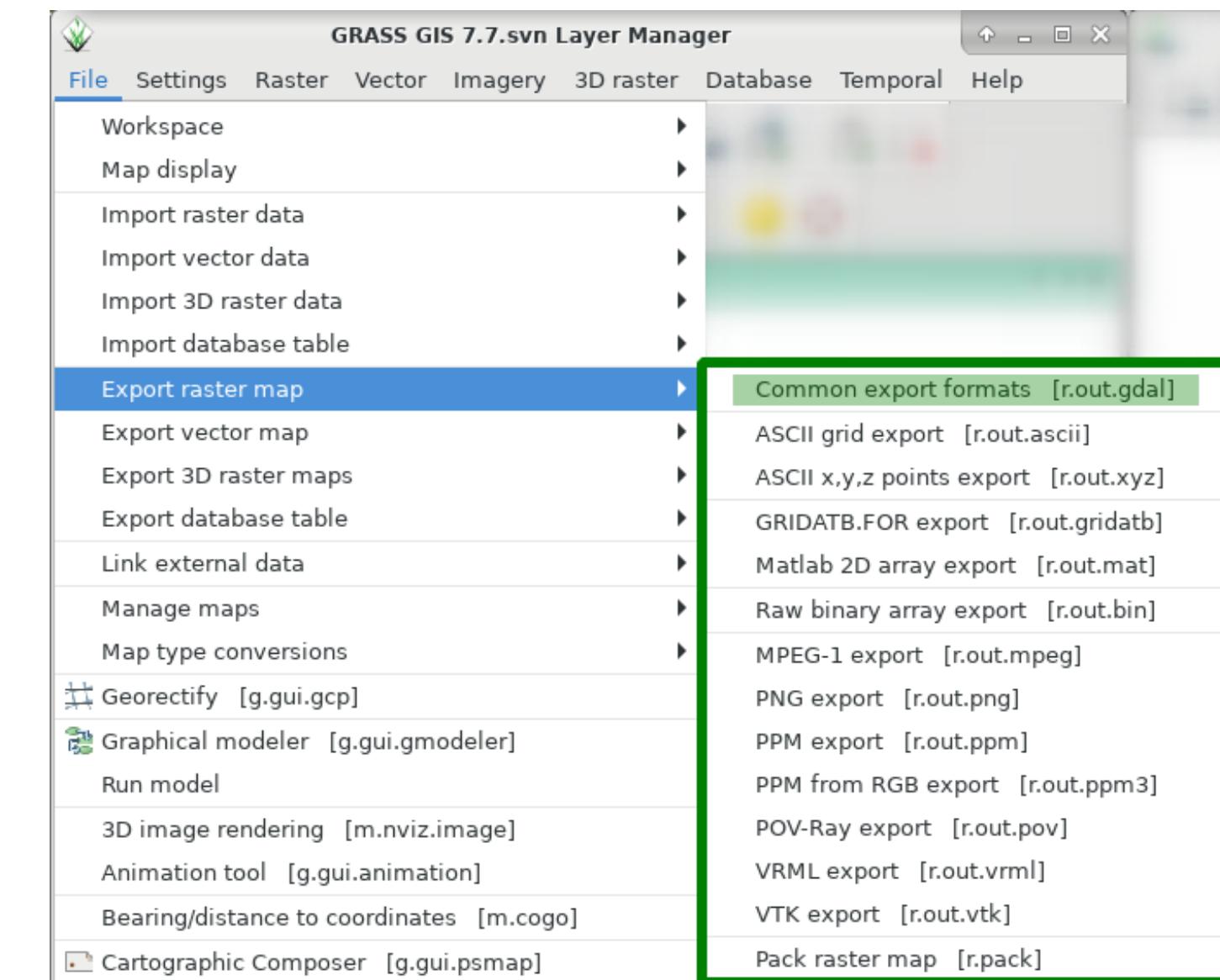
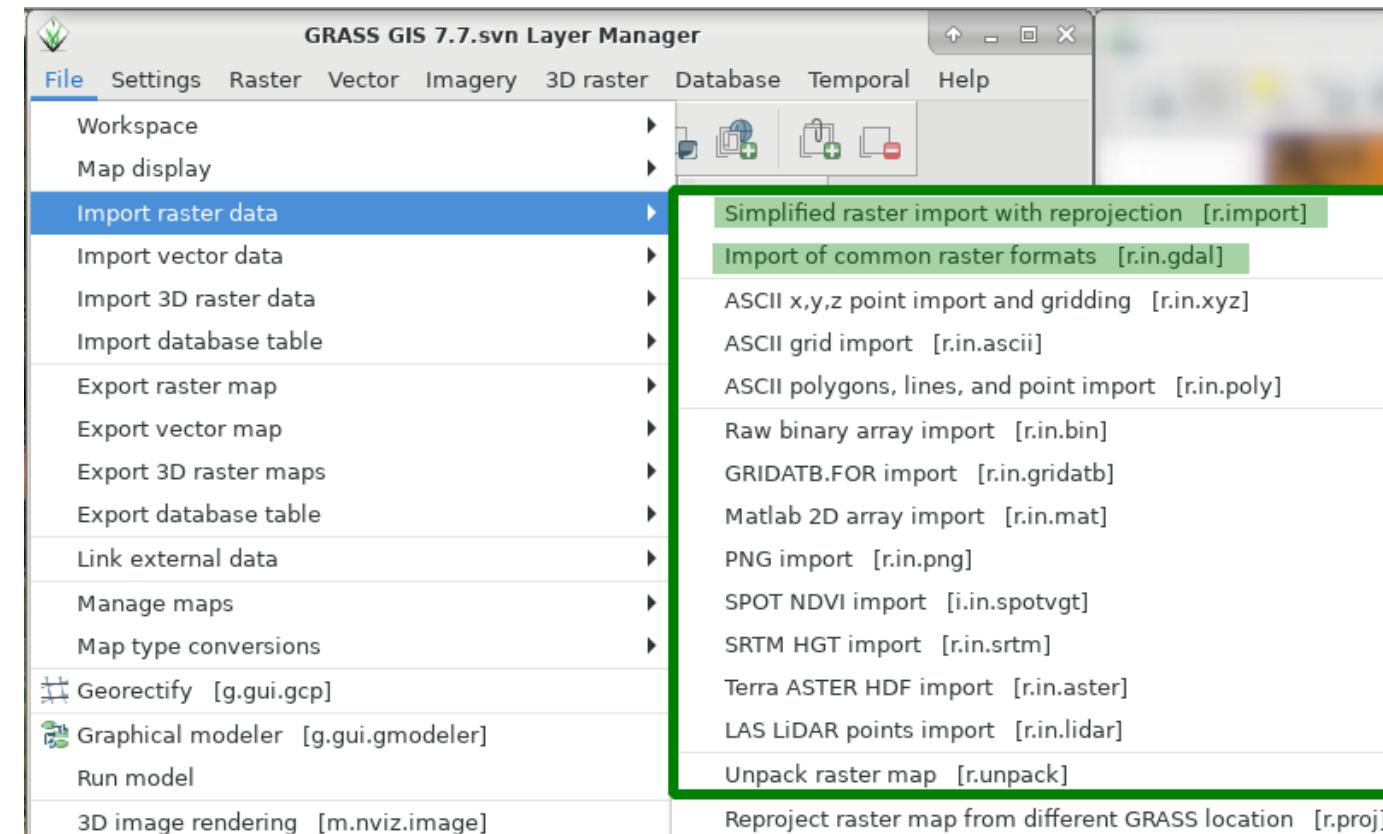
Data in standard GIS formats.

Store in directory:  
/home/user/gisdata/  
or a shared network directory

Data in GRASS GIS formats.

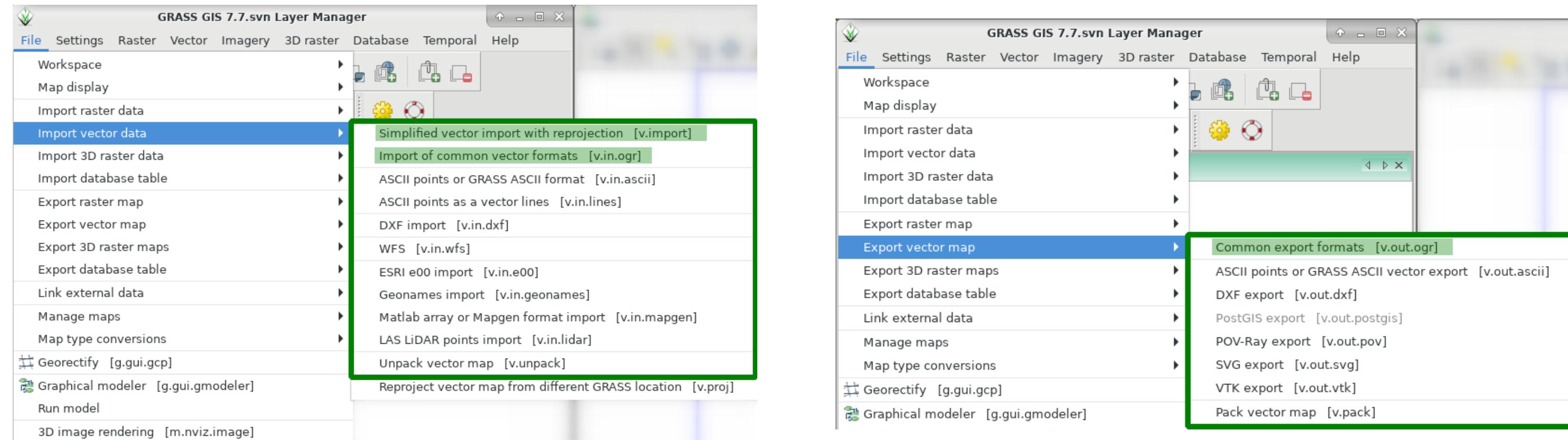
Store in directory:  
/home/user/grassdata/  
or a shared network directory

# Modules for import/export of raster maps



GRASS relies on **GDAL** to import and export raster data

# Modules for import/export of vector maps

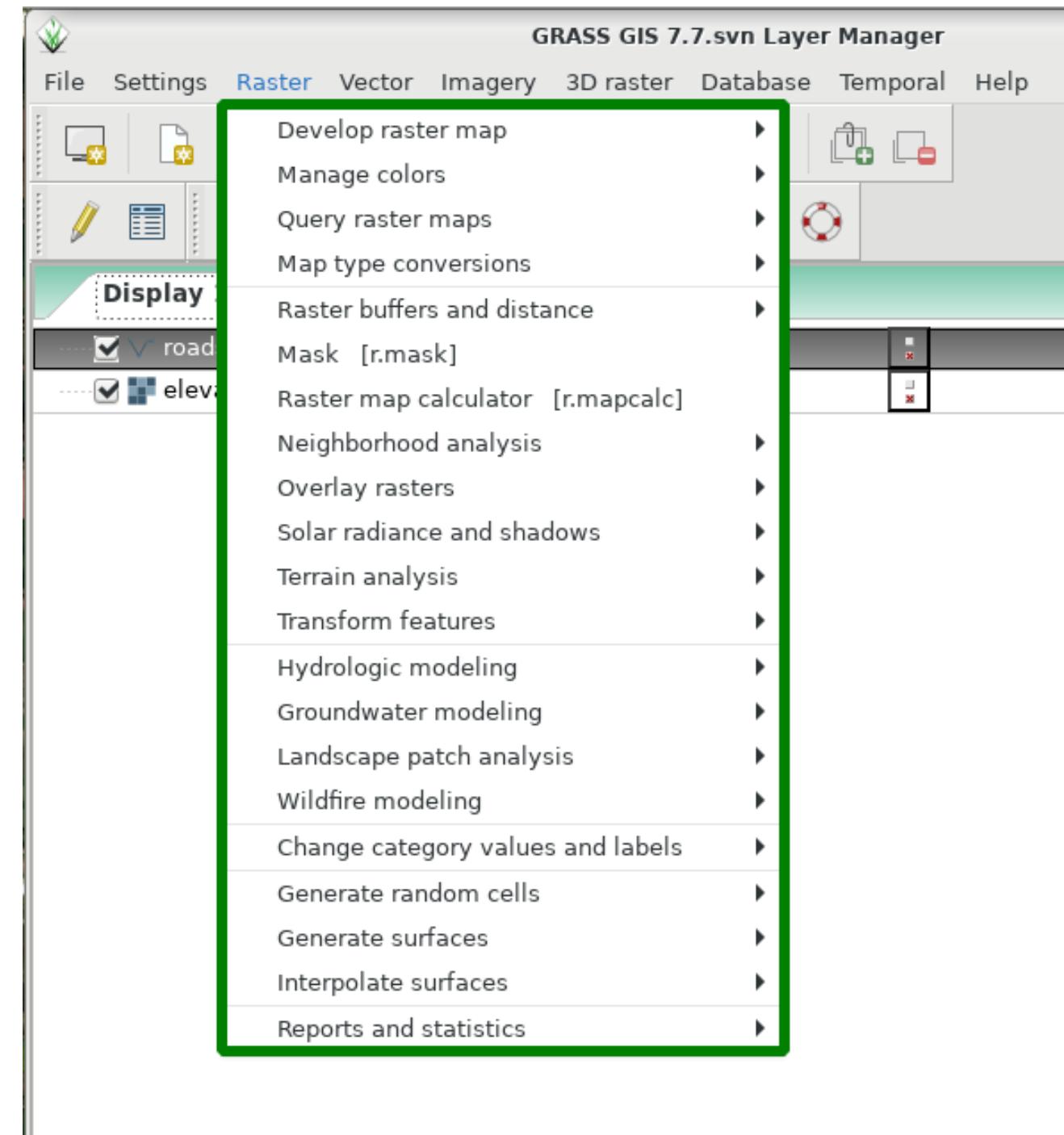


GRASS relies on OGR to import and export vector data

# Raster data processing



# Raster menu

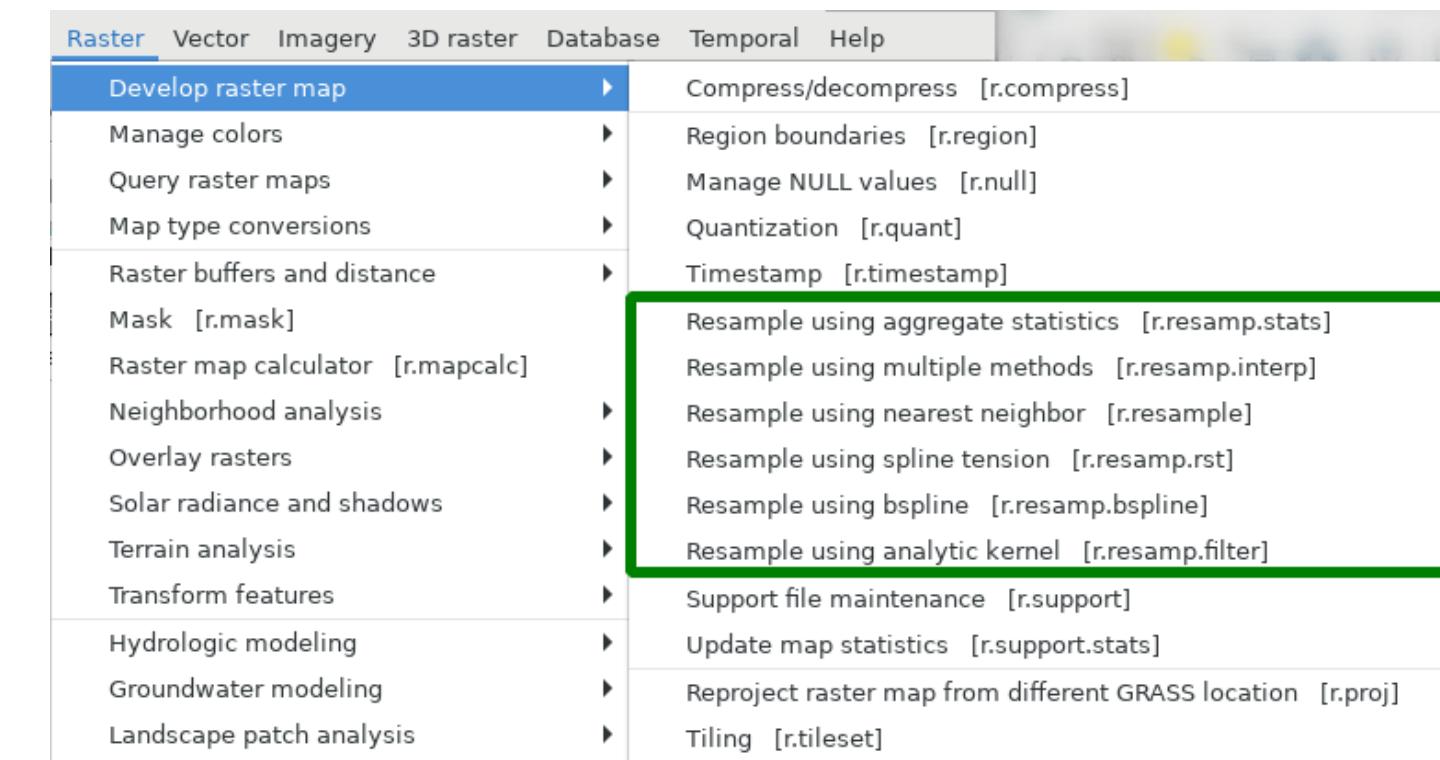


- Raster data: DEM, land cover, climatic maps, etc.
- Imagery data: Landsat, Sentinel, MODIS, SPOT, QuickBird, etc.

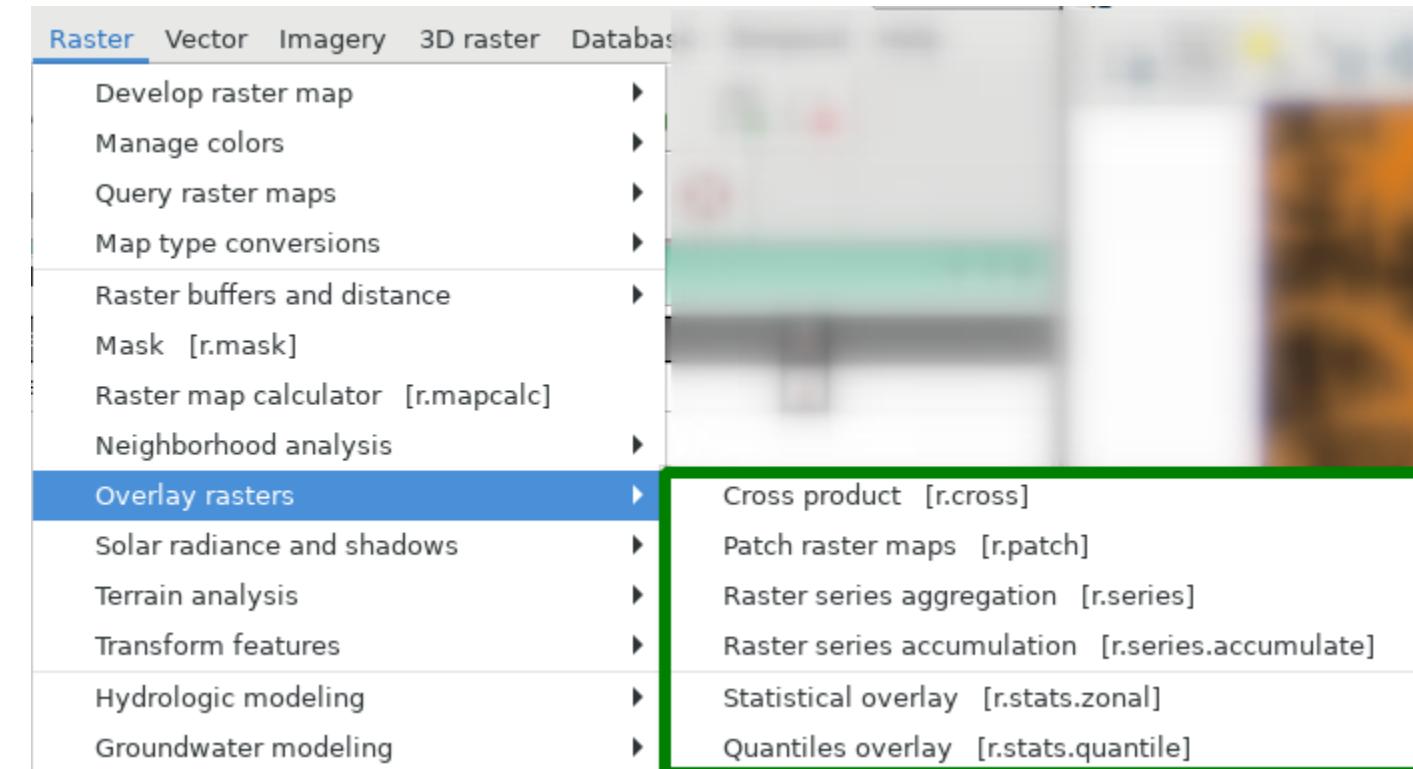
## Raster processing manual

# Resampling

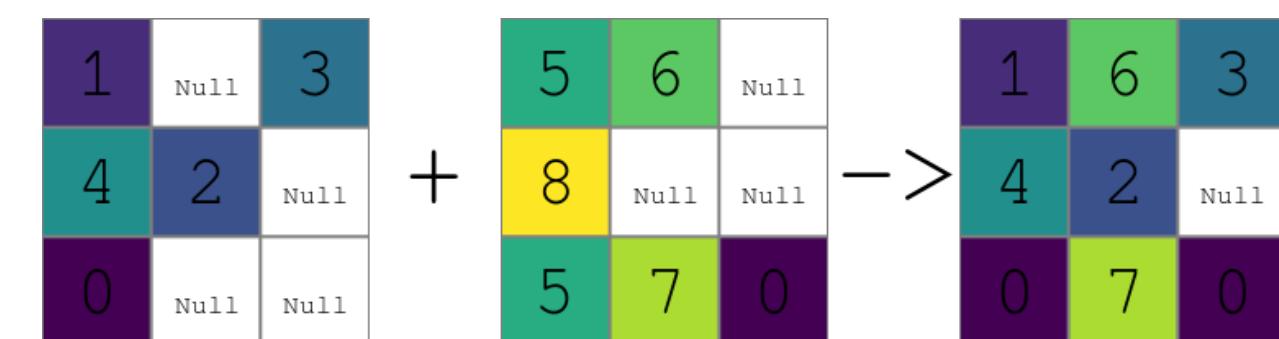
- **r.resamp.interp**: Resamples raster map to a finer grid using different interpolation methods: nearest, bilinear, bicubic (downscaling)
- **r.resamp.stats**: Resamples raster map layers to a coarser grid using aggregation (i.e., upscaling)



# Raster overlay



- **r.series**: Allows to aggregate a list of maps with different methods, i.e., average, min, max, etc.
- **r.patch**: Creates a composite raster map using category values from one (or more) map(s) to fill in areas of "no data" in another map



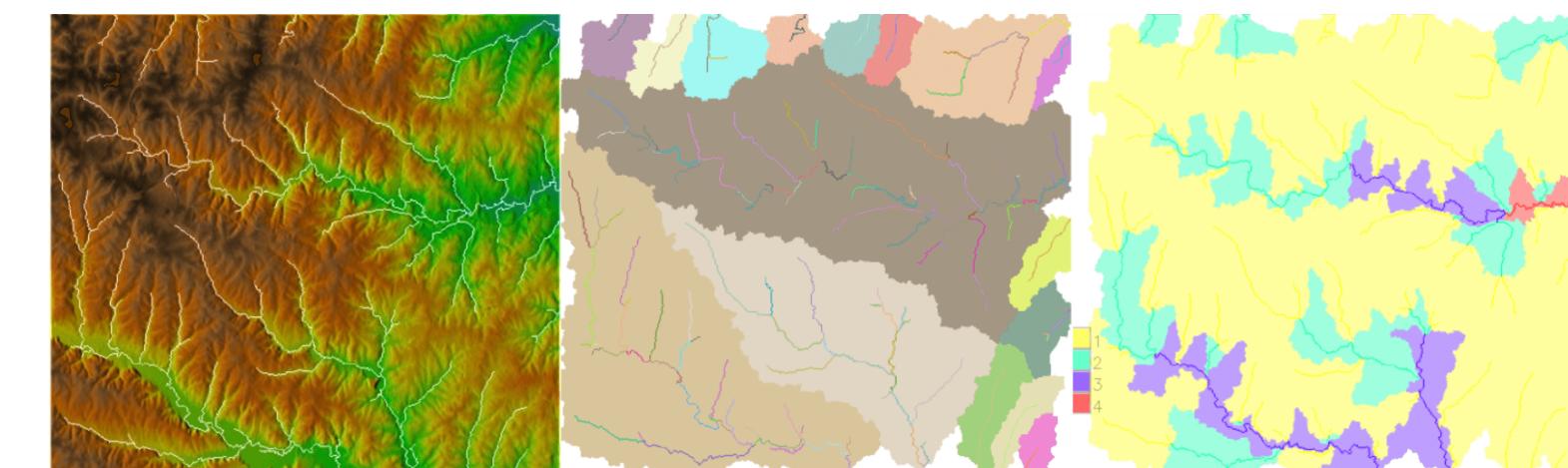
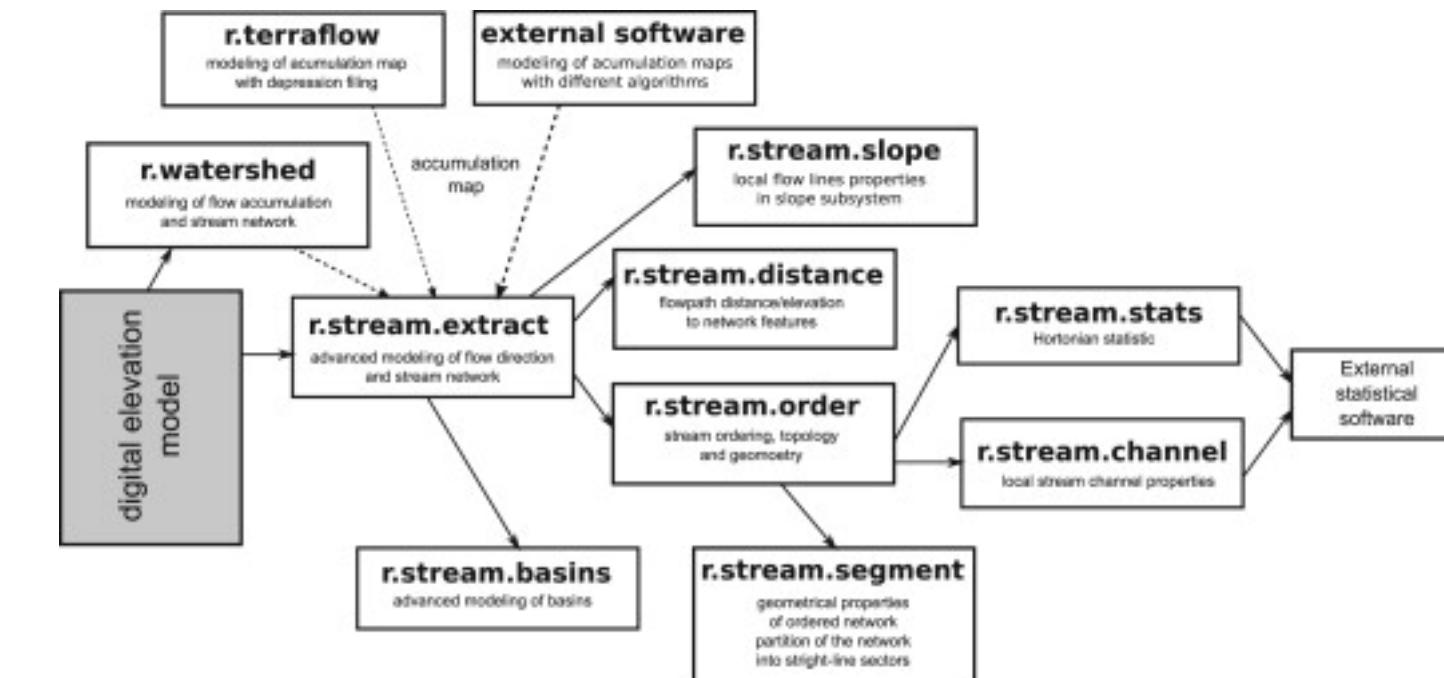
# Hydrological modeling

Hydrologic modeling

- ▶ Groundwater modeling
- ▶ Landscape patch analysis
- ▶ Wildfire modeling
- ▶ Change category values and labels
- ▶ Generate random cells
- ▶ Generate surfaces
- ▶ Interpolate surfaces
- ▶ Reports and statistics

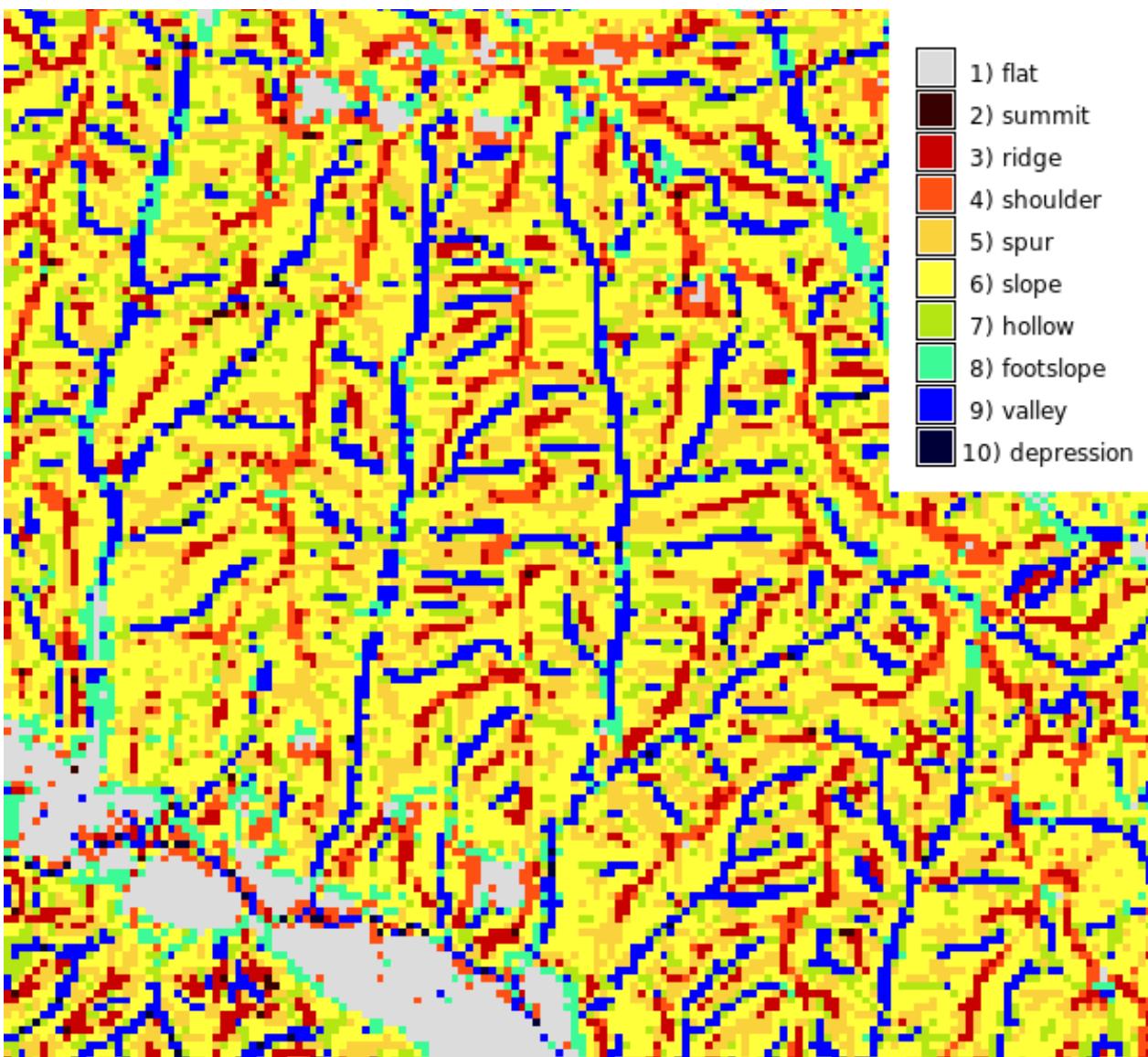
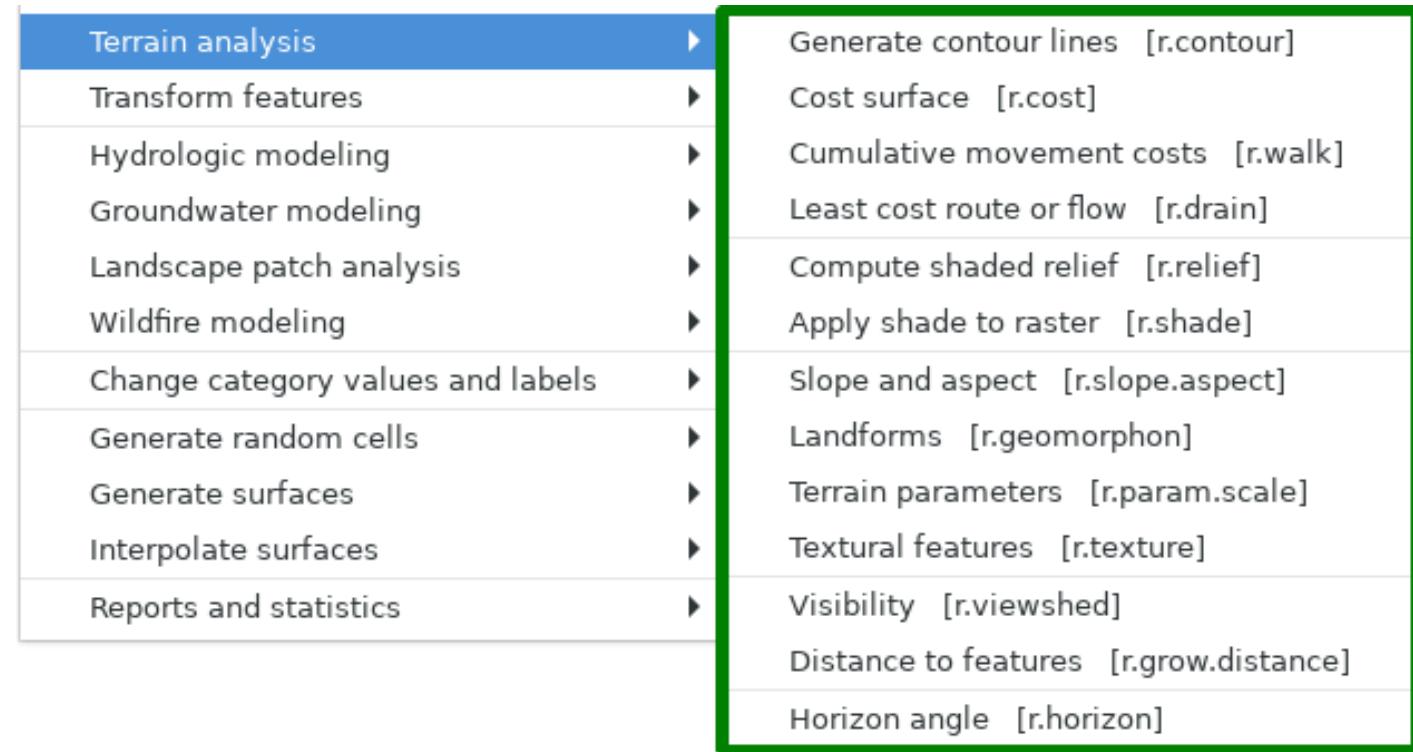
- Carve stream channels [r.carve]
- Fill lake [r.lake]
- Depressionless map and flowlines [r.fill.dir]
- Flow accumulation [r.terraflow]
- Flow lines [r.flow]
- Watershed analysis [r.watershed]
- Watershed subbasins [r.basins.fill]
- Watershed basin creation [r.water.outlet]
- Extraction of stream networks [r.stream.extract]
- SIMWE Overland flow modeling [r.sim.water]
- SIMWE Sediment flux modeling [r.sim.sediment]
- Topographic index map [r.topidx]
- TOPMODEL simulation [r.topmodel]
- USLE K-factor [r.uslek]
- USLE R-factor [r.usler]

... plus several other add-ons, for example:



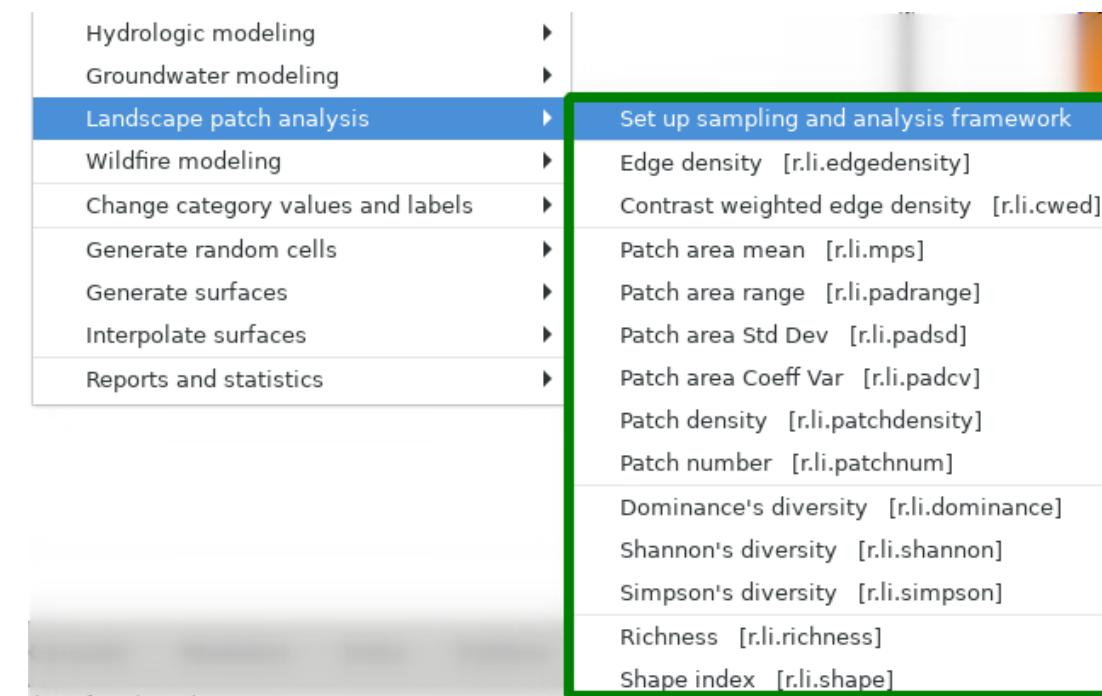
Jasiewics and Metz, 2011

# Terrain analysis



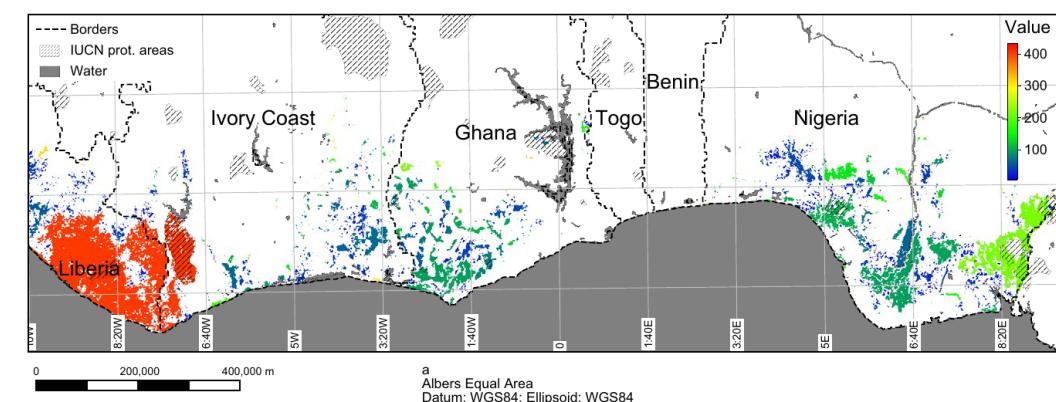
Output of `r.geomorphon`

# Landscape analysis



... plus several add-ons for "patch analysis"

- *r.pi.rectangle* - Performs statistical analysis on values of patches from the given raster map.
- *r.pi.energy* - Individual-based dispersal model for connectivity analysis - energy based.
- *r.pi.energy.pr* - Individual-based dispersal model for connectivity analysis (energy based) using iterative patch removal.
- *r.pi.fragment.dist* - Calculates correlation of two raster maps by calculating correlation function of two corresponding rectangular areas for each raster point and writing the result into a new raster map.
- *r.pi.enn* - Determines patches of given value and performs a nearest-neighbor analysis.
- *r.pi.index* - Computation of fragmentation indices.
- *r.pi.enn.pr* - Patch relevance for Euclidean Nearest Neighbor patches.
- *r.pi.neigh* - Neighbourhood analysis - value of patches within a defined range.
- *r.pi.enn* - Analysis of n-th Euclidean Nearest Neighbor distance.
- *r.pi.nlm* - Creates a random generated map with values 0 or 1 by given landcover and fragment count.
- *r.pi.nlm.circ* - Creates a random landscape with defined attributes.
- *r.pi.nlm.stats* - Neutral Landscape Generator - index statistics
- *r.pi.corearea* - Variable edge effects and core area analysis
- *r.pi.corr:mw* - Moving window correlation analysis.
- *r.pi.csr:mw* - Complete Spatial Randomness analysis on moving window.
- *r.pi.export* - Export of patch based information.
- *r.pi.graph* - Graph Theory for connectivity analysis.
- *r.pi.graph.pr* - Graph Theory - iterative removal (patch relevance analysis).
- *r.pi.graph.red* - Graph Theory - decreasing distance threshold option.
- *r.pi.grow* - Size and suitability based region growing.
- *r.pi.import* - Import and generation of patch raster data
- *r.pi.index* - Basic patch based indices
- *r.pi.lm* - Linear regression analysis for patches.
- *r.pi.prob.mw* - Probability analysis of 2 random points being in the same patch.
- *r.pi.rectangle* - Generates a rectangle based on a corner coordinate.
- *r.pi.searchtime* - Individual-based dispersal model for connectivity analysis (time-based)
- *r.pi.searchtime.pr* - Individual-based dispersal model for connectivity analysis (time-based) using iterative removal of patches
- *r.pi.searchtime.mw* - Individual-based dispersal model for connectivity analysis (time-based) using moving window

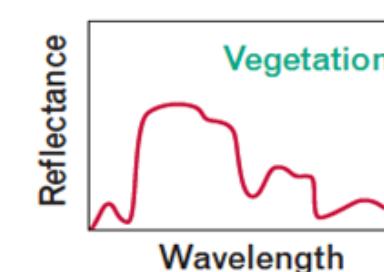
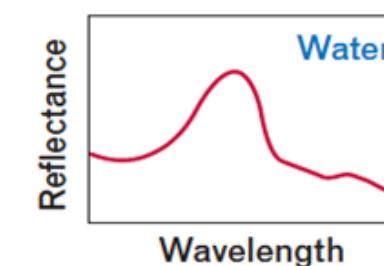
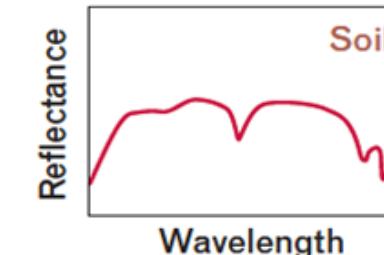
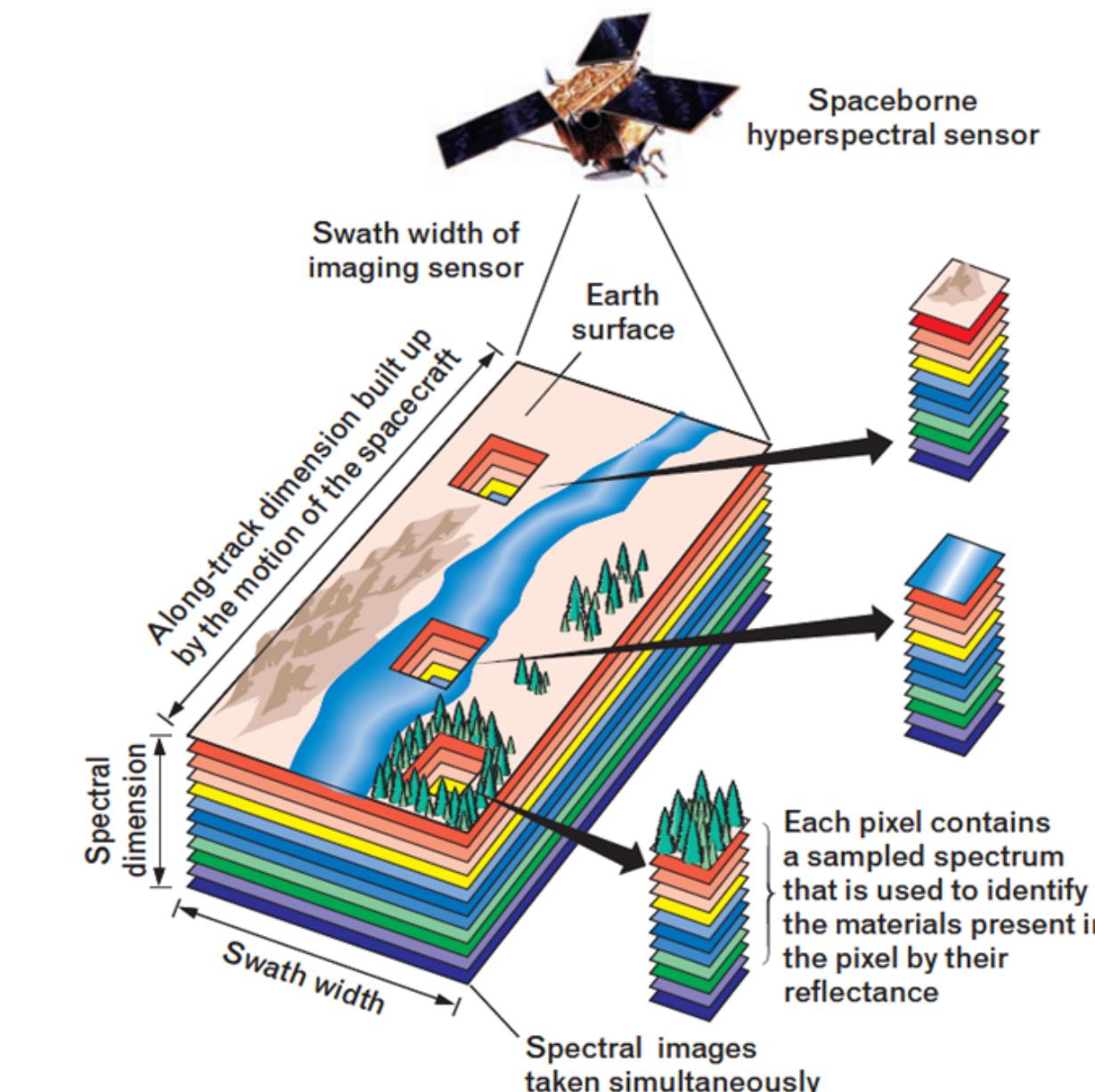
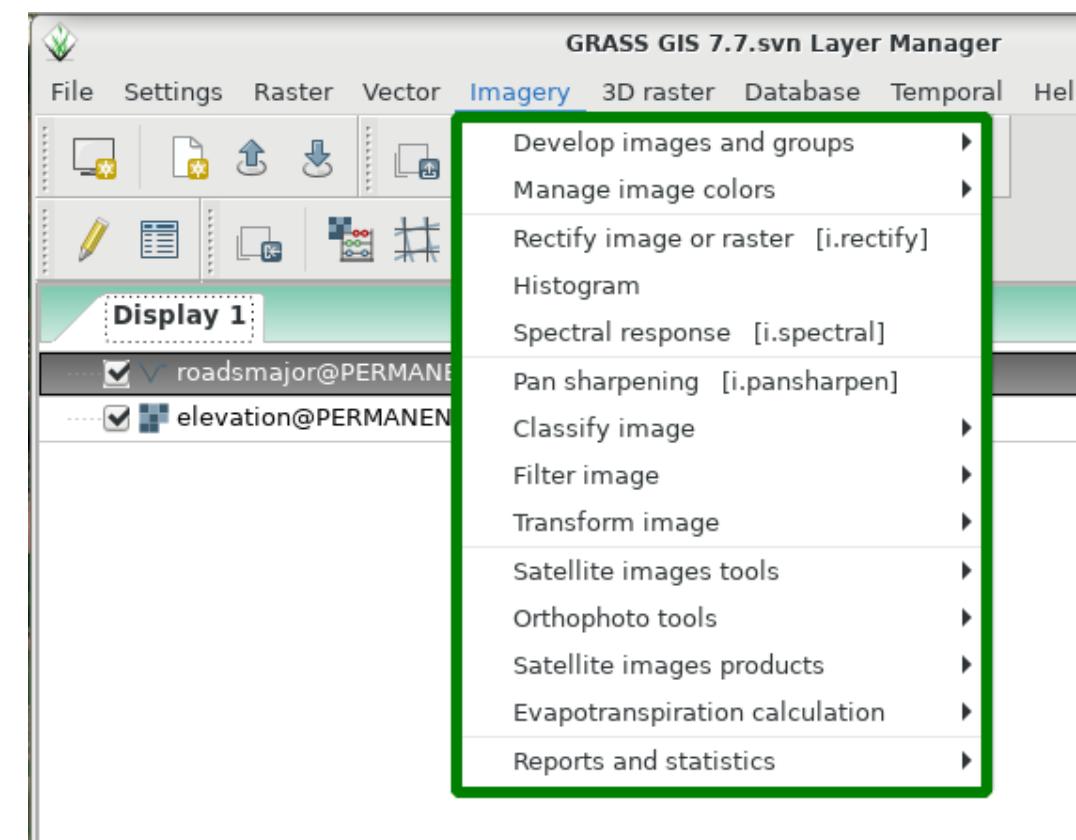


Wegman et al., 2017

# Satellite imagery processing

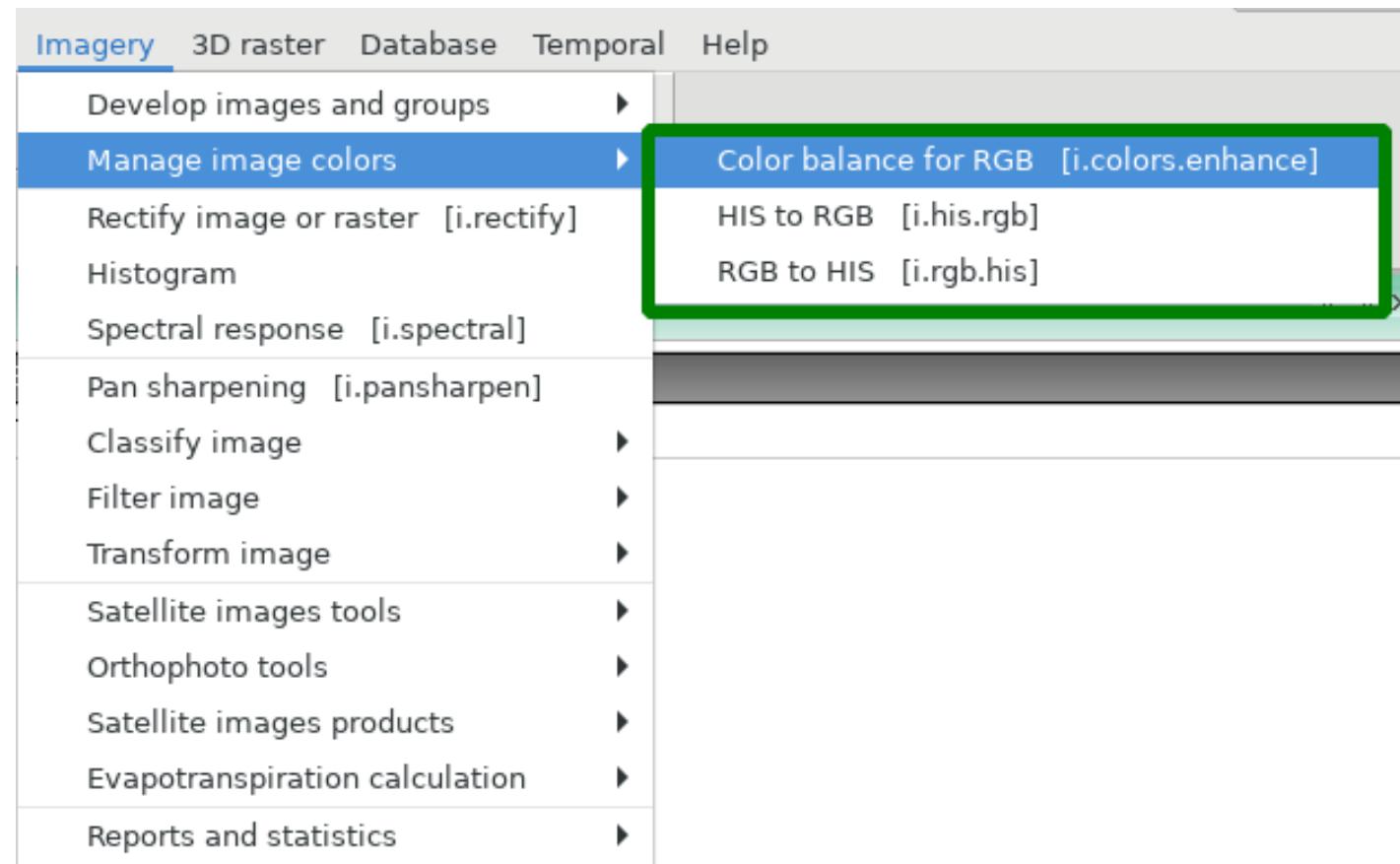


# Imagery menu

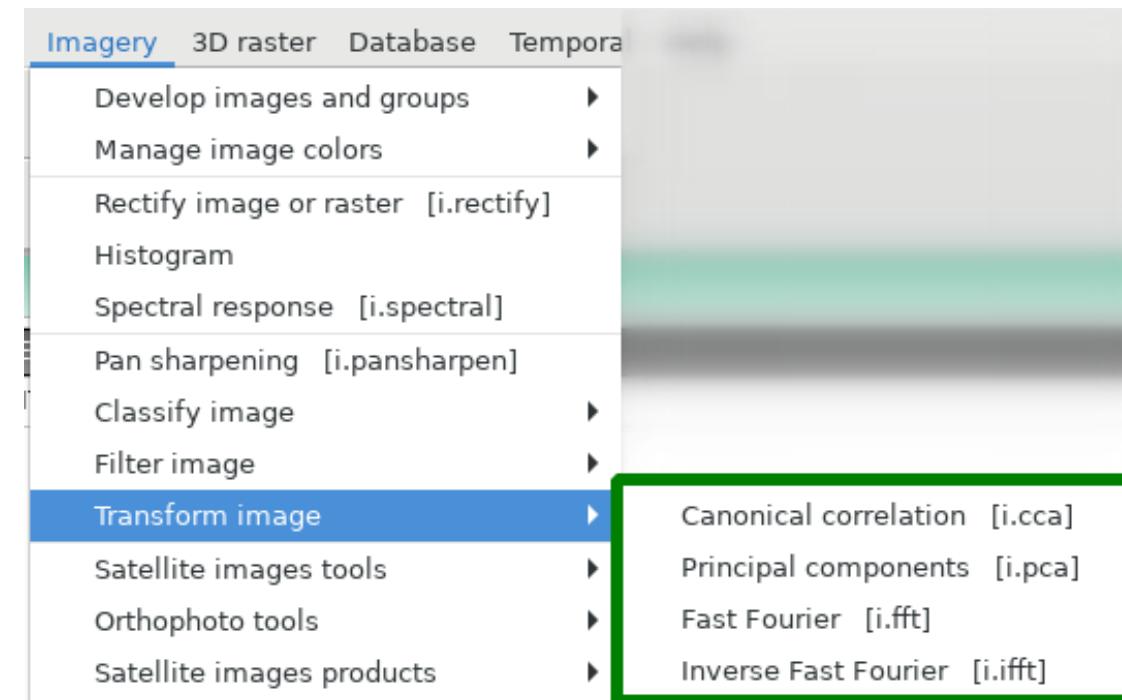


## Image processing manual

# Manage colors

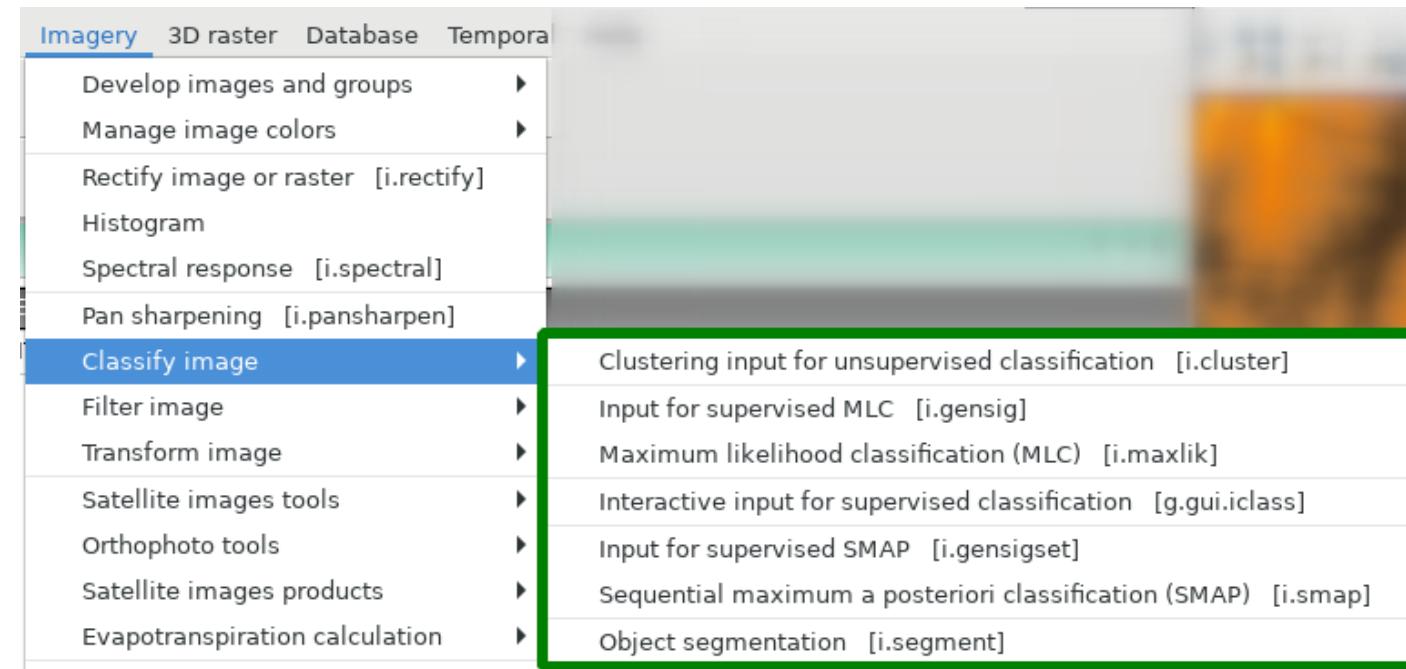


# Transformations



- **i.pca:** Principal components analysis
- **i.fft:** Fast Fourier Transform
- **i.pansharpen:** Image fusion algorithms to sharpen multispectral with high-res panchromatic channels

# Classification

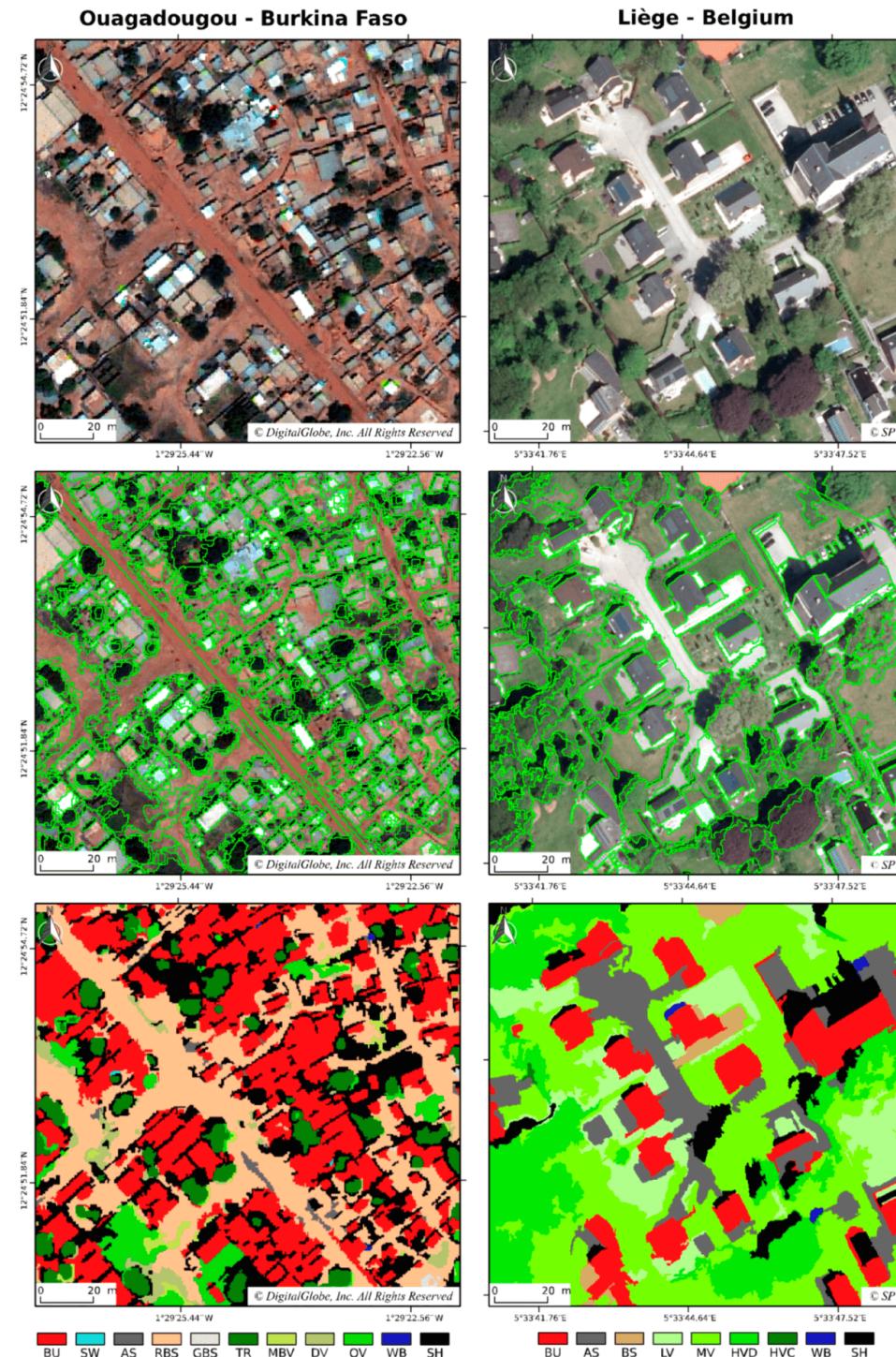


- **r.learn.ml**: Supervised classification and regression with Machine Learning
- **r.fuzzy.system**: Full fuzzy logic standalone classification system
- **i.ann.maskrcnn**: Supervised classification using convolutional neural networks
- **i.object.activelearning**: Active learning for classifying raster objects

Supervised and Unsupervised methods available. See [Image classification](#) wiki for details and mini tutorial

# Segmentation and OBIA

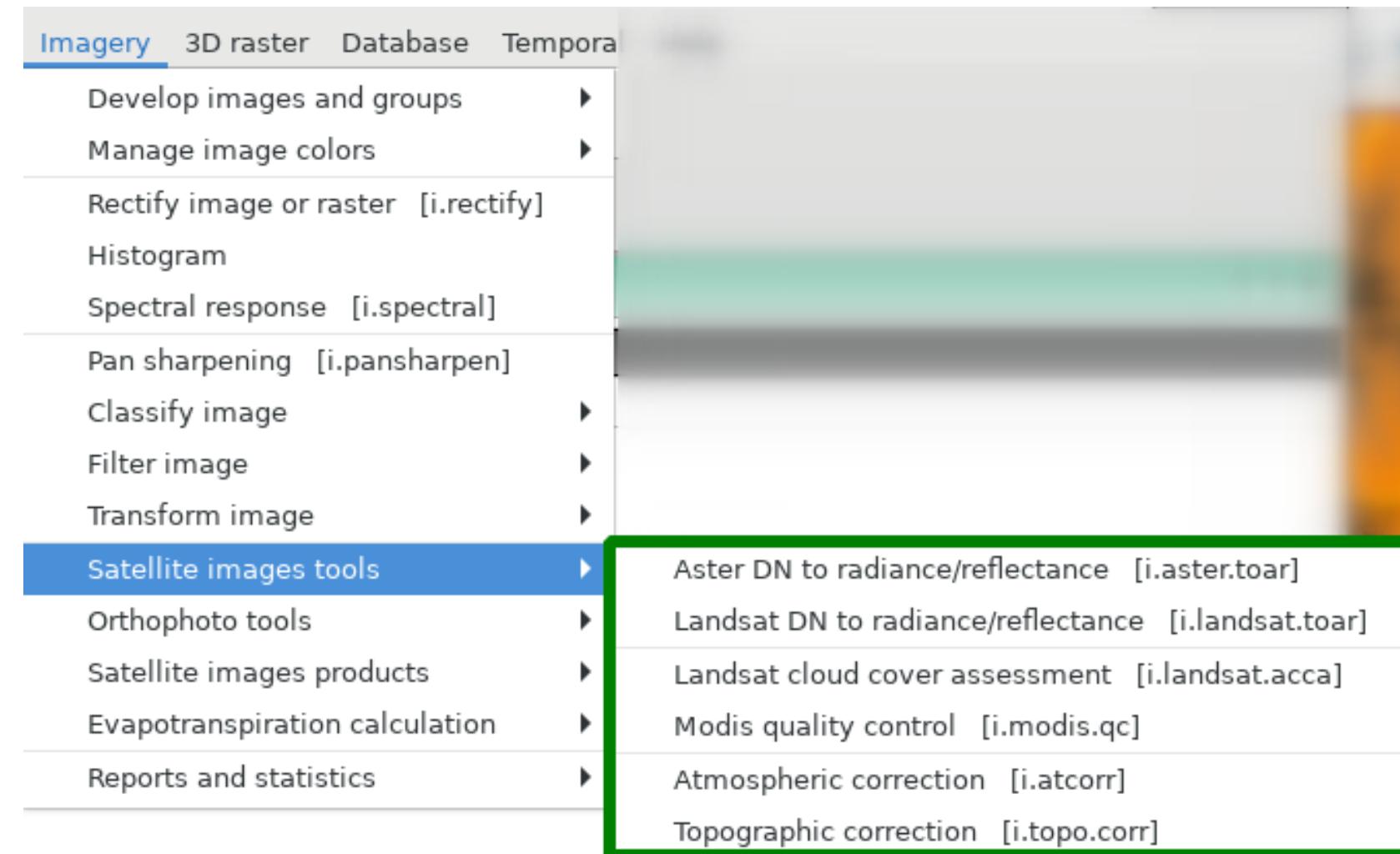
## OBIA-automated classification



- **i.segment:** Identifies segments (objects) from imagery data
- **i.segment.hierarchical:** Hierarchical segmentation
- **i.segment.stats:** Calculates statistics describing raster areas
- **i.segment.uspo:** Unsupervised segmentation parameter optimization
- **i.superpixels.slic:** Performs image segmentation using SLIC method

## OBIA processing chain

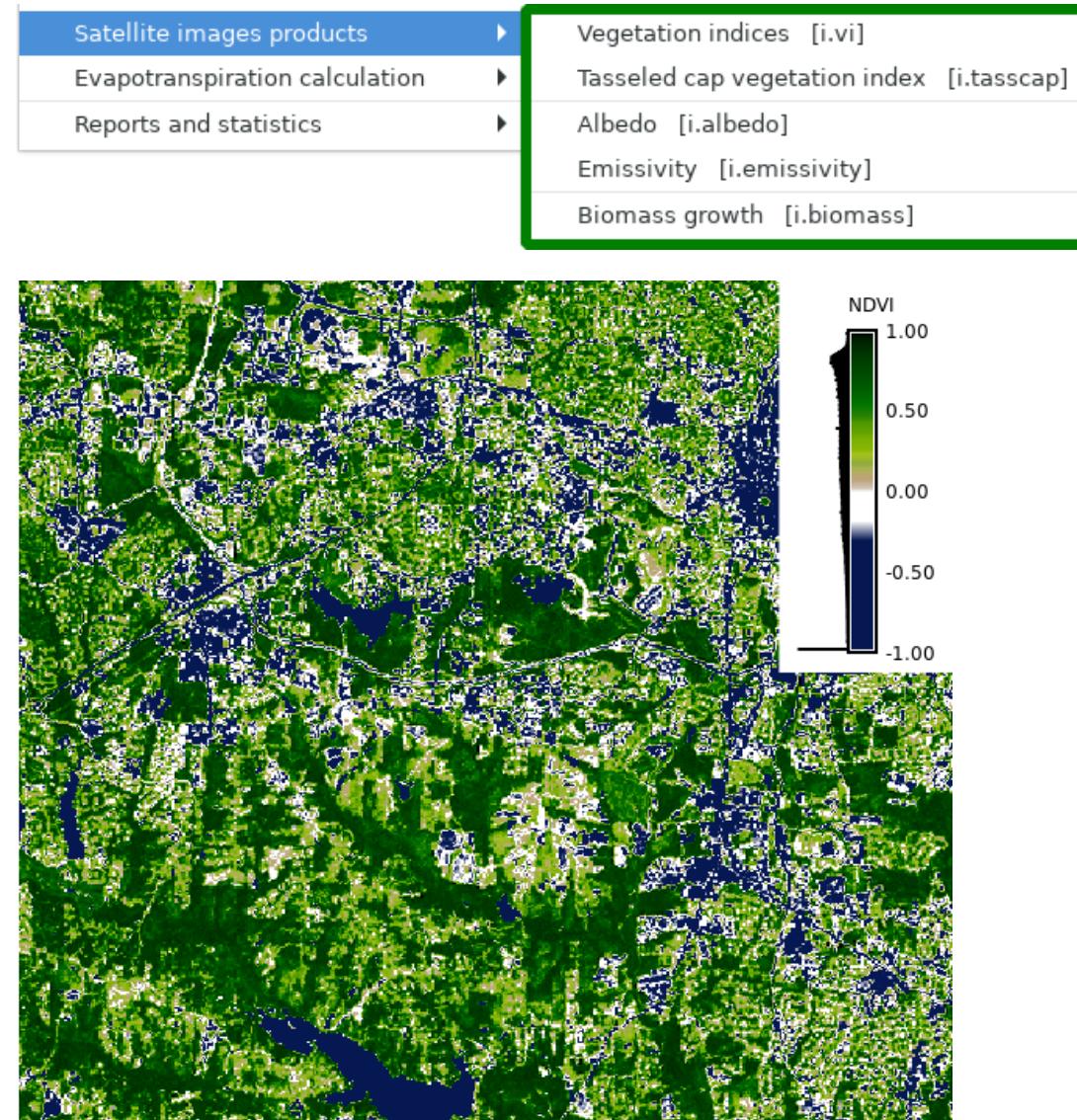
# Generic RS tools and tools for specific sensors



Sentinel-2A Band 02 after *i.atcorr*

... plus add-ons for MODIS, Sentinel2, Landsat, SRTM, GPM, etc.

# RS derived products

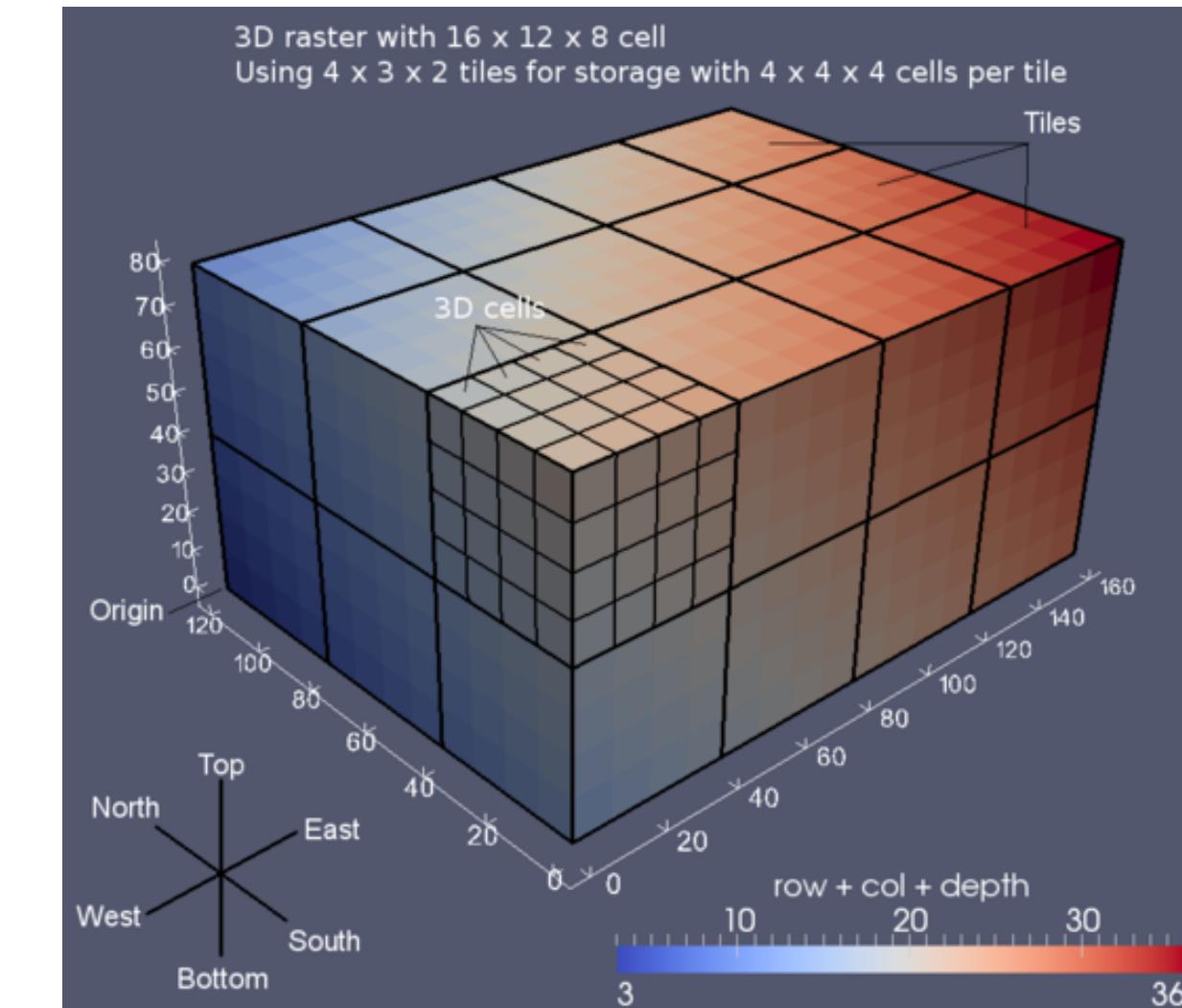
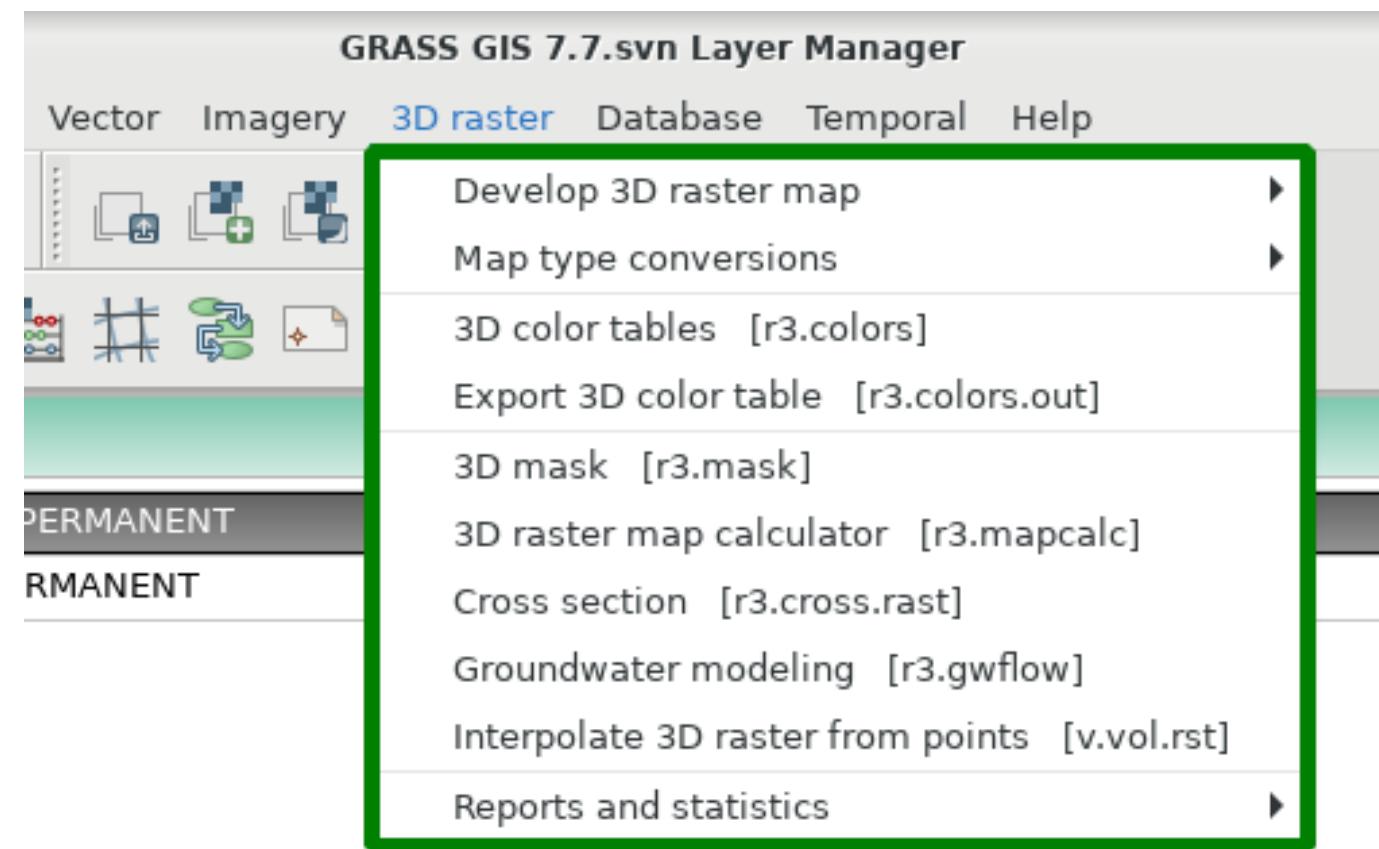


- **i.wi**: Calculates different types of water indices
- **i.lswt**: Computes Lake Surface Water Temperatures from TOA Brightness Temperatures
- **i.landsat8.swlst**: Split-window algorithm estimating LST from Landsat 8 OLI/TIRS
- **i.rh**: Water in atmosphere: relative humidity, water vapour
- **i.water**: Water detection from satellite data derived indices

# 3D raster processing



# 3D raster menu



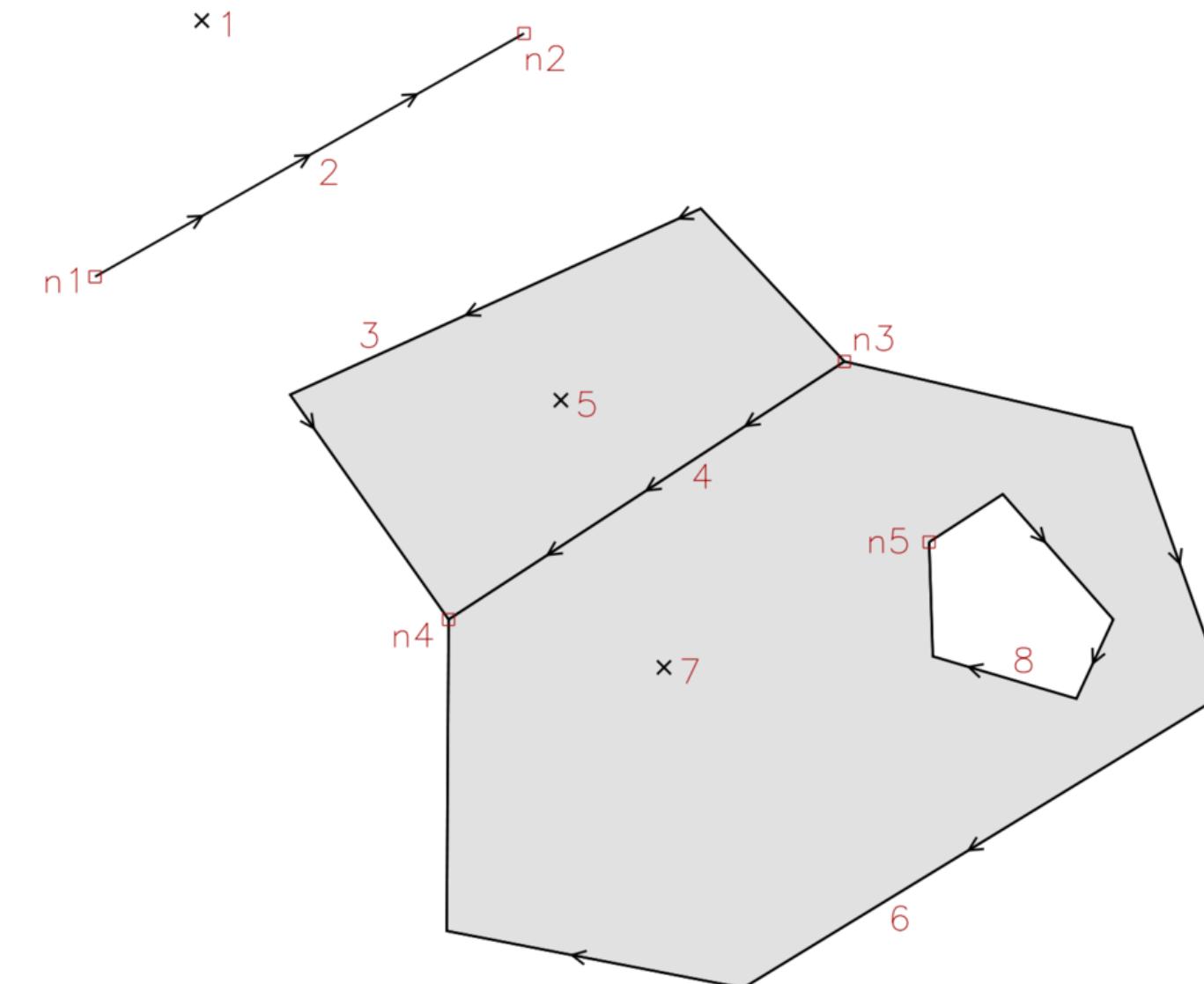
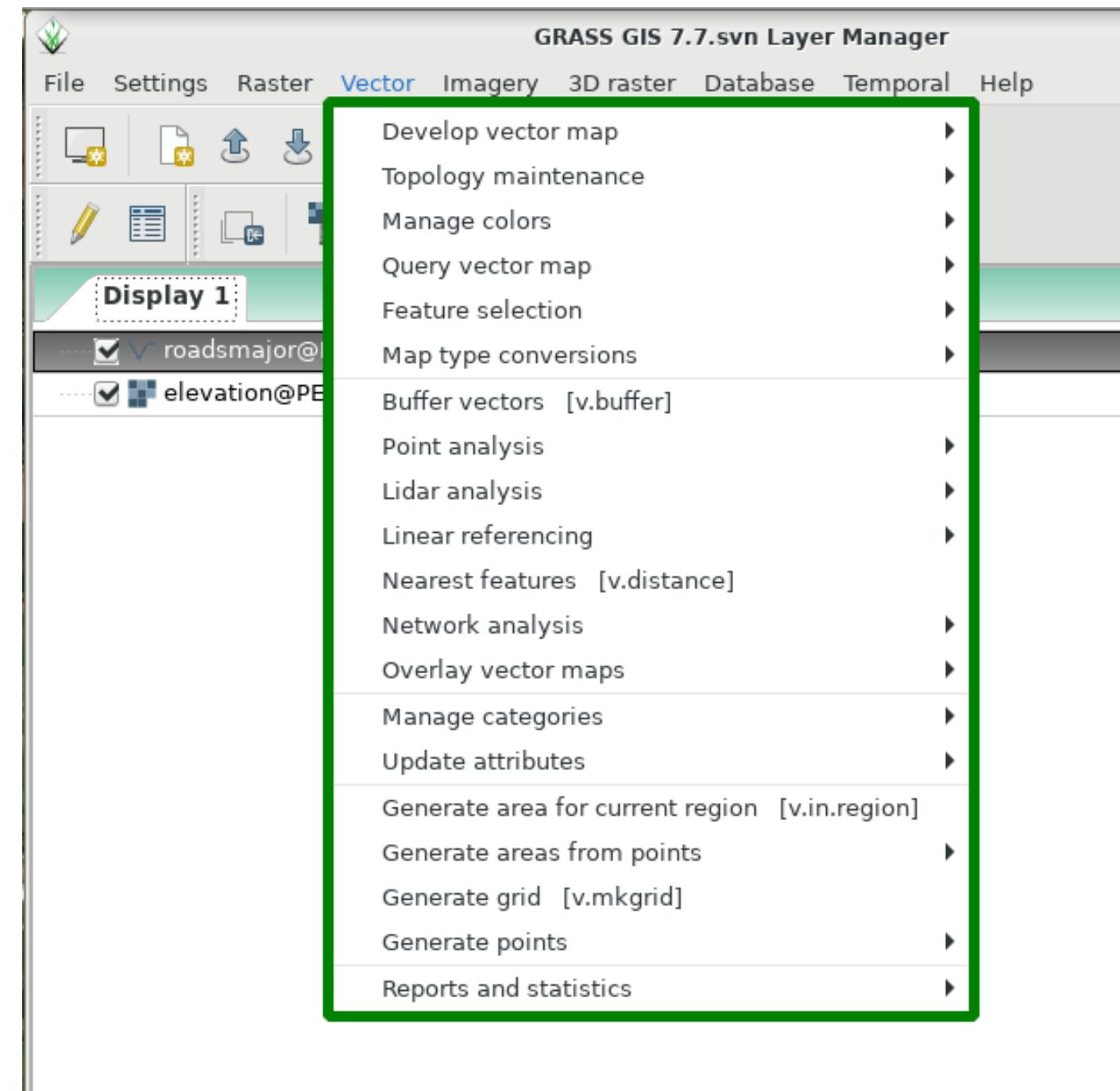
3D raster coordinate system and internal tile layout

## 3D raster processing manual

# Vector data processing



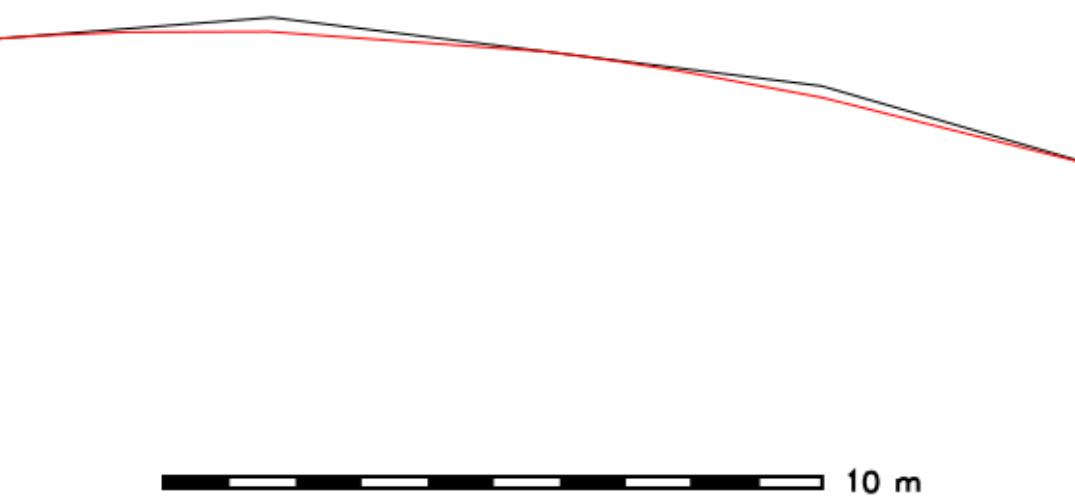
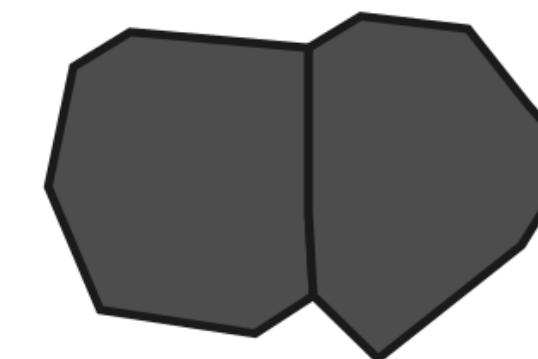
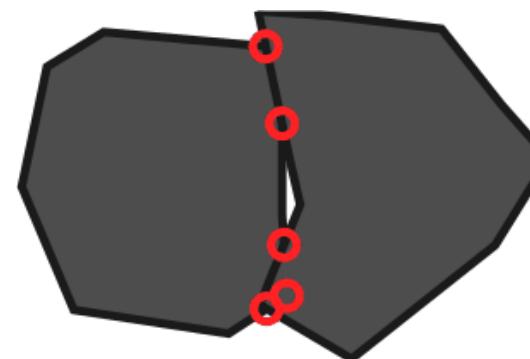
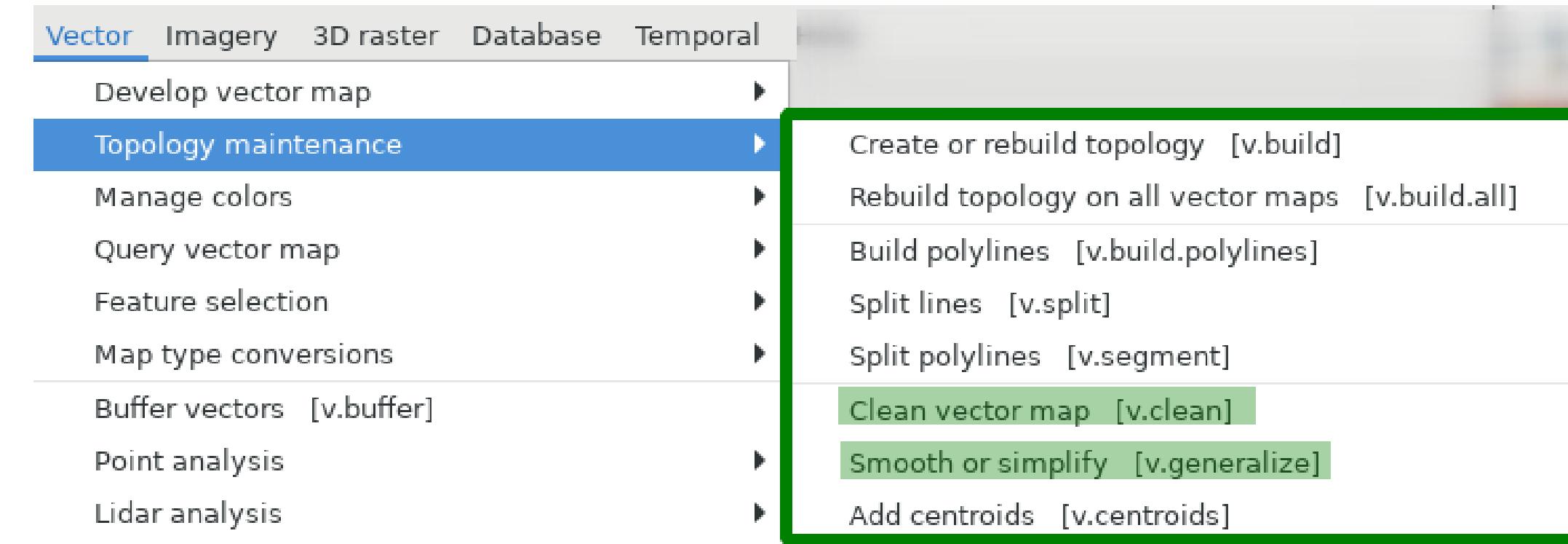
# Vector menu



Topological vector formats in GRASS GIS

## Vector processing manual

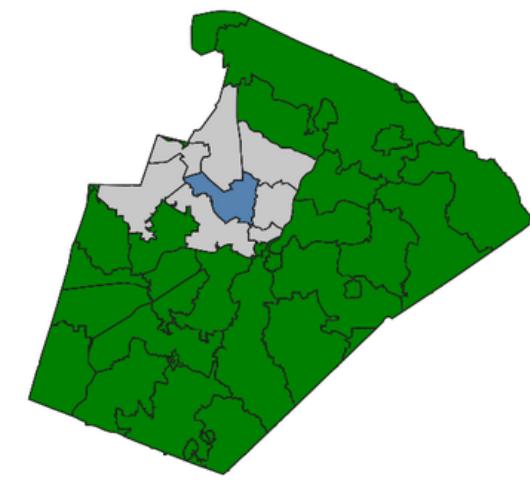
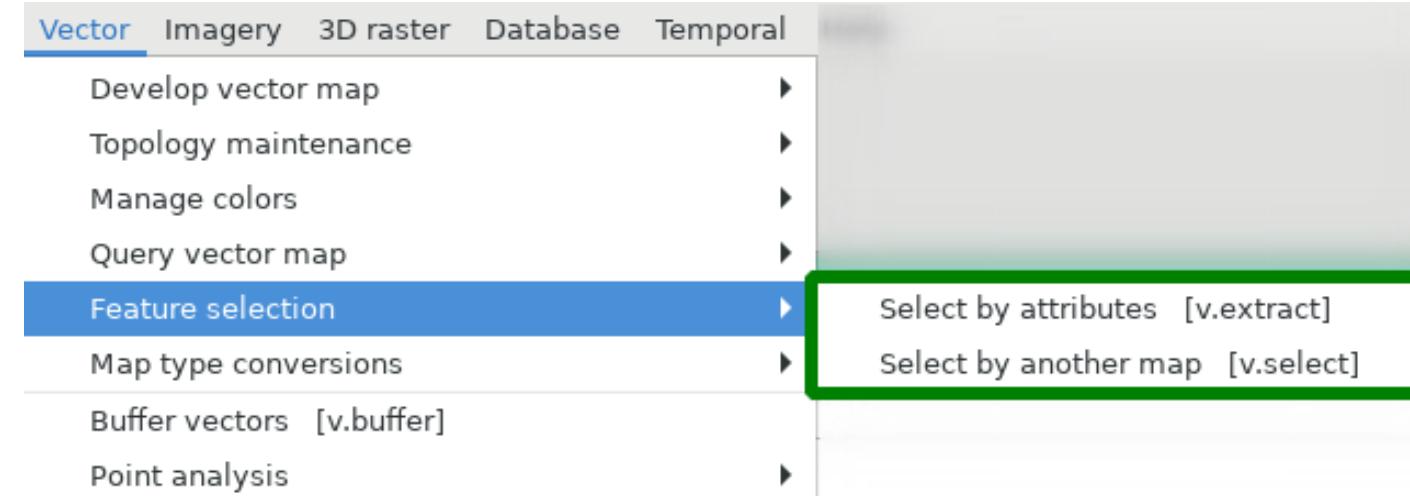
# Topology maintenance



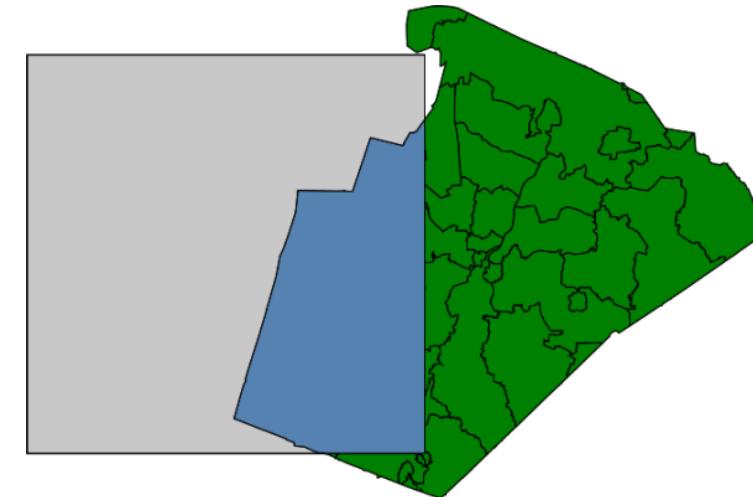
Cleaning topological errors in vector map

Smoothing. See also the [v.generalize](#) wiki

# Selection and overlaying



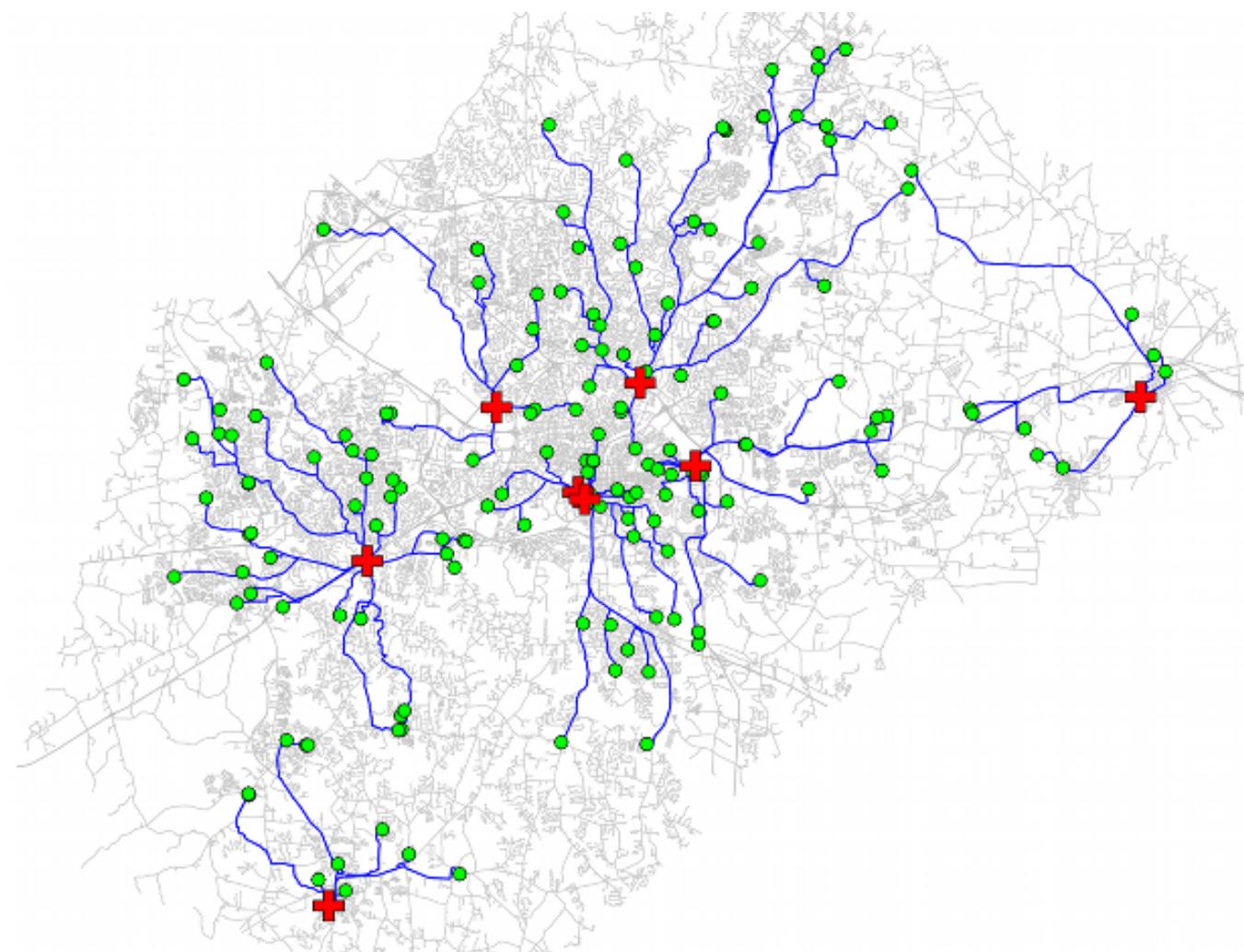
v.select operator *TOUCHES*



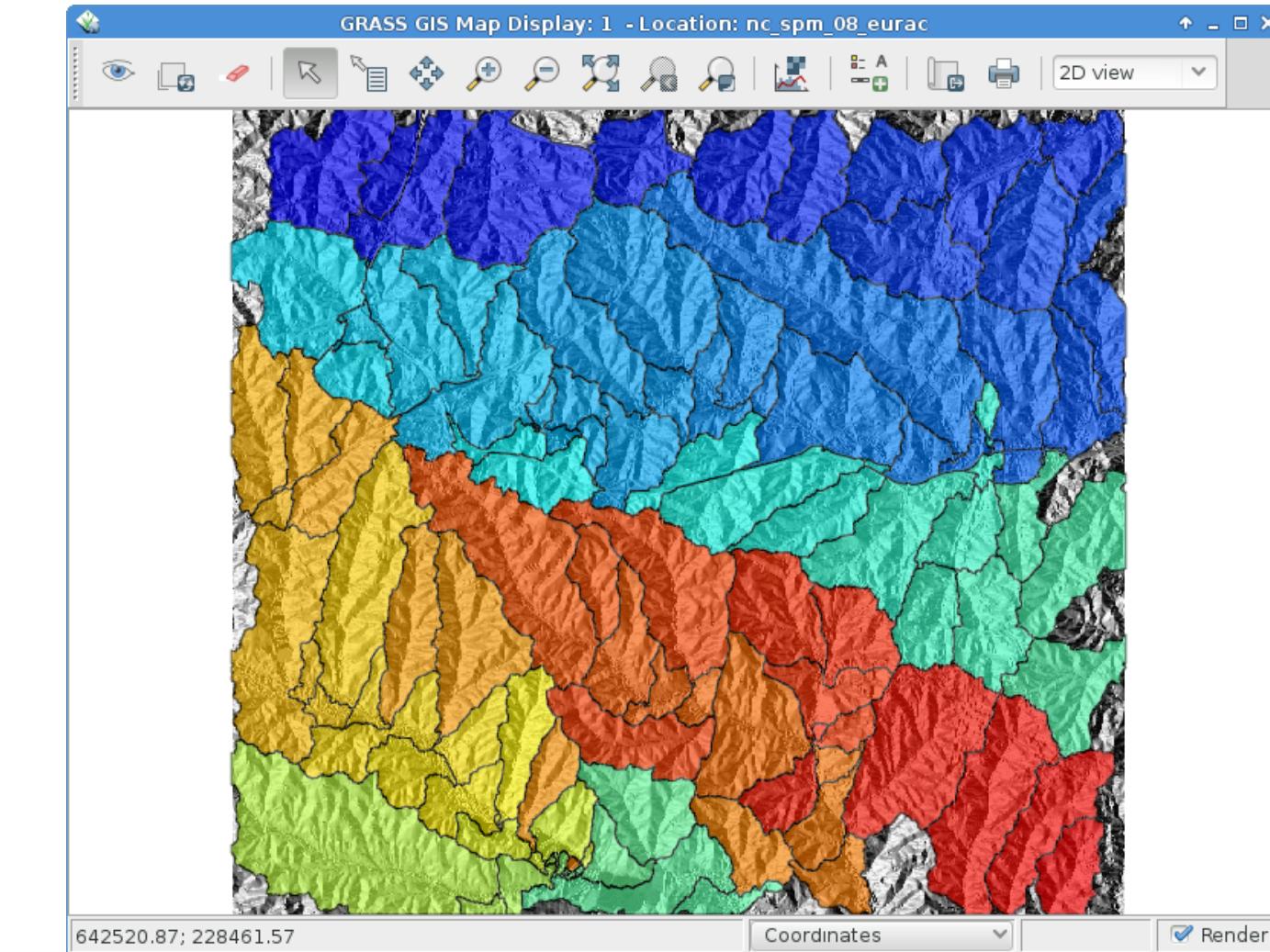
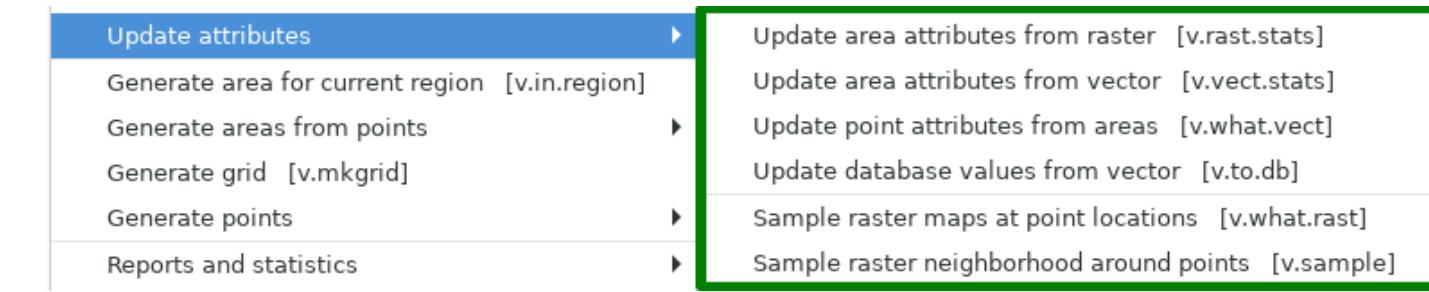
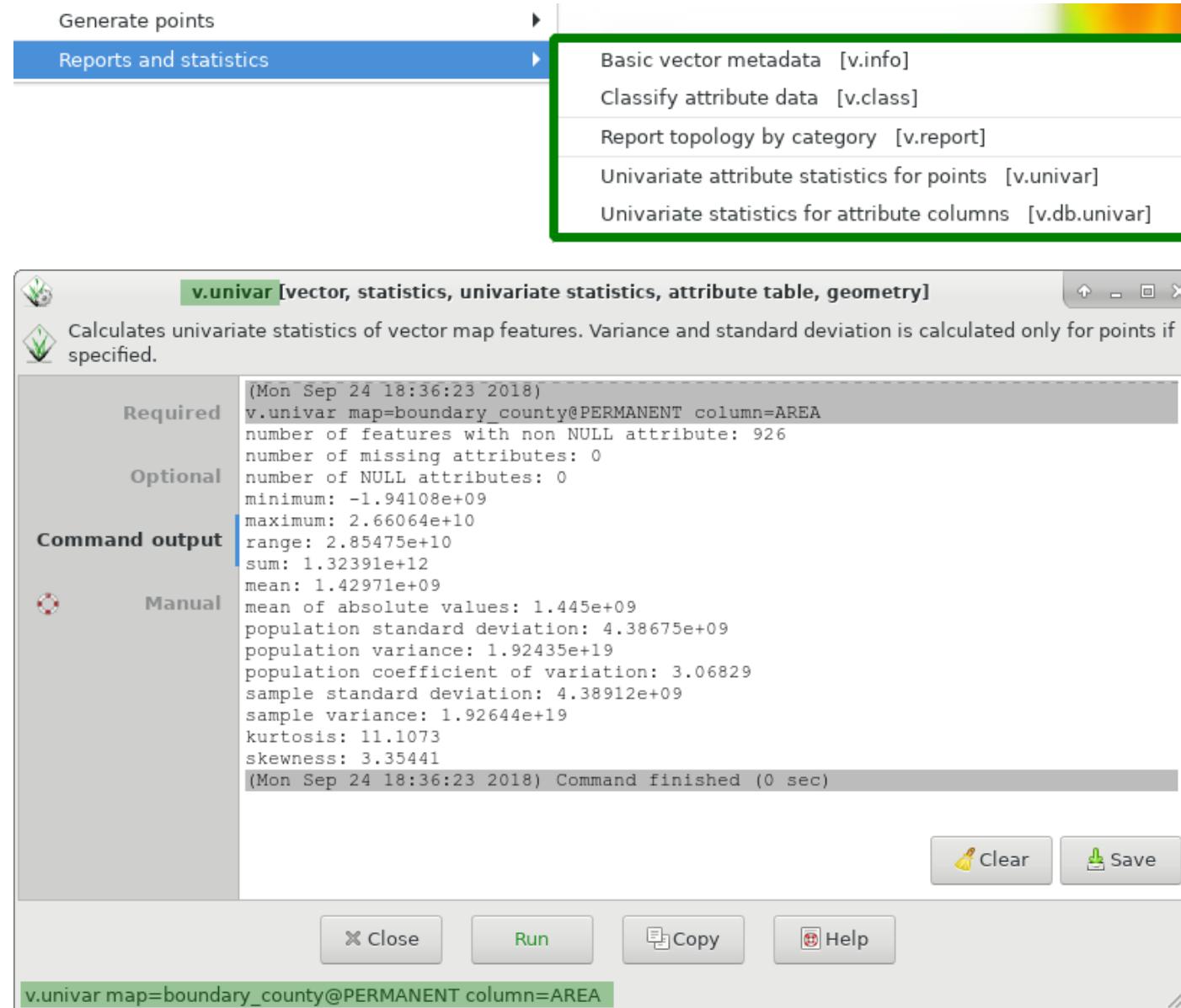
v.overlay operator *NOT*

# Network analysis

Network analysis	▶	Vector network analysis tool
Overlay vector maps	▶	Network preparation [v.net]
Manage categories	▶	Allocate subnets [v.net.alloc]
Update attributes	▶	Split net [v.net.iso]
Generate area for current region [v.in.region]	▶	Shortest path [v.net.path]
Generate areas from points	▶	Shortest path for sets of features [v.net.distance]
Generate grid [v.mkgrid]	▶	Shortest path using timetables [v.net.timetable]
Generate points	▶	Shortest path for all pairs [v.net.allpairs]
Reports and statistics	▶	Visibility network [v.net.visibility]
		Bridges and articulation points [v.net.bridge]
		Maximum flow [v.net.flow]
		Vertex connectivity [v.net.connectivity]
		Components [v.net.components]
		Centrality [v.net.centrality]
		Steiner tree [v.net.steiner]
		Minimum spanning tree [v.net.spanningtree]
		Traveling salesman analysis [v.net.salesman]



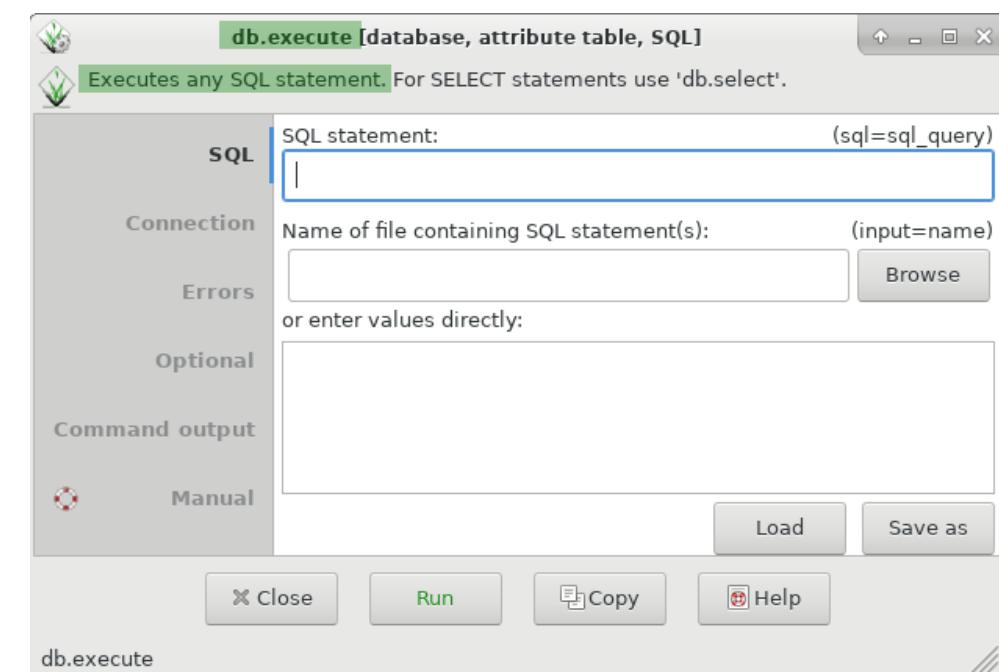
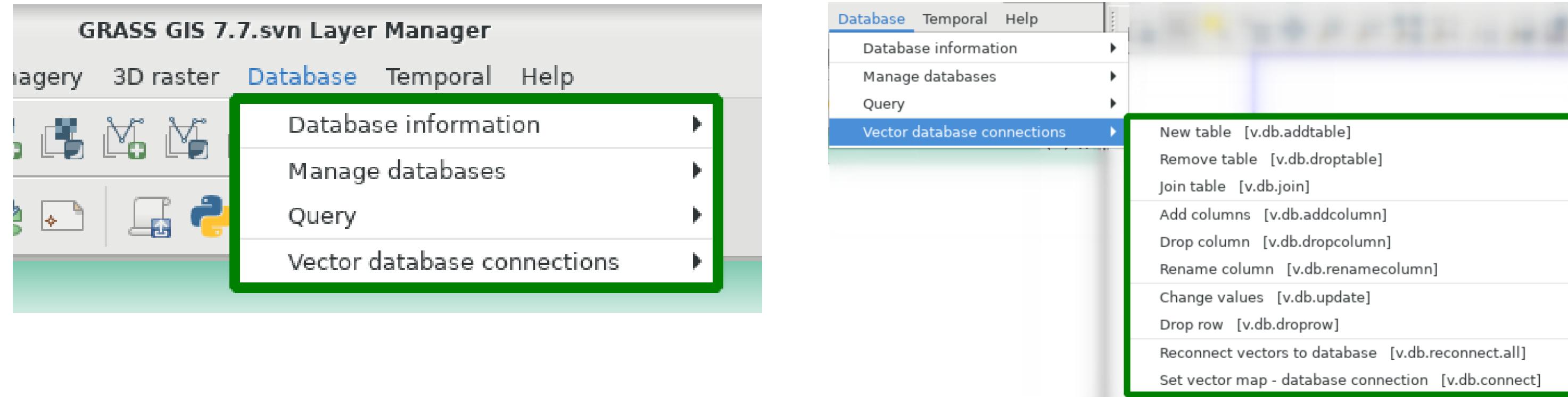
# Reporting, stats and update of attributes



# Database management



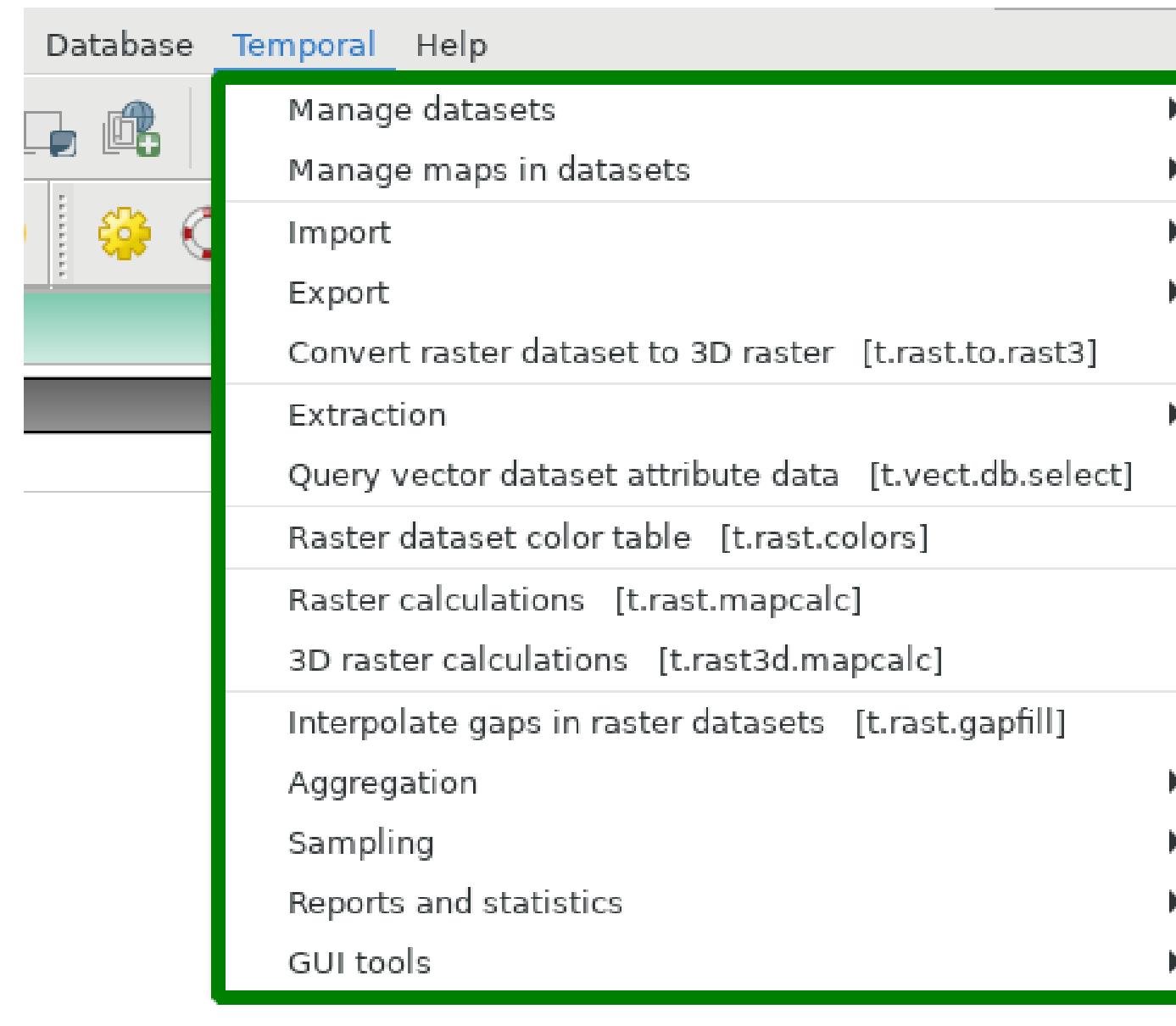
# Database management menu



# Temporal data processing



# Temporal menu



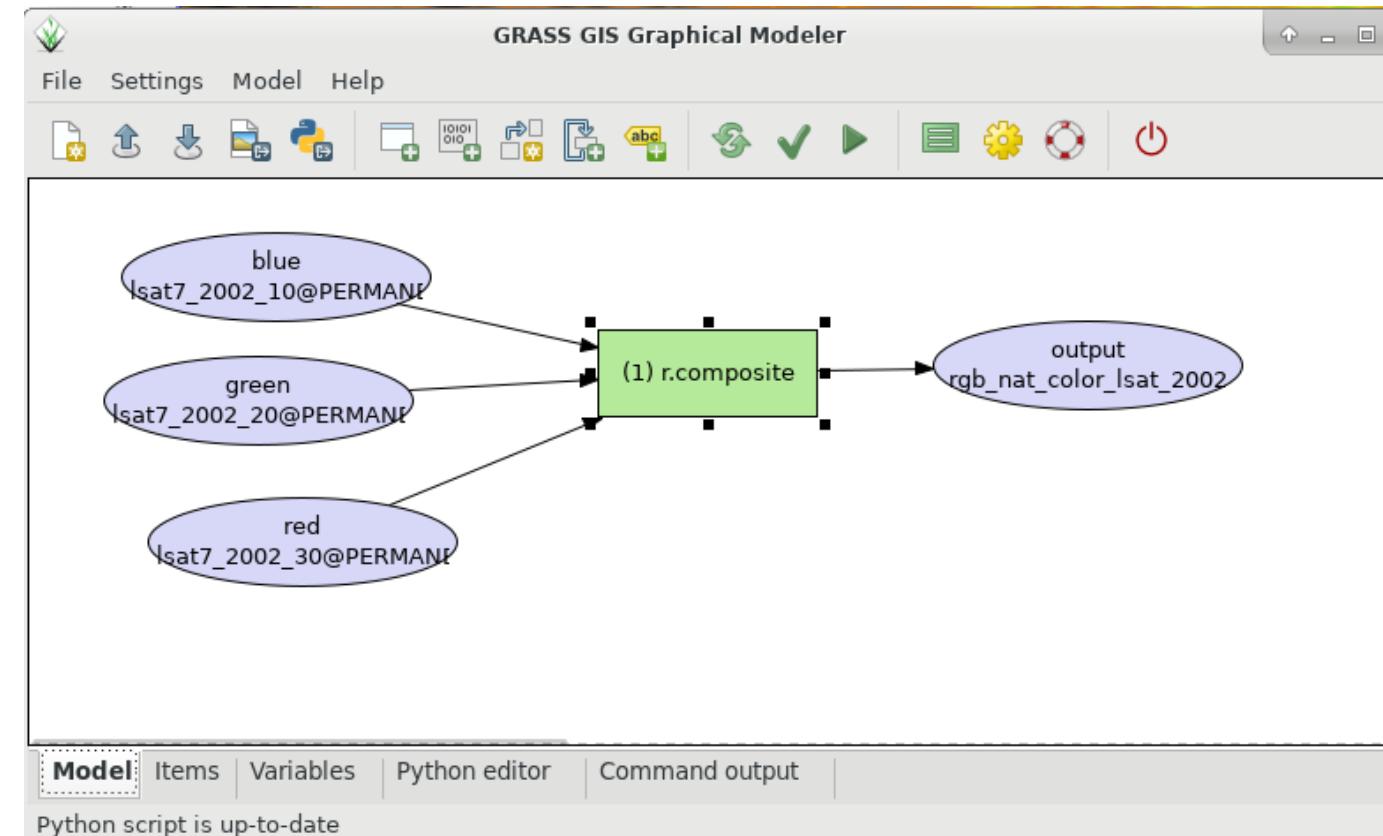
- import/export
- temporal topology
- aggregation
- accumulation
- temporal algebra
- temporal interpolation

We'll see this in more detail on Thursday ☺



# Graphical modeler

# Graphic model and Python translation



The screenshot shows the Python script editor within the GRASS GIS Graphical Modeler. The title bar reads 'GRASS GIS Graphical Modeler'. The main window displays a Python script generated from the graphical model. The script starts with a module header and imports sys, os, and atexit. It defines a cleanup function that does nothing, and a main function that runs the 'r.composite' command with red, green, and blue inputs and an output layer named 'rgb\_nat\_color\_lsat\_2002'. The script concludes with a return statement and an if \_\_name\_\_ == "\_\_main\_\_": block that calls the main function. At the bottom, there are buttons for Refresh, Save As, and Run. The Python editor tab is highlighted. A status message at the bottom says 'Python script is up-to-date'.

```
%module
#% description: Script generated by wxGUI Graphical Modeler.
#%end

import sys
import os
import atexit

from grass.script import parser, run_command

def cleanup():
    pass

def main():
    run_command("rcomposite",
               red = "lsat7_2002_30@PERMANENT",
               green = "lsat7_2002_20@PERMANENT",
               blue = "lsat7_2002_10@PERMANENT",
               levels = 32,
               output = "rgb_nat_color_lsat_2002")

    return 0

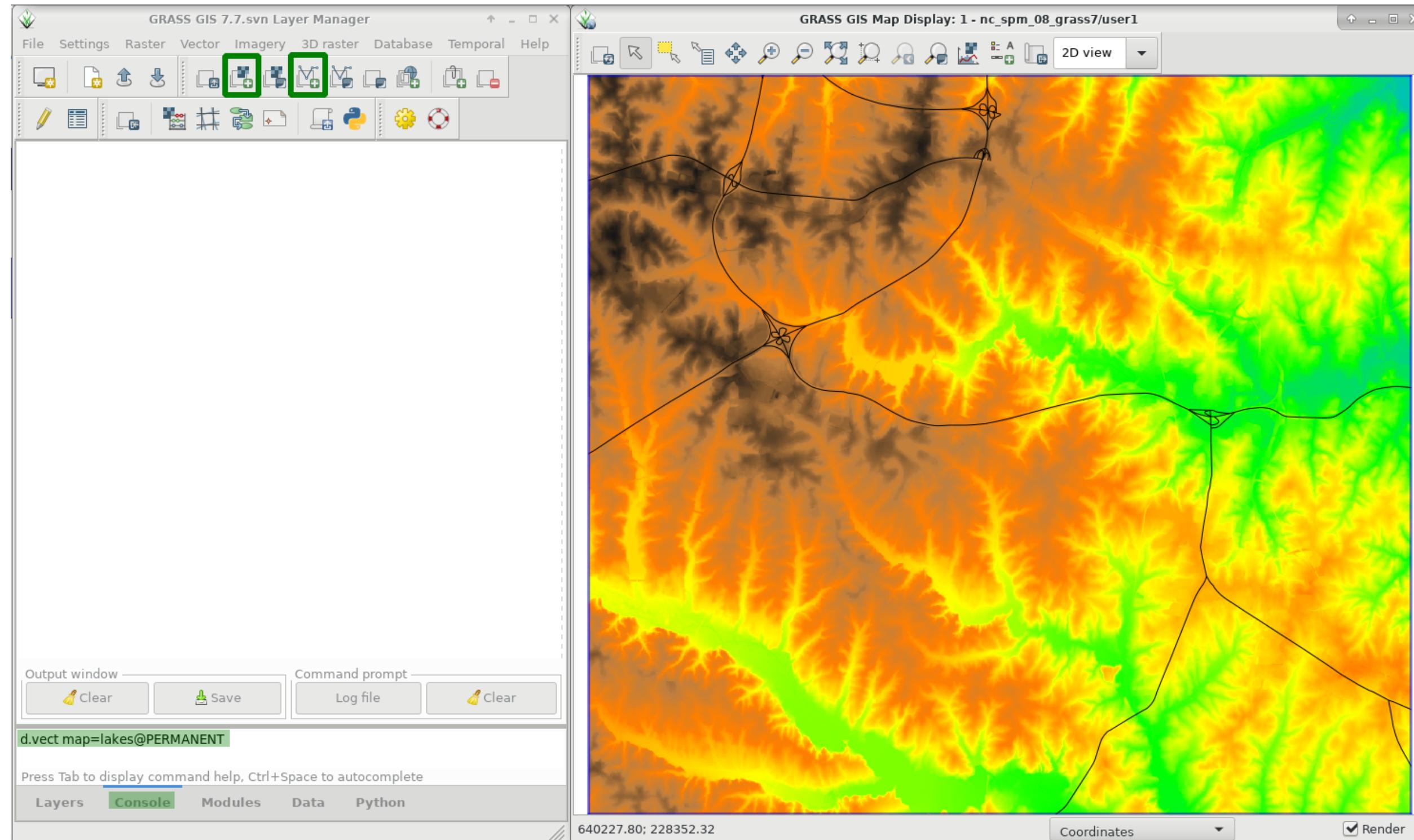
if __name__ == "__main__":
    options, flags = parser()
    atexit.register(cleanup)
    sys.exit(main())
```

See `g.gui.gmodeler` manual page for further details.

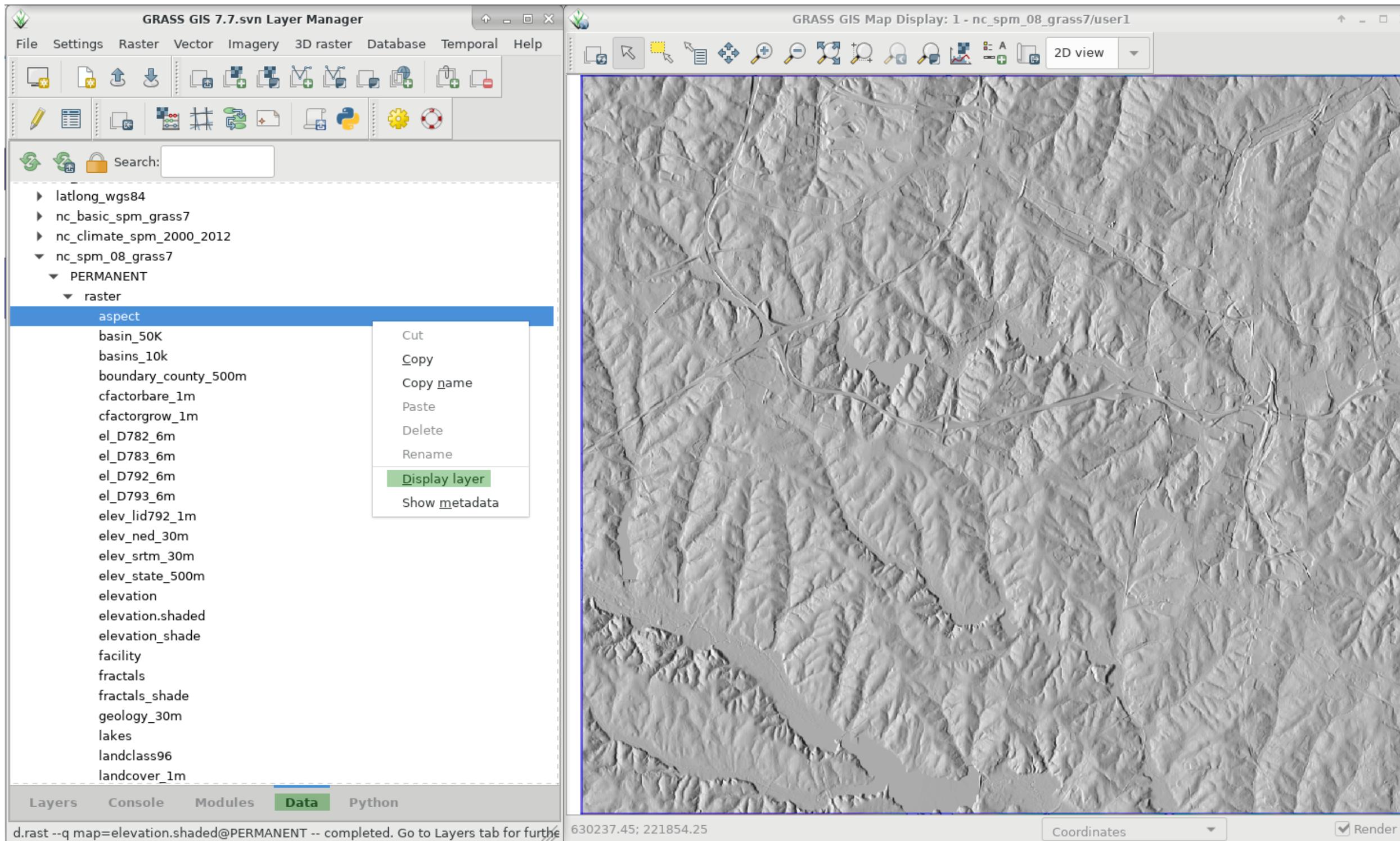


# Visualization tools

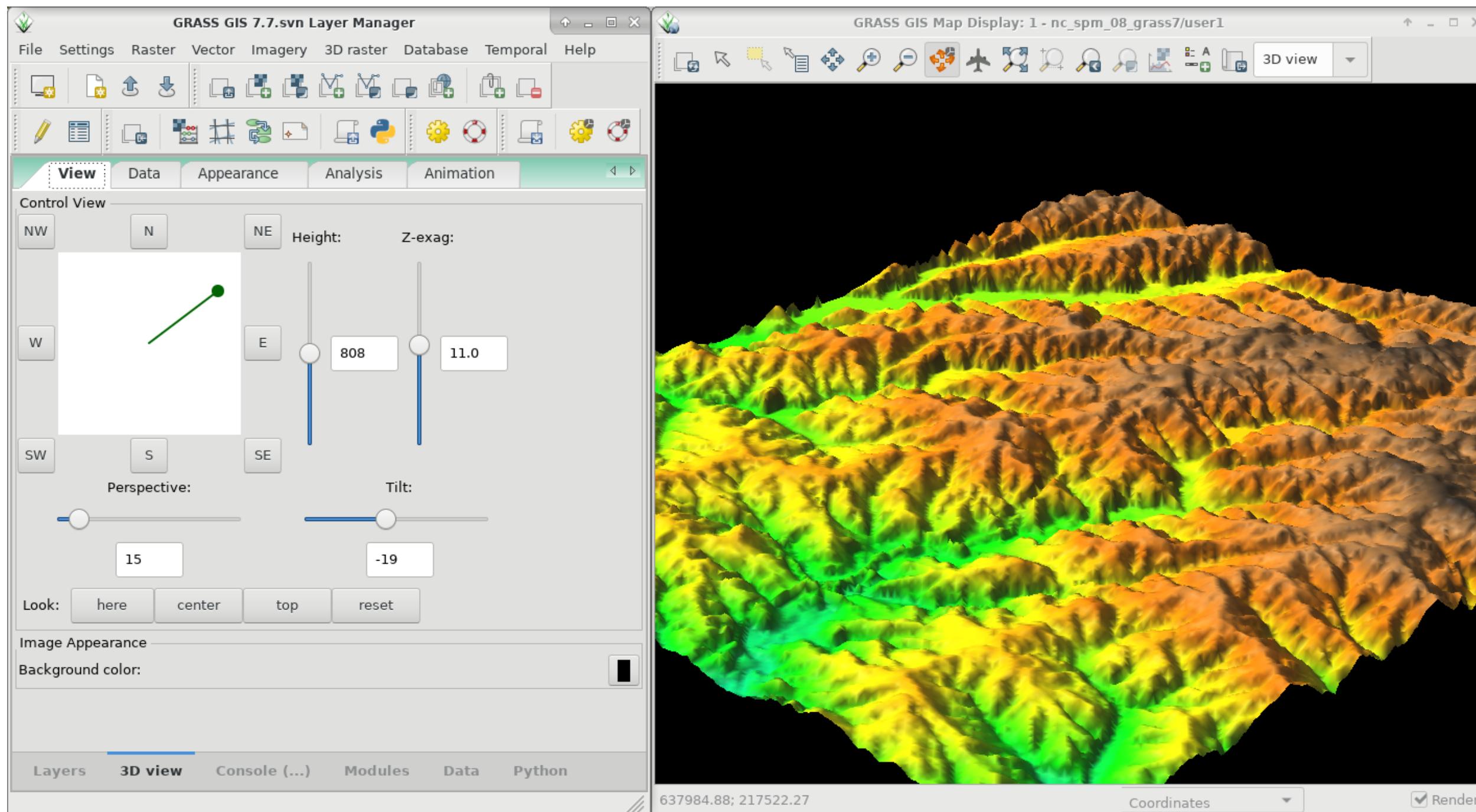
# Map display: console tab



# Map display: data tab

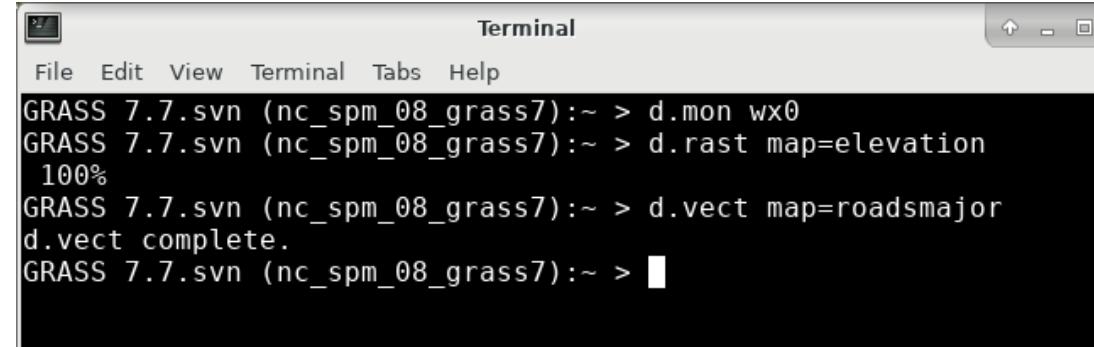


# Map display: 3D view

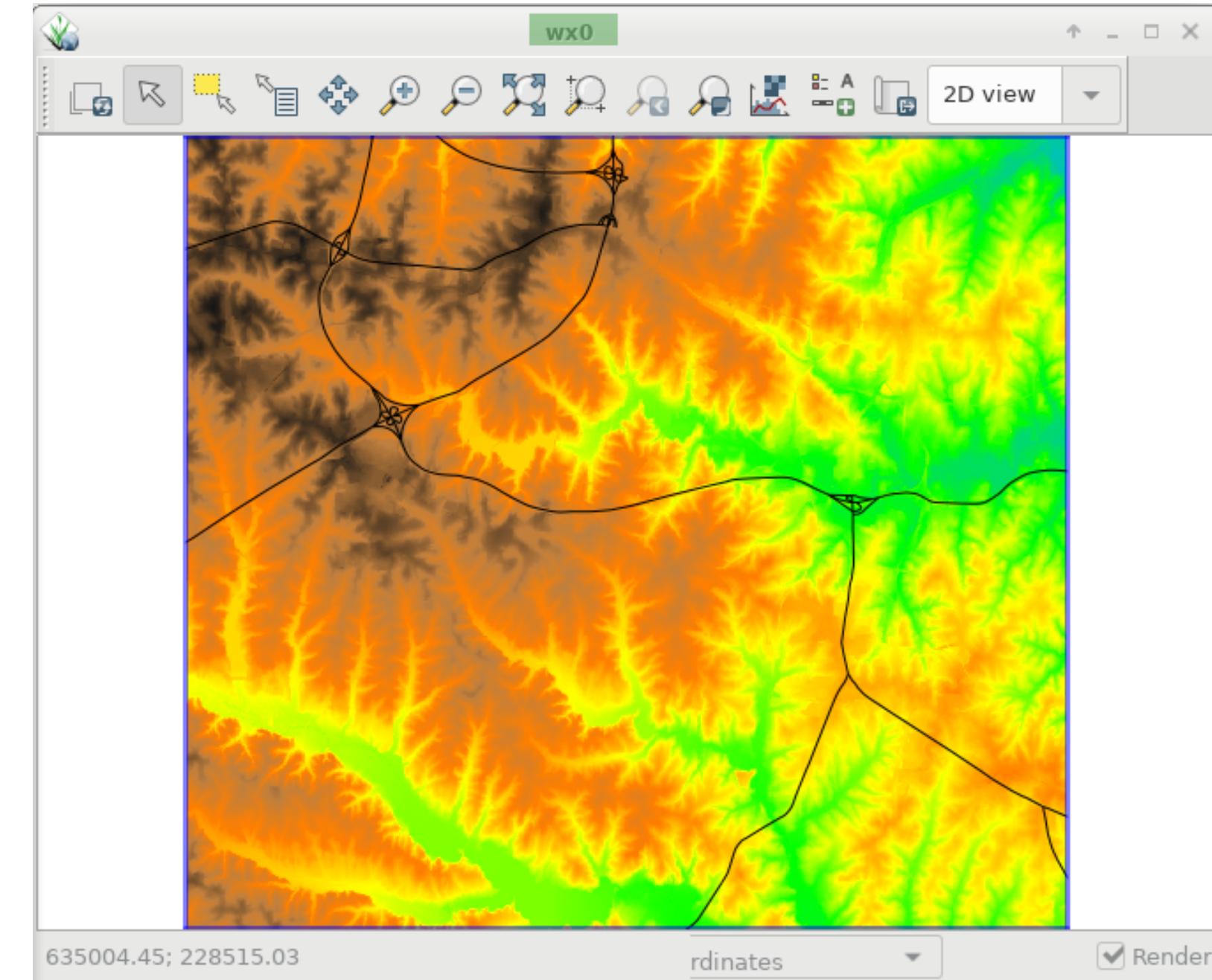


# Wx-monitors

In the terminal:

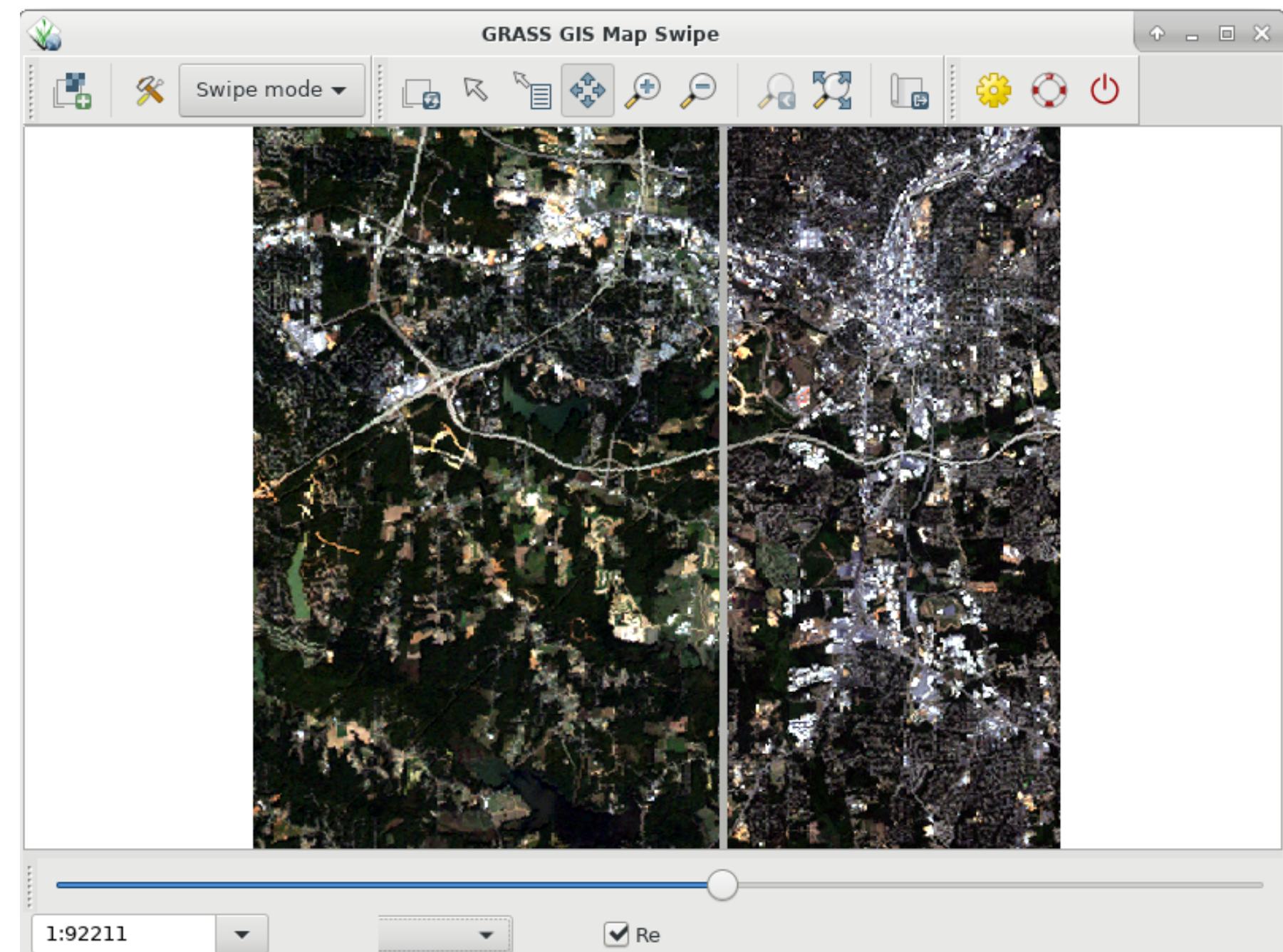


```
Terminal
File Edit View Terminal Tabs Help
GRASS 7.7.svn (nc_spm_08_grass7):~ > d.mon wx0
GRASS 7.7.svn (nc_spm_08_grass7):~ > d.rast map=elevation
100%
GRASS 7.7.svn (nc_spm_08_grass7):~ > d.vect map=roadsmajor
d.vect complete.
GRASS 7.7.svn (nc_spm_08_grass7):~ > █
```



The wx-monitors have the same **buttons** than the main Map Display in the GUI

# Map-swipe



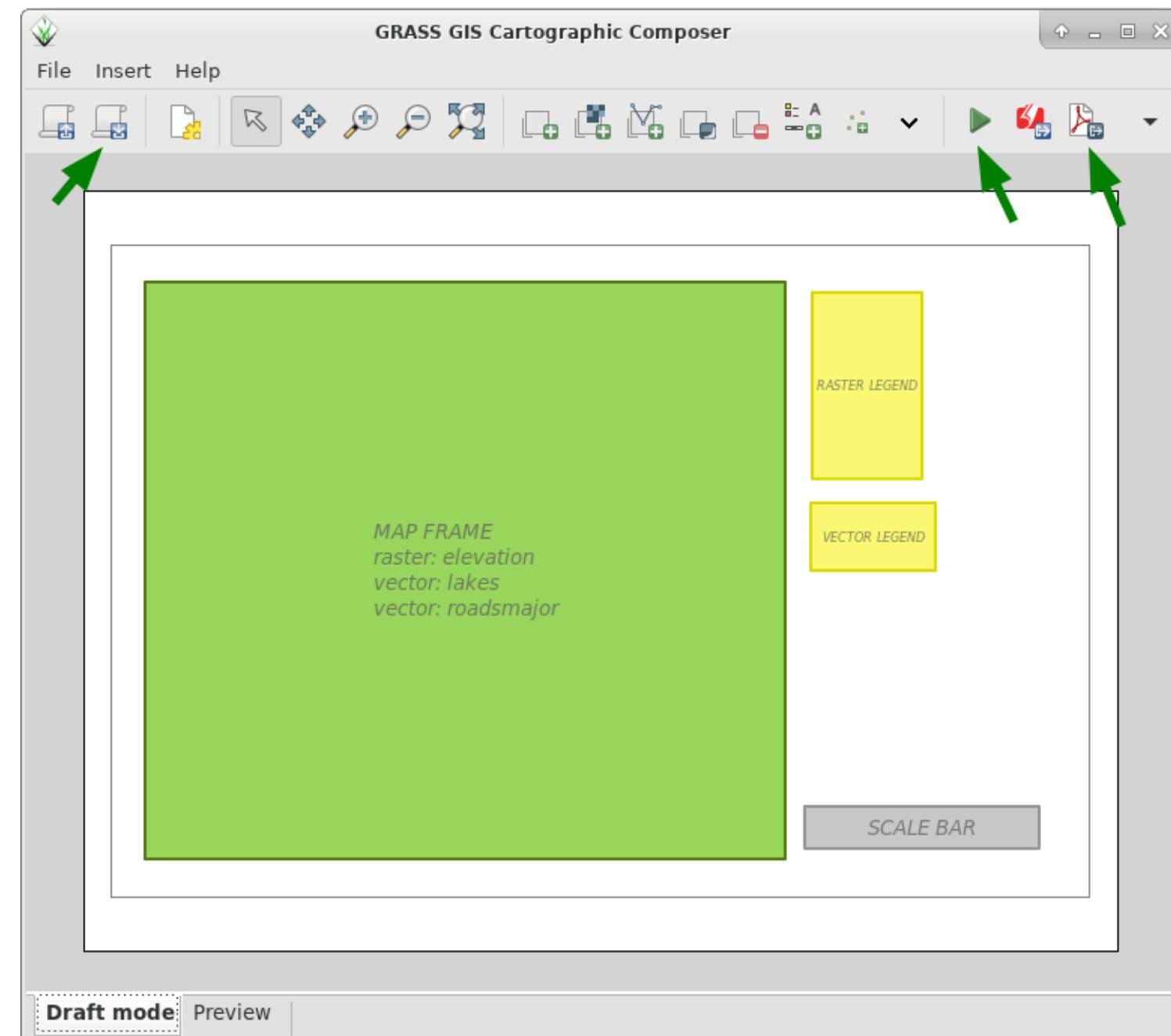
g.gui.mapswipe

# Animation tool



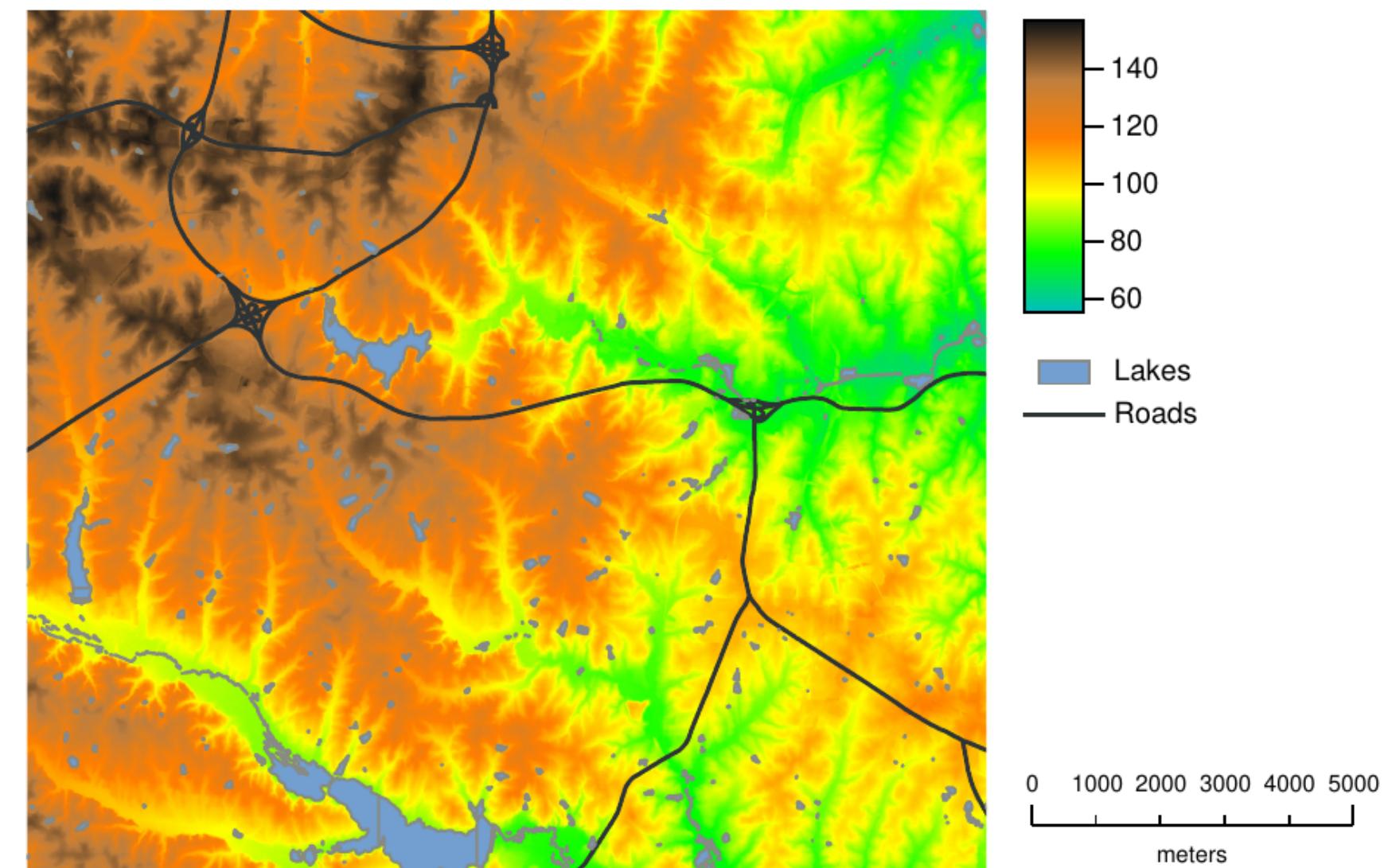
g.gui.animation

# Cartographic composer



g.gui.psmap

# Export as .ps .eps or .pdf



```
# timestamp: 2018-09-19 18:10
# location: nc_spm_08_grass7
# mapset: user1
# page orientation: Portrait
# g.region raster=elevation@PERMANENT nsres=10.75697211 ewres=10.752688

maploc 0.450 0.666 4.624 4.174
border n

paper
    width 7.48031496063
    height 5.51181102362
    left 0.196850393701
    right 0.196850393701
    bottom 0.393700787402
    top 0.393700787402
end
```

```
paper
    width 7.48031496063
    height 5.51181102362
    left 0.196850393701
    right 0.196850393701
    bottom 0.393700787402
    top 0.393700787402
end
```

```
raster elevation@PERMANENT
```

```
vareas lakes@PERMANENT
    layer 1
    masked n
    color 136:138:133
    width 1.0
    fcolor 114:159:207
    label Lakes
    lpos 1
```

raster map

```
top 0.393700787402
end

raster elevation@PERMANENT

vareas lakes@PERMANENT
    layer 1
    masked n
    color 136:138:133

    width 1.0
    fcolor 114:159:207
    label Lakes
    lpos 1
    end

vlines roadsmajor@PERMANENT
    type line
    layer 1
    masked n
```

vector of areas

```
width 1.0
fcolor 114:159:207
label Lakes
lpos 1
end

vlines roadsmajor@PERMANENT
  type line
  layer 1
  masked n
  color 46:52:54
  width 1.5
  style solid
  linecap butt
  label Roads
  lpos 2
  end

colortable y
raster elevation@PERMANENT
where 5.245 0.735
```

vector of lines

```
width 1.5
style solid
linecap butt
label Roads
lpos 2
end
colortable y
raster elevation@PERMANENT
where 5.245 0.735
discrete n
tickbar n
font Helvetica
fontsize 10
color black
end
vlegend
where 5.234 2.249
font Helvetica
fontsize 10
```

color table, i.e., raster legend

```
discrete n
tickbar n
font Helvetica
fontsize 10
color black
end

vlegend
where 5.234 2.249
font Helvetica
fontsize 10
width 0.4
cols 1
border none
end

scalebar s
where 6.040 4.591
length 5000.0
units auto
```

vector legend

```
vlegend  
  where 5.234 2.249  
  font Helvetica  
  fontsize 10  
  width 0.4  
  cols 1  
  border none  
  end  
scalebar s  
  where 6.040 4.591  
  length 5000.0  
  units auto  
  height 0.1  
  segment 5  
  numbers 1  
  fontsize 8  
  background y  
  end
```

scale bar

# Add-ons



# Some (other) cool add-ons

- **i.modis**: Toolset for download and processing of MODIS products using pyModis
- **i.sentinel**: Toolset for download and processing of Copernicus Sentinel products
- **r.hants**: Approximates a periodic time series with harmonics
- **r.seasons**: Extracts seasons from a time series
- **r.bioclim**: Calculates bioclimatic indices
- ... and other 300+ more in the official repo!!



Don't forget to check <https://grass.osgeo.org/grass7/manuals/addons/> from time to time

GRASS  
GIS

**Thanks for your attention!!**





Move on to:

## Exercise 1: Getting familiar with GRASS GIS

Presentation powered by

