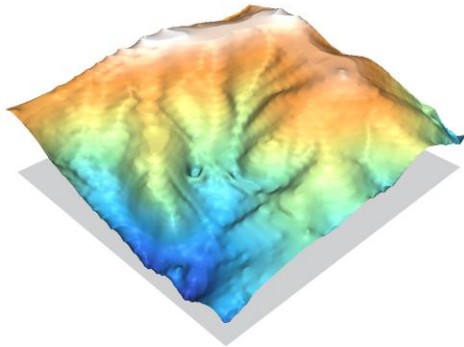


GRASS 8.x Crash course



Giorgio Agugiaro

Last update: 29 July 2026



License

This presentation is licensed under the [Creative Commons License CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/). According to CC BY-NC-SA 4.0 permission is granted to share this document, i.e. copy and redistribute the material in any medium or format, and to adapt it, i.e. remix, transform, and build upon the material under the following conditions:

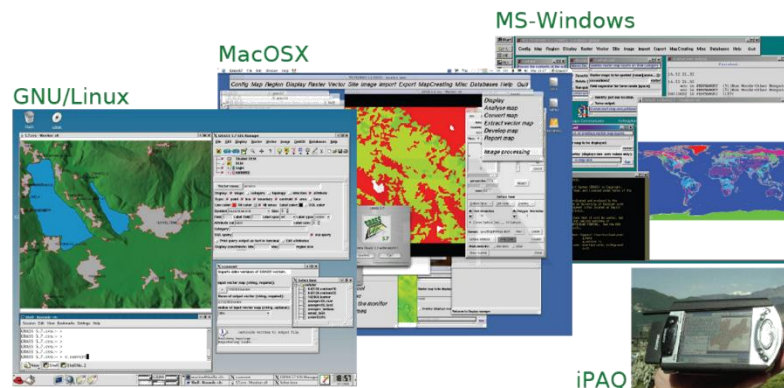


- **Attribution:** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **NonCommercial:** You may not use the material for commercial purposes.
- **ShareAlike:** If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions:** You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Brief overview of GRASS

Some history

- Developed since 1982, first version as GRASS in 1985
- Licensed under GPL since 1999
- Written in C, but portable (and already ported) to many platforms
- Homepage: <https://grass.osgeo.org>
- Short movie about GRASS (1987) narrated by William Shatner (Capt. Kirk!)
<https://www.youtube.com/watch?v=U3Hf0qI4JLc> (14:27)



Intro

GRASS modules

GRASS DB

Running GRASS

New setup

Running modules

Test data

Help

GRASS modules

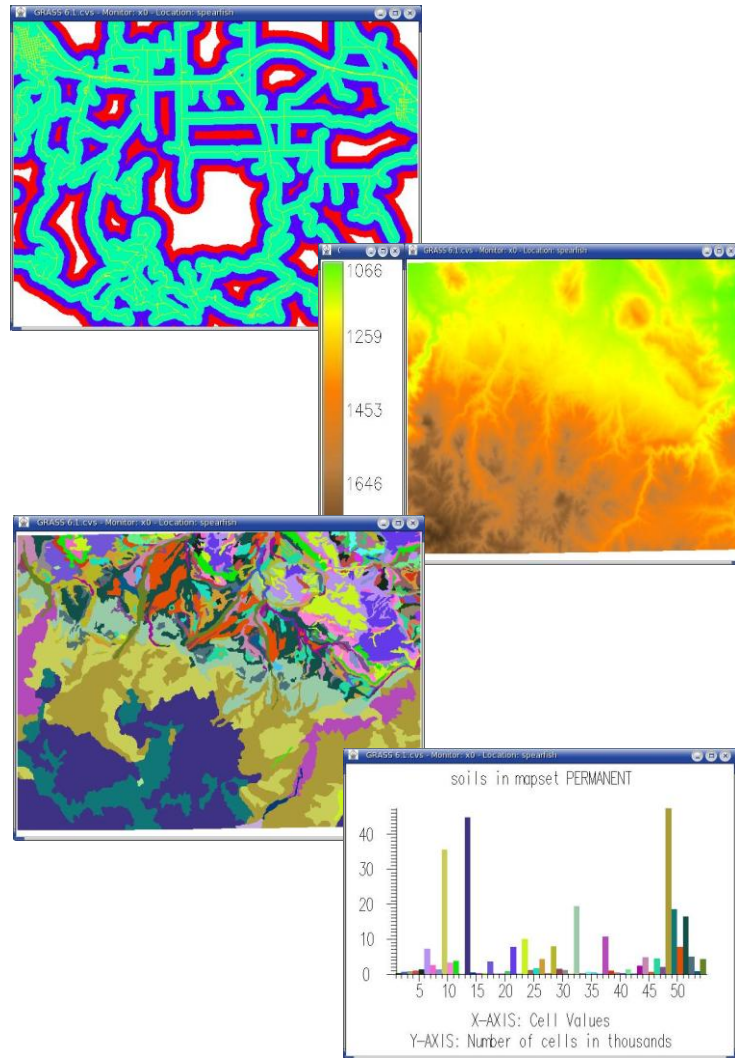
- GRASS commands (or **modules**) are generally thematically grouped according to prefixes in the command name
- They can be used either from the command line, or the GRASS GUI, or from QGIS

<i>GRASS Command Overview</i>			
<i>prefix</i>	<i>function class</i>	<i>type of command</i>	<i>example</i>
d.*	display	graphical output	d.rast: views raster map d.vect: views vector map
db.*	database	database management	db.select: selects value(s) from table
g.*	general	general file operations	g.rename: renames map
i.*	imagery	image processing	i.smap: image classifier
ps.*	postscript	map creation in Postscript format	ps.map: map creation
r.*	raster	raster data processing	r.buffer: buffer around raster features r.mapcalc: map algebra
r3.*	voxel	raster voxel data processing	r3.mapcalc: volume map algebra
v.*	vector	vector data processing	v.overlay: vector map intersections

GRASS modules

Typical **raster** operations

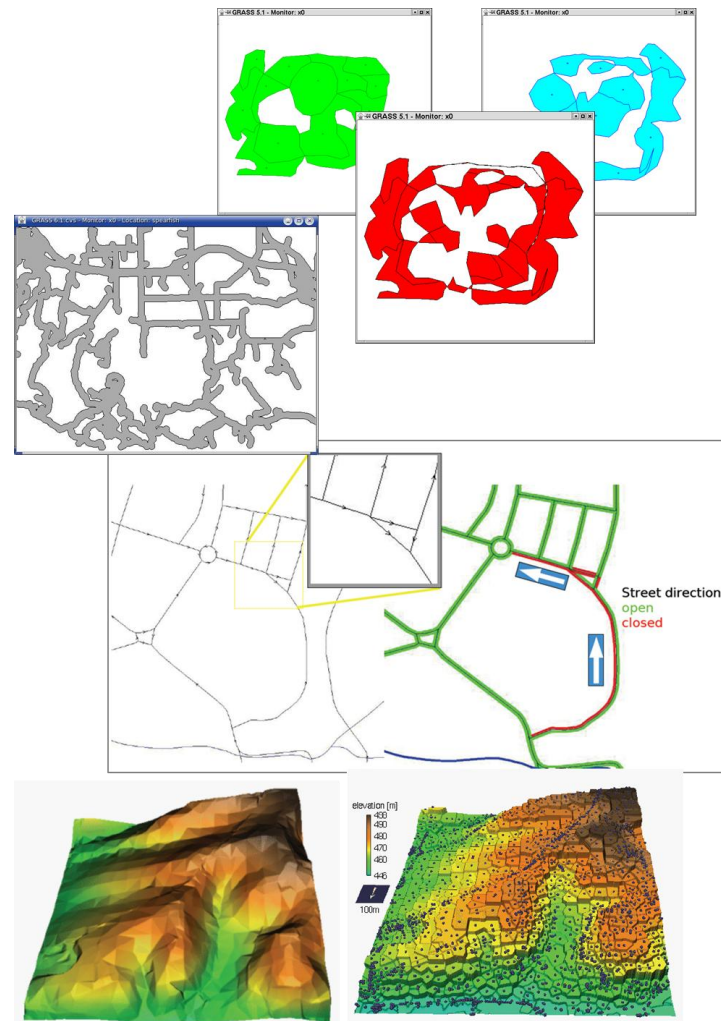
- Vectorization
 - r.to.vect
 - r.contour (creates contour lines)
- Digitalization
 - r.digit
- Creation of buffers
 - r.buffer
- Classification
 - r.reclass, r.recode
- Map algebra
 - r.mapcalc, r.stats, r.neighbors, ...
- DEM creation and analysis
 - interpolation (IDW, spline, ...)
 - slope, aspect, shaded relief maps, etc.
- Surface cost analysis
 - r.cost, r.drain
- Specific analyses (astronomy, hydrology, ...)
 - r.sun, r.watershed, ...



GRASS modules

Typical **vector** operations

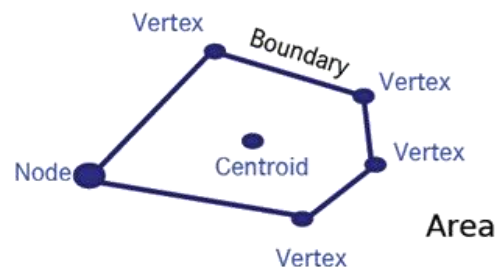
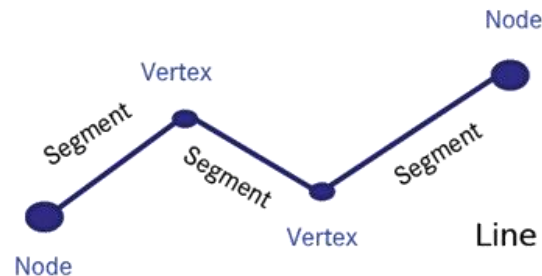
- Rasterization
 - v.to.rast, ...
- Digitalization (with input of attributes)
 - v.digit
- Creation of buffers
 - v.buffer
- Classification and attribute management
 - v.reclass, v.category
- Overlay
 - v.overlay, v.patch
- Network analysis
 - v.net.xxx (several modules)
- Interpolation
 - v.voronoi, v.delaunay



GRASS vector data

- Point – 3Dpoint
- Centroid
- Line
- Boundary
- Area (boundary + centroid)
- Face (3D area)
- Kernel (3D centroid)
- Volumes (faces + kernel)

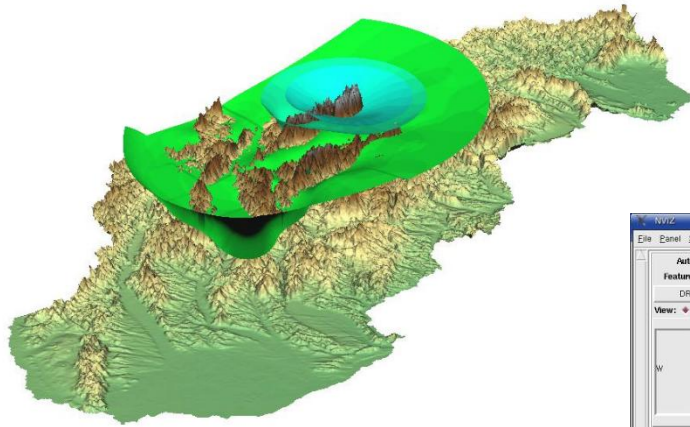
NOTA BENE: Unlike shapefiles, in GRASS different vector entities can co-exist in the same layer!



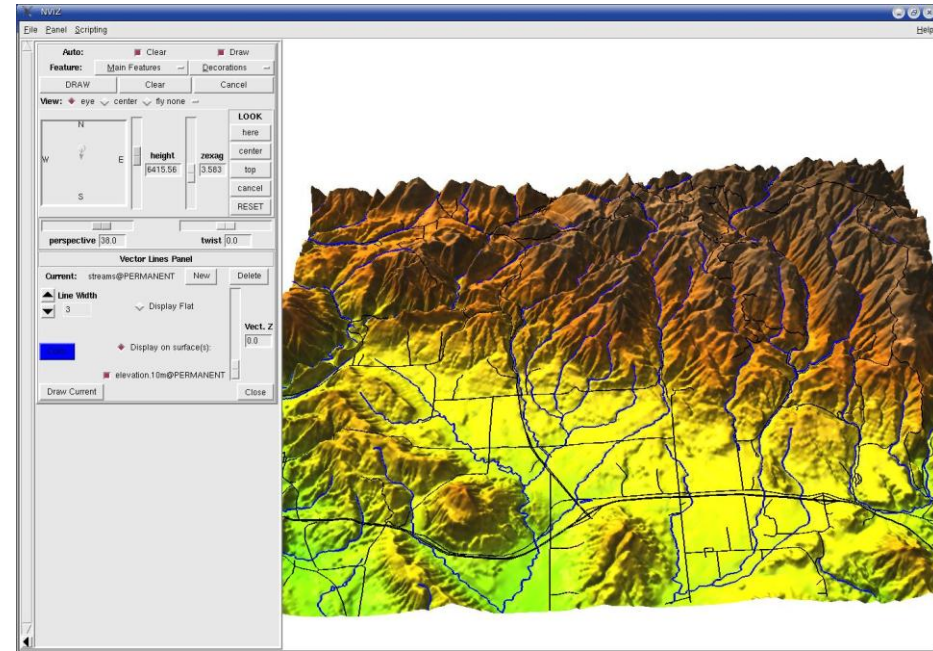
Faces



Intro
GRASS modules
 GRASS DB
 Running GRASS
 New setup
 Running modules
 Test data
 Help



NVIZ is a 3D visualization module within GRASS for 2D/3D vector, raster and voxel data

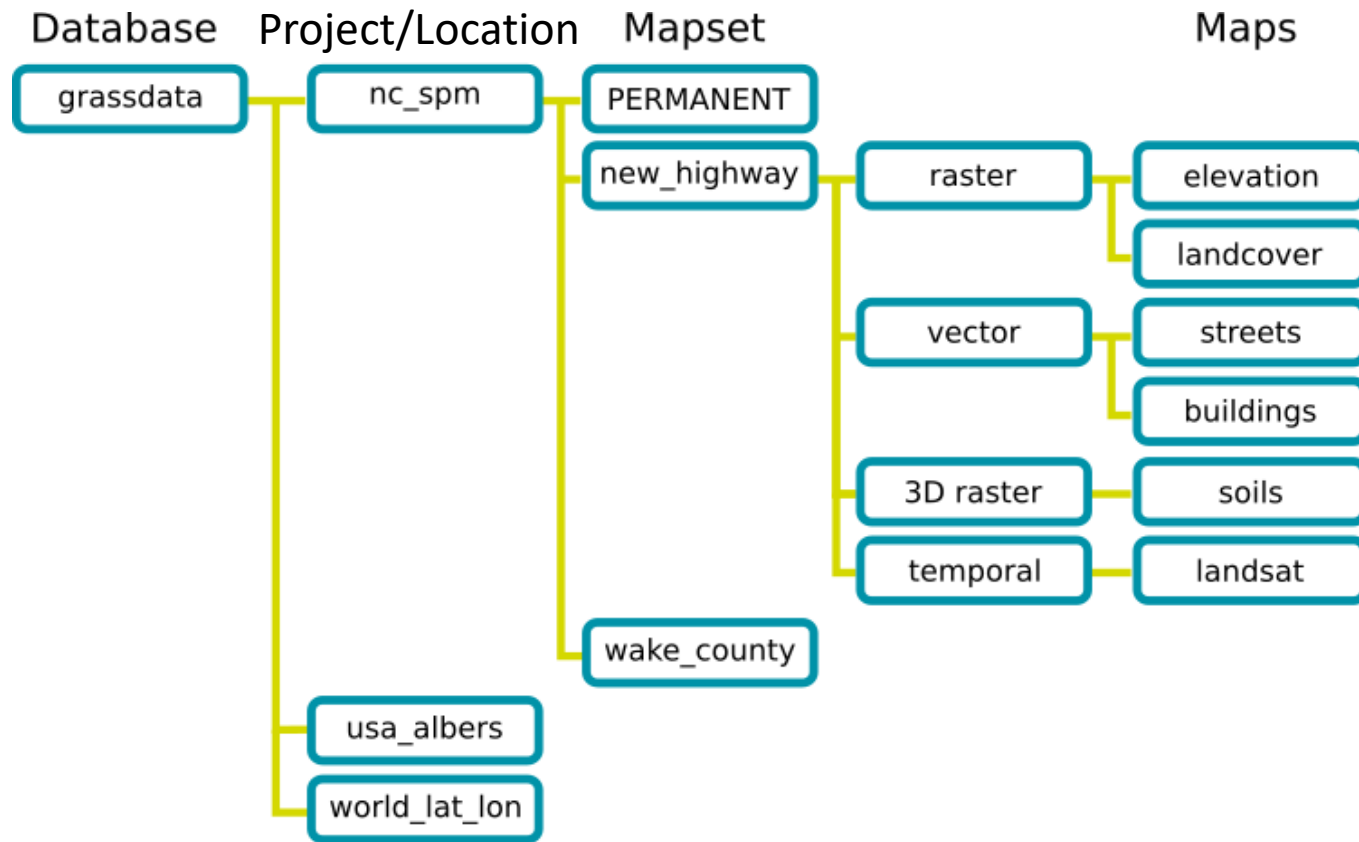


GRASS “database” structure

In GRASS, all datasets are stored inside a **single folder** called the "**GRASS Database**" and they are further divided into PROJECT(s) and MAPSET(s).

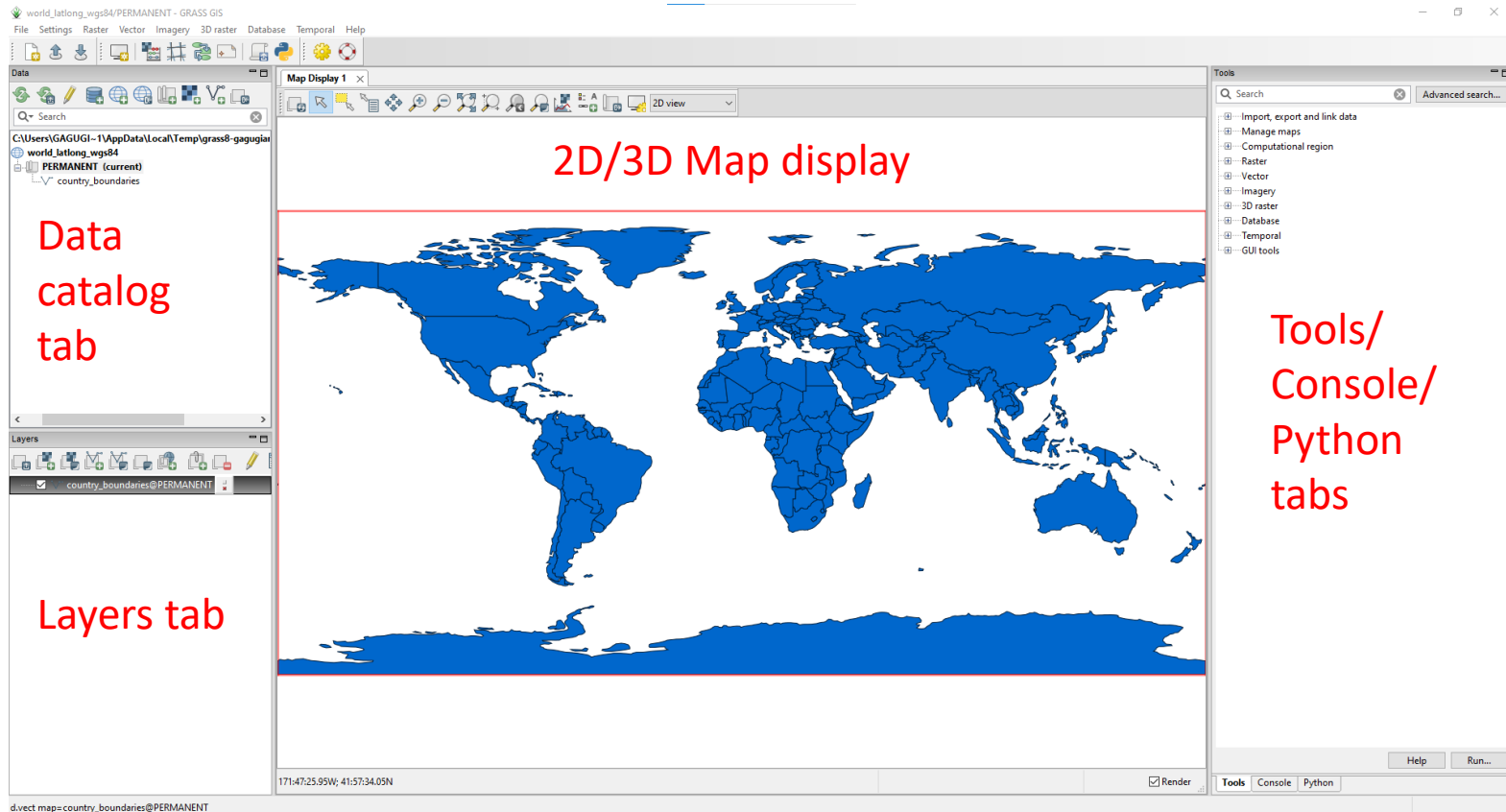
- A **PROJECT** is a folder containing all data having the same projection and – possibly – region (see later).
 - **Please note:** up to GRASS 8.4, the PROJECT folder was named **LOCATION**, with the name changed to PROJECT since version 8.5
- A **MAPSET** is a folder of a PROJECT containing thematically similar data. Mapset **PERMANENT** is always present and read-only!
 - One PROJECT can have multiple MAPSETs.
 - Each MAPSET contains files and folders of maps in vector or raster format (2D and/or 3D)
- The **REGION** defines the size and the resolution (cell size) of the work area
 - First defined upon creation of the PROJECT, but can be changed later
 - Each MAPSET can have its own region

GRASS “database” structure



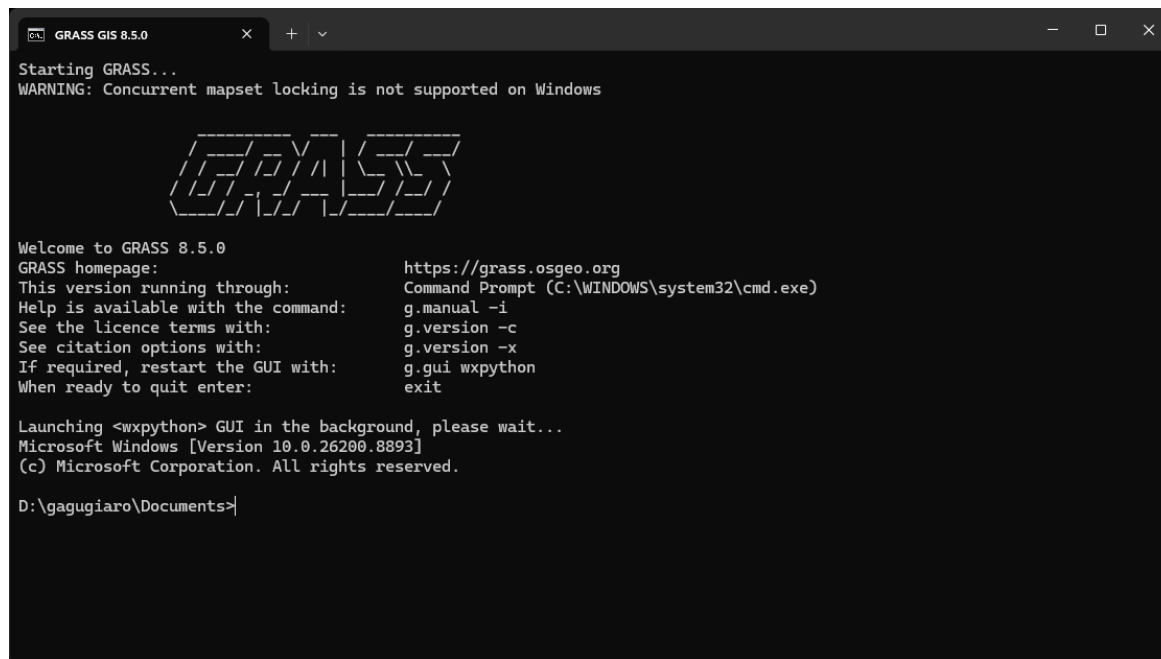
Running GRASS - GUI

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



Running GRASS – Command prompt

- In the background, minimized, there is always a window with the command prompt.
- It can be used to type GRASS commands directly, too



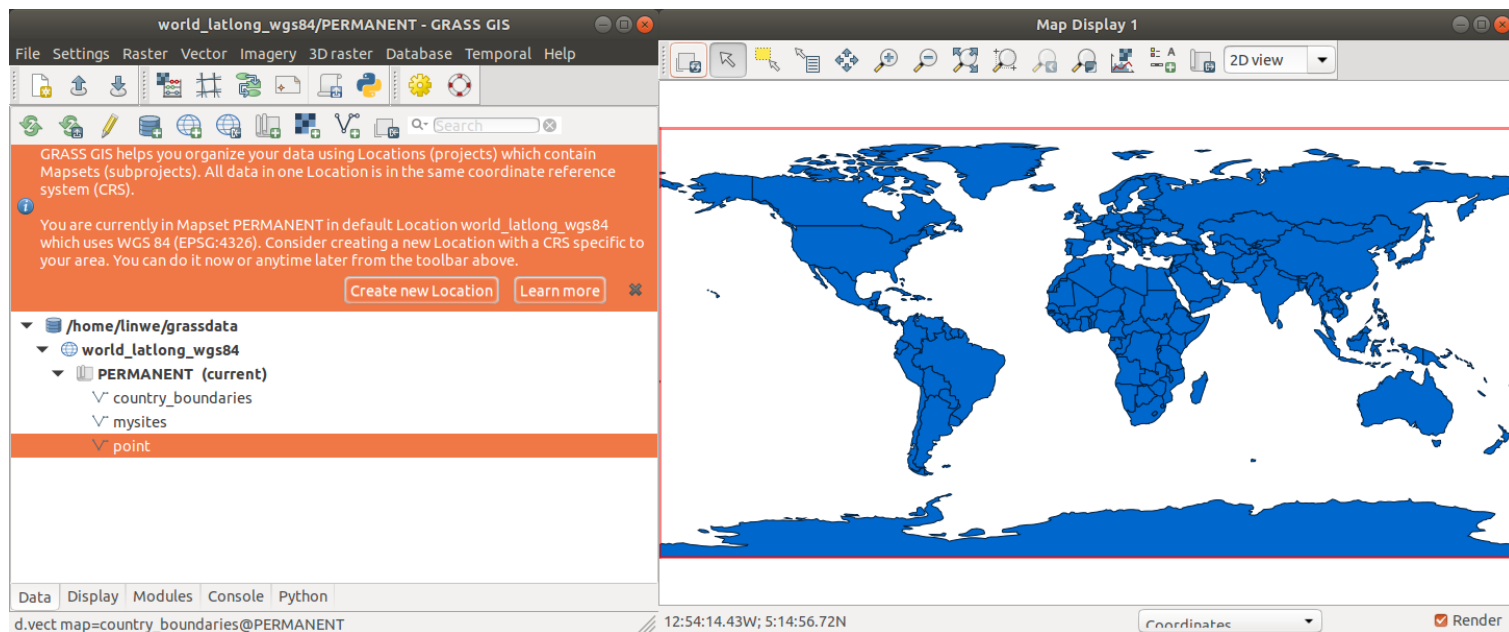
```
GRASS GIS 8.5.0
Starting GRASS...
WARNING: Concurrent mapset locking is not supported on Windows

Welcome to GRASS 8.5.0
GRASS homepage: https://grass.osgeo.org
This version running through: Command Prompt (C:\WINDOWS\system32\cmd.exe)
Help is available with the command: g.manual -i
See the licence terms with: g.version -c
See citation options with: g.version -x
If required, restart the GUI with: g.gui wxpython
When ready to quit enter: exit

Launching <wxpython> GUI in the background, please wait...
Microsoft Windows [Version 10.0.26200.8893]
(c) Microsoft Corporation. All rights reserved.

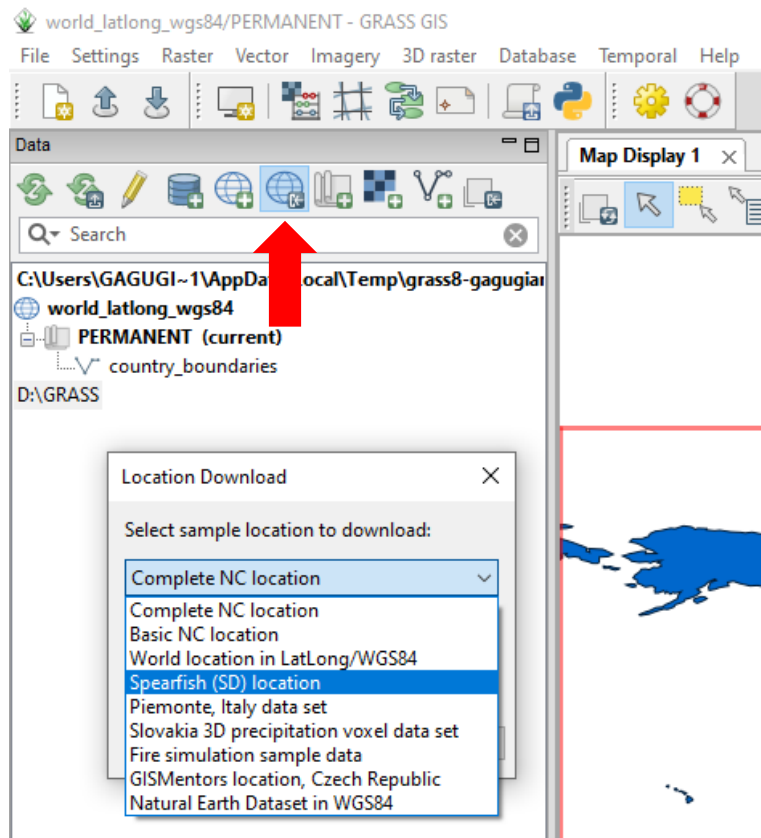
D:\gagugiaro\Documents>
```

- When launching GRASS for the first time, it will load the default Location "world_latlong_wgs84" where you can find a map layer called "country_boundaries" showing a world map in the WGS84 coordinate system
- If you know the CRS of your data or study area, you can fill **EPSG code** or description, and Location Wizard finds appropriate CRS from a predefined list of projections.



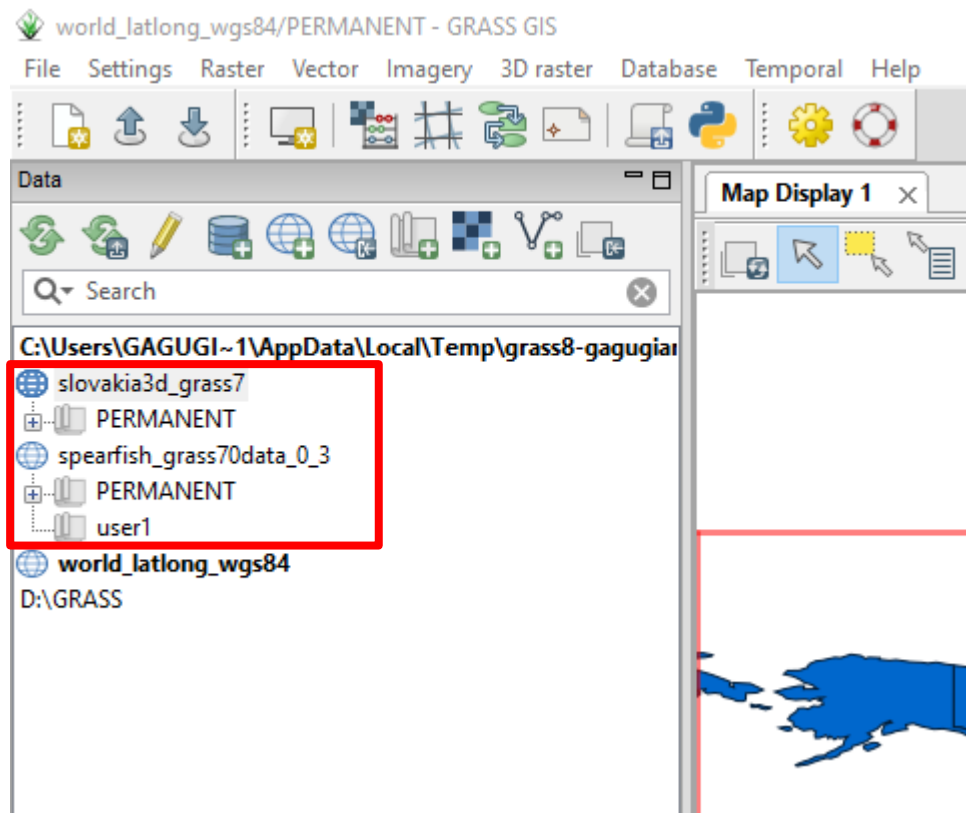
Adding sample datasets

- You can easily install sample datasets
- Click on the "Download sample location to current GRASS database" button and select the desired sample datasets. Here, for example, we choose Spearfish and Slovakia3D precipitation voxel data set



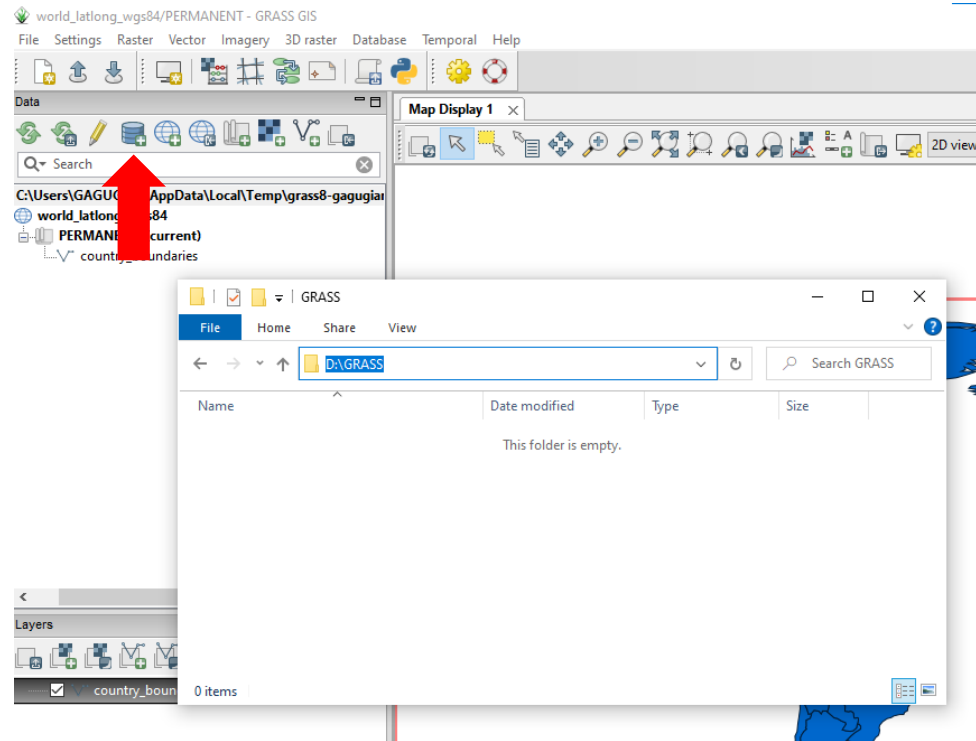
Adding sample datasets

- You can easily install sample datasets
- Click on the "Download sample location to current GRASS database" button and select the desired sample datasets. Here, for example, we choose Spearfish and Slovakia3D precipitation voxel data set
- Once installed, you will see a new locations called **slovakia3d_grass7** and **spearfish_grass70data_0_3**



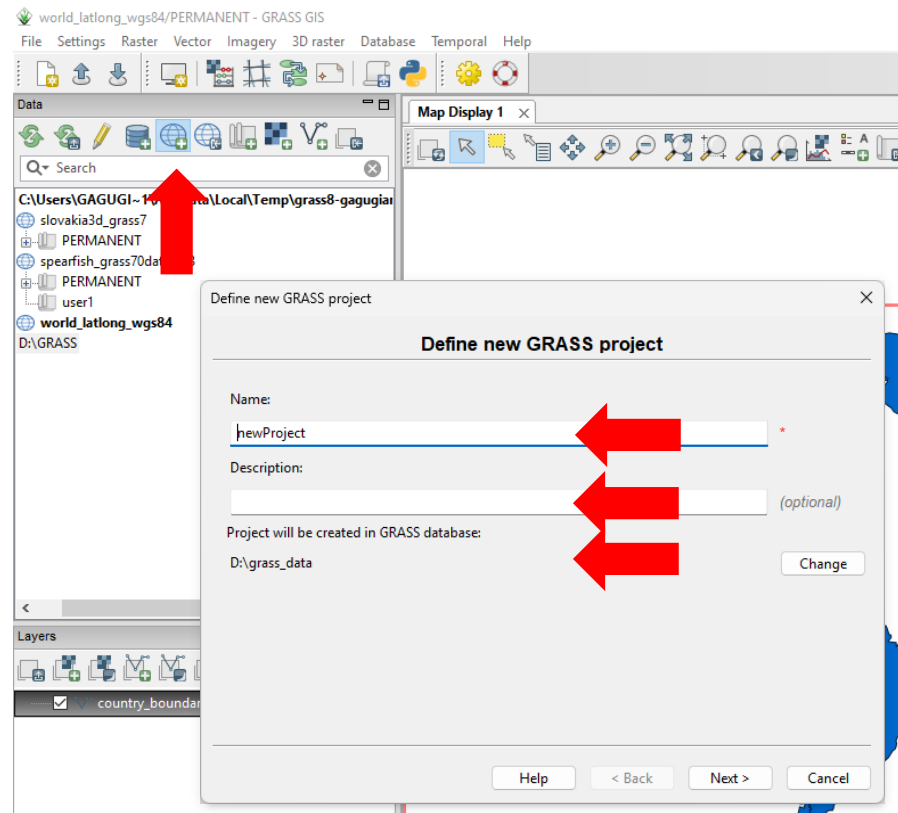
Defining a customized GRASS data folder

- It is a good idea to set up your own folder where to store your GRASS data.
- Click on the "Add existing or create new database" and choose a folder (here, for example, D:\GRASS)
- You will be immediately requested if you want to add a project/location
- If you have the necessary information, you can proceed
- Alternatively, you can also copy/paste an already existing location



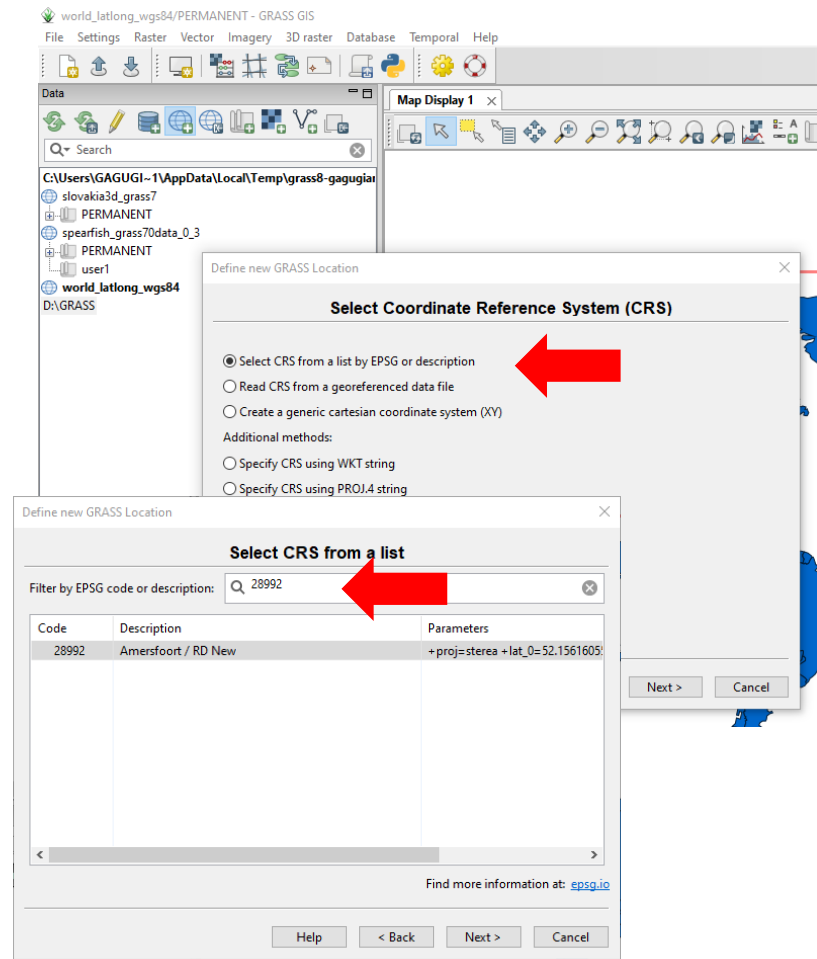
Adding a new GRASS location

- Click on the "Create new project/location in current GRASS database"
- Choose a name (e.g. Delft)
- Optionally add a description
- Optionally select which GRASS database to use (e.g. The previously set D:\GRASS)



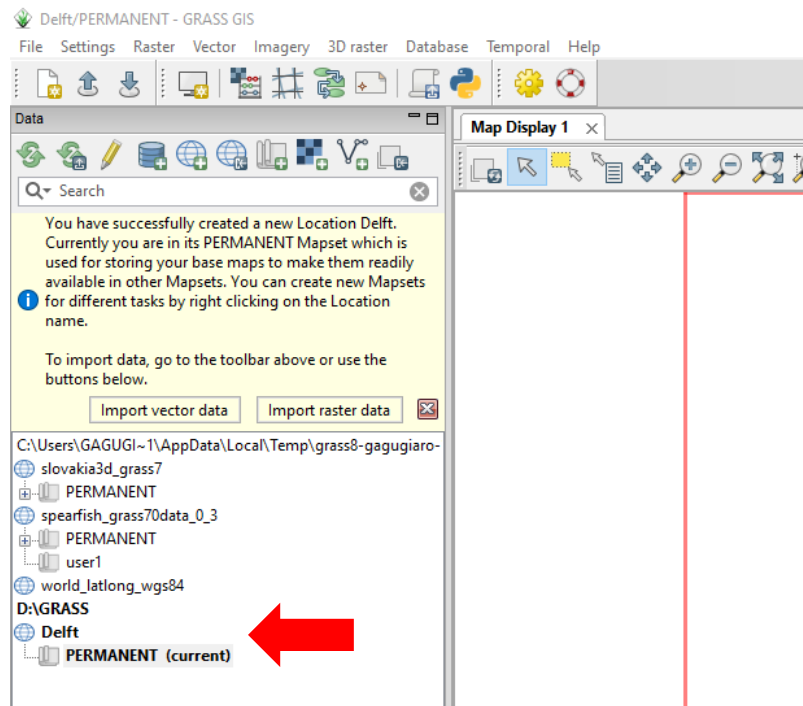
Adding a new GRASS location

- You will be asked to provide the needed parameters
- The easiest way is to choose the first option and to select the CRS from a list of EPSG codes
- You can type the code or a string and you can then pick the desired CRS. Here, for example 28992 for the NL
- Accepts the standard settings and...



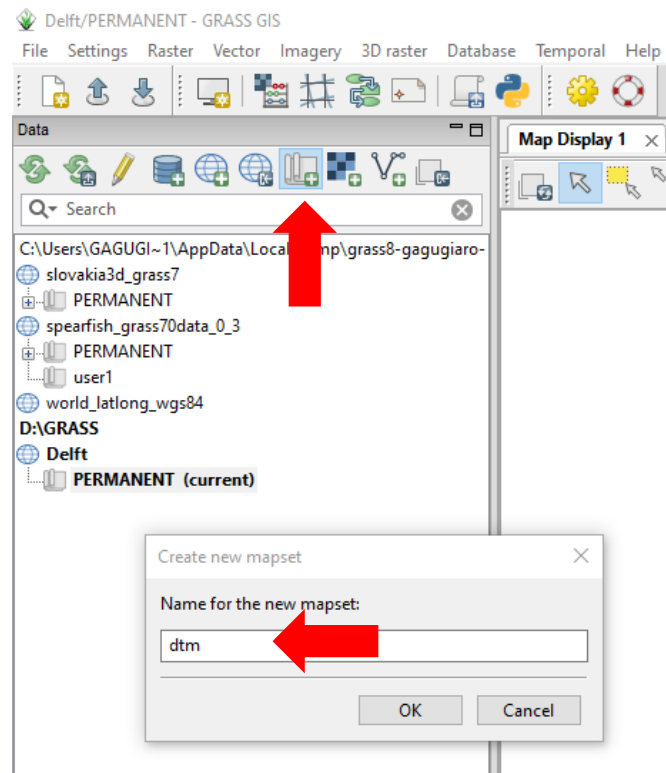
Adding a new GRASS project/location

- You will be asked to provide the needed parameters
- The easiest way is to choose the first option and to select the CRS from a list of EPSG codes
- You can type the code or a string and you can then pick the desired CRS. Here, for example 28992 for the NL
- Accepts the standard settings and...
- ...the new project/location, with the default PERMANENT mapset will be created
- You are now offered the option to directly import new data (to the PERMANENT mapset), or you can alternatively create a your own mapset



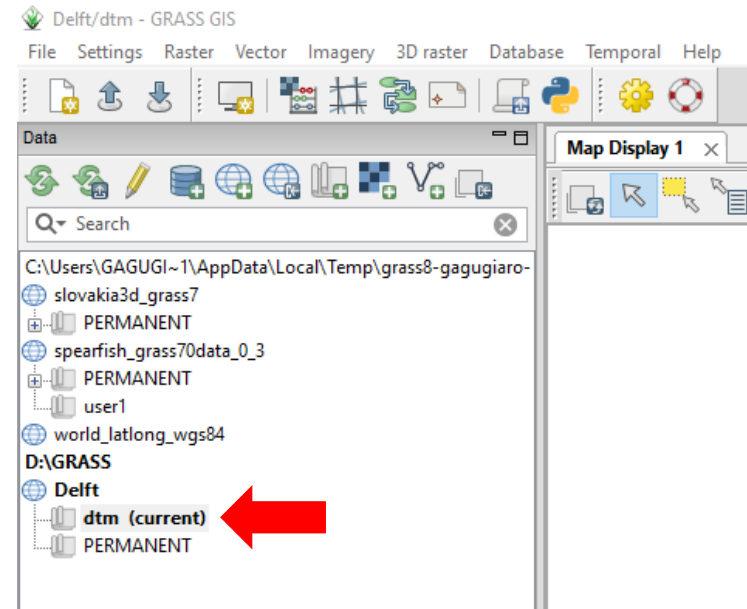
Adding a new GRASS mapset

- Click on the "Create new mapset in current project/location" button and set the name of the new mapset. Here, for example, "dtm"



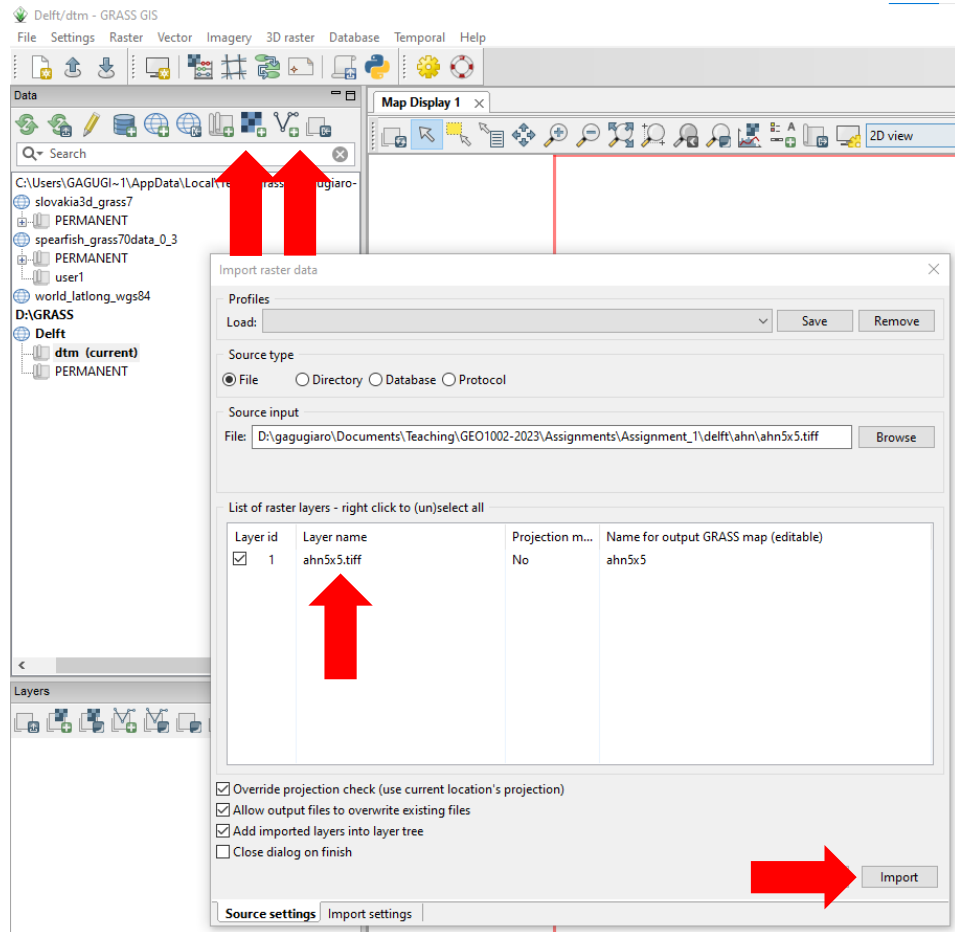
Adding a new GRASS mapset

- A new mapset is then added
- Please note the **"current" suffix**. It reminds you of the mapset you are currently working in, i.e. All maps you will create will be written in the current mapset!



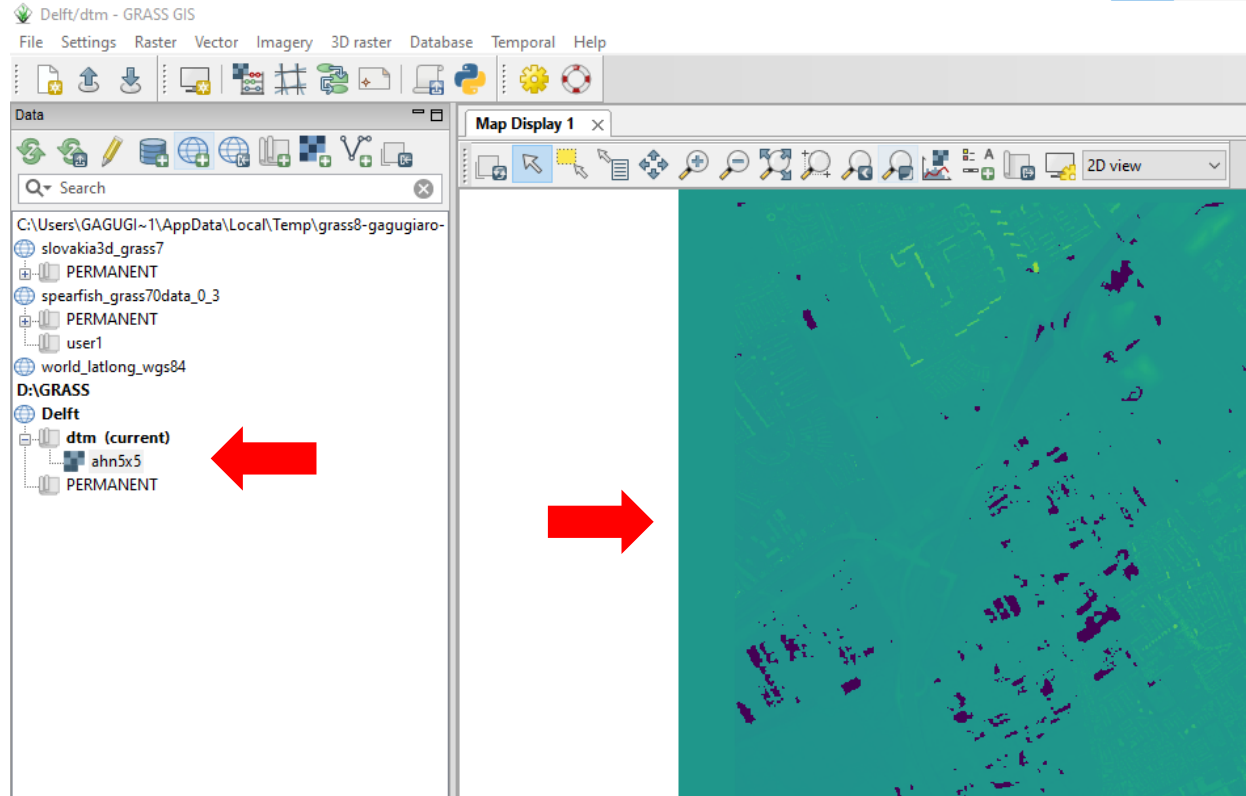
Importing data into the current mapset

- Once you have selected which mapset to work on, you can import data (vector or raster) by clicking on the proper "Import raster data" or "Import vector data" buttons
- Here, for example, we decide to import raster data. A window will open asking for the details. Here we import a till file containing the DTM of a portion of Delft
- Set the options and then click on "Import"



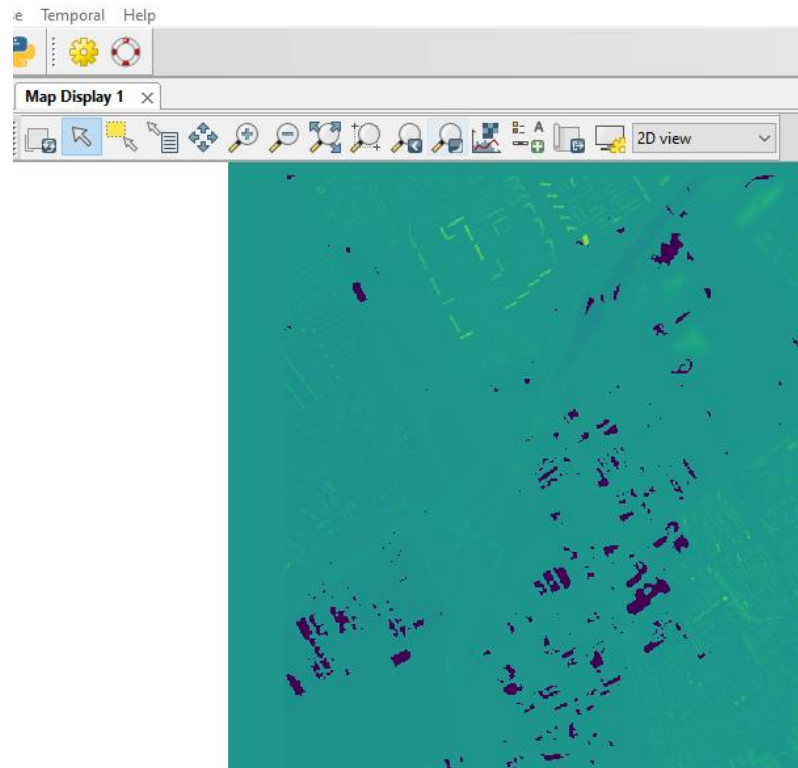
Importing data into the current mapset

- The file will be converted into a GRASS raster layer, added to the current mapset, and visualised in the Map Display



Setting the GRASS computational region

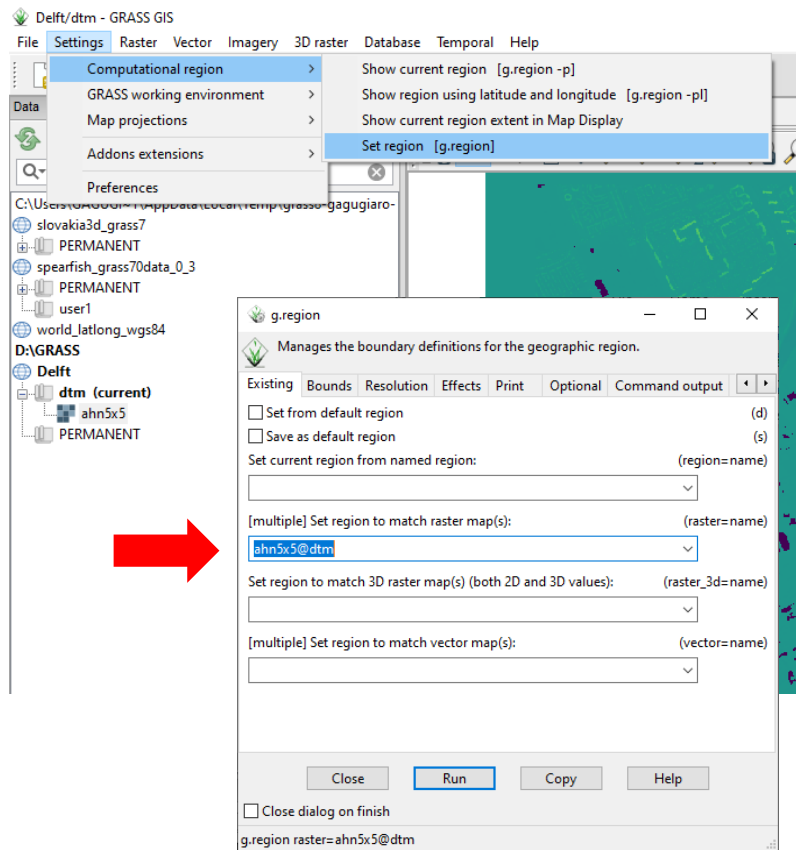
- When working in GRASS, it is FUNDAMENTAL to set the **computational region** correctly. Whatever operation you will carry out, it will be done **only within** the boundaries of such region
- The region is displayed by means of a **red boundary in the map display**
- It can be set and adapted according to your needs



No red boundary visible, the region is not set correctly!

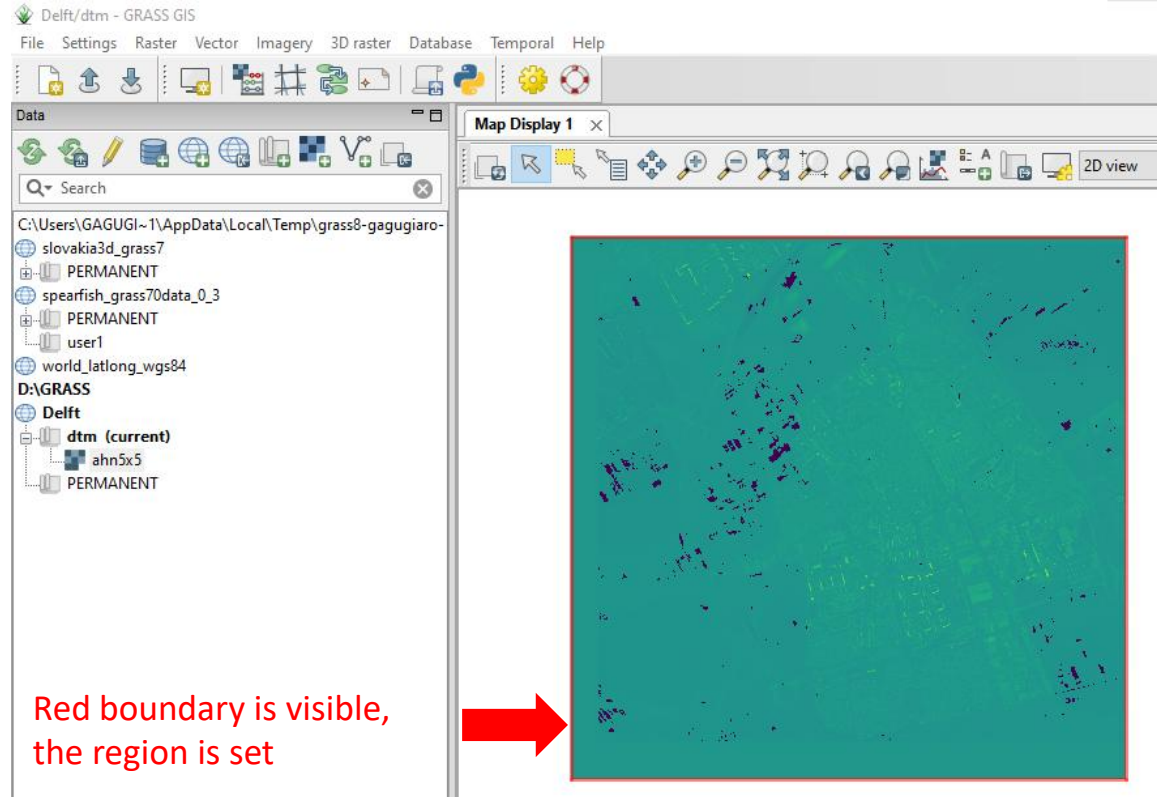
Setting the GRASS computational region

- You can set the region from the GUI via Settings\Computational region\Set region
- A command GUI will open. For example, you can set it to the extents of the previously imported DTM map (it is just an example!)



Setting the GRASS computational region

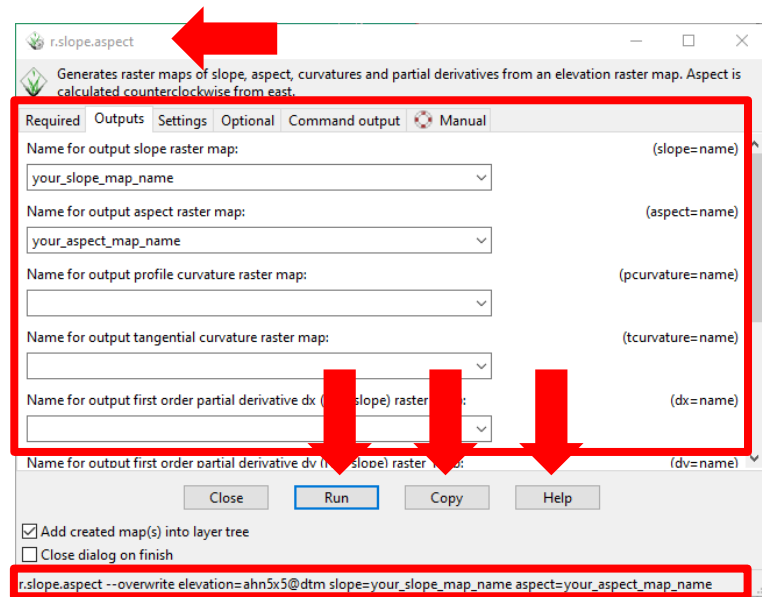
- Now you can see the current region as a red boundary around the selected raster layer



Red boundary is visible,
the region is set

Working with GRASS: Module windows

- In general, nearly every GRASS module has an associated window GUI. The name of the module is written on the top of the window
- All user-defined parameters and flags can be set manually
- Once you are done, you simply press the "Run" button
- You can see (and copy) the full command with parameters and flags in the bottom part of the window. This is useful for learning how parameters are set, and to use the command in scripts to automate the process
- For help, simply press the "Help" button



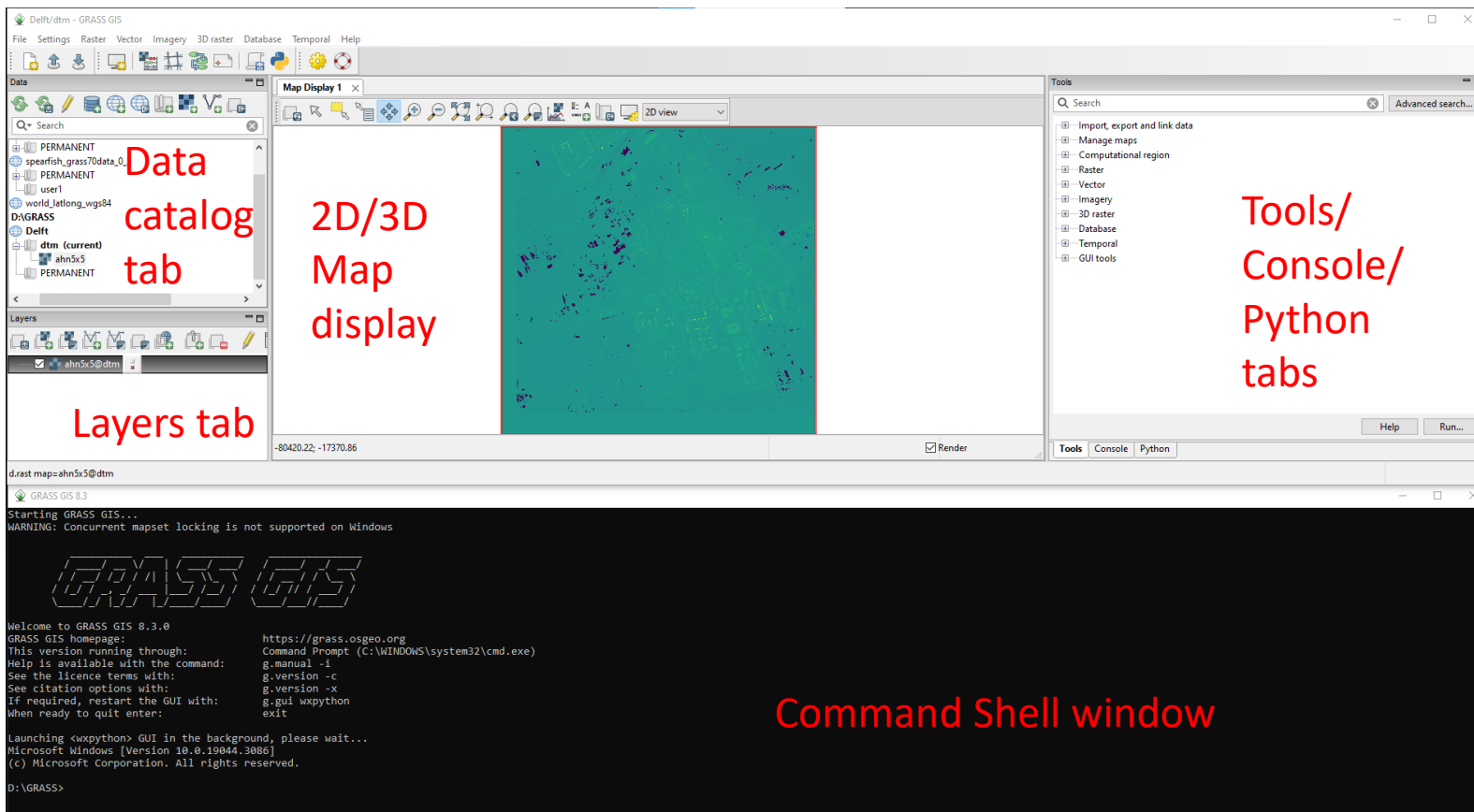
Working with GRASS

There are **different, alternative ways** to launch and run the GRASS modules:

- a) Using the **GUI drop-down menus** (the names of the modules are also suggested) with mouse and/or keyboard
- b) Double clicking on the module name in the **Tools tab**
- c) Writing the module name (with, optionally, parameters) in the **Console tab**
- d) Writing the module name (with, optionally, parameters) in the **Command shell**
- e) Writing Python code in the **Python tab**
- f) Writing Python code in the **Python code editor**
- g) From within **QGIS + GRASS Plugin** (not recommended)
- h) A combination thereof

Working with GRASS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



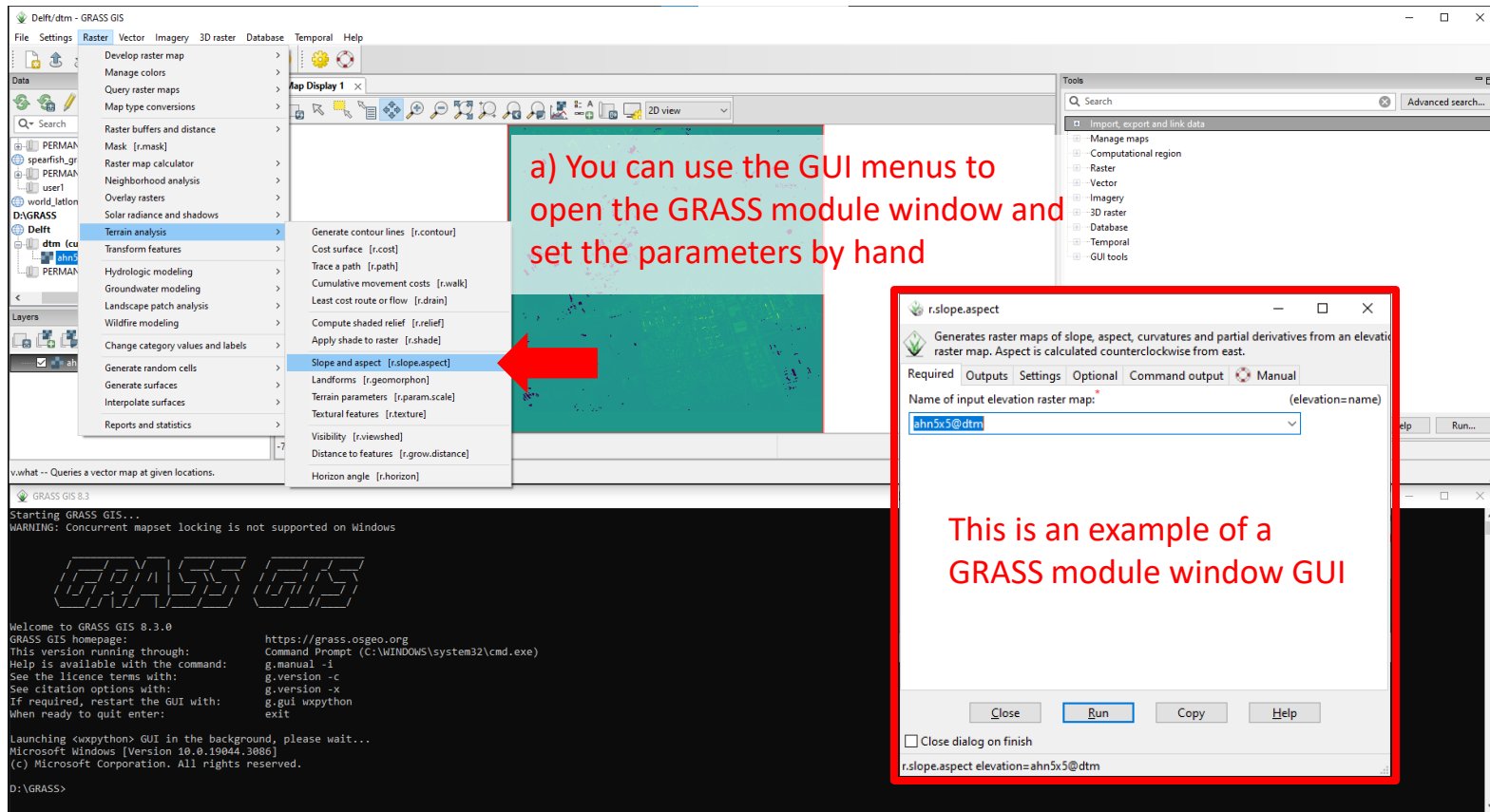
The screenshot displays the GRASS GIS 8.3.0 interface. The main window is titled 'Delft/dtm - GRASS GIS' and features a menu bar (File, Settings, Raster, Vector, Imagery, 3D raster, Database, Temporal, Help) and a toolbar. The interface is divided into several panes:

- Data catalog tab:** Located on the left, it shows a tree view of the data catalog with folders like 'PERMANENT', 'spearfish_grass70data_0', 'user1', 'world_latlong_wgs84', 'D:\GRASS', 'Delft', 'dtm (current)', 'ahn5x5', and 'PERMANENT'.
- 2D/3D Map display:** The central pane shows a 2D view of a map with a green and blue color scheme.
- Layers tab:** Located below the Data catalog, it shows a list of layers, including 'ahn5x5@dtm'.
- Tools/Console/Python tabs:** Located on the right, they provide access to various tools and scripts.
- Command Shell window:** A terminal window at the bottom showing the GRASS GIS startup process, including the 'GRASS GIS' logo and the command prompt 'D:\GRASS>'.

Red text labels are overlaid on the image to identify these components: 'Data catalog tab', '2D/3D Map display', 'Layers tab', 'Tools/Console/Python tabs', and 'Command Shell window'.

Working with GRASS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



a) You can use the GUI menus to open the GRASS module window and set the parameters by hand

This is an example of a GRASS module window GUI

The screenshot shows the GRASS GIS 8.3.0 GUI. The 'Raster' menu is open, and the 'Slope and aspect [r.slope.aspect]' option is selected. A red arrow points to this option. The 'r.slope.aspect' module window is open, showing the 'Name of input elevation raster map:' field with the value 'ahn5x5@dtm'. The window also has tabs for 'Required', 'Outputs', 'Settings', 'Optional', 'Command output', and 'Manual'. The 'Run' button is highlighted.

GRASS GIS 8.3.0

Starting GRASS GIS...

WARNING: Concurrent mapset locking is not supported on Windows

GRASS GIS

Welcome to GRASS GIS 8.3.0

GRASS GIS homepage: <https://grass.osgeo.org>

This version running through:

Help is available with the command:

See the licence terms with:

See citation options with:

If required, restart the GUI with:

When ready to quit enter:

Launching <wxpython> GUI in the background, please wait...

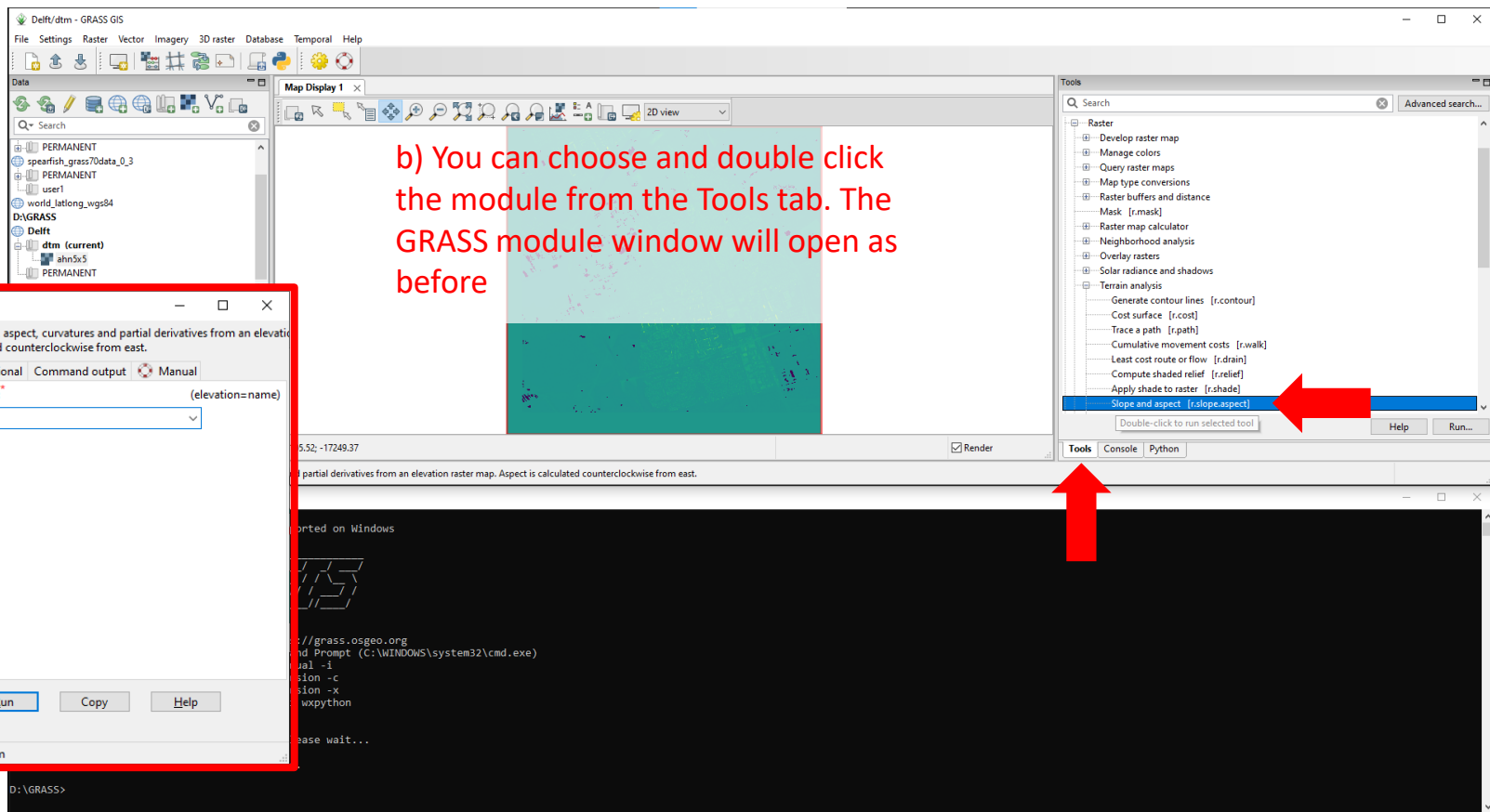
Microsoft Windows [Version 10.0.19044.3086]
(c) Microsoft Corporation. All rights reserved.

D:\GRASS>

Working with GRASS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Runn
Test d
Help

b) You can choose and double click the module from the Tools tab. The GRASS module window will open as before



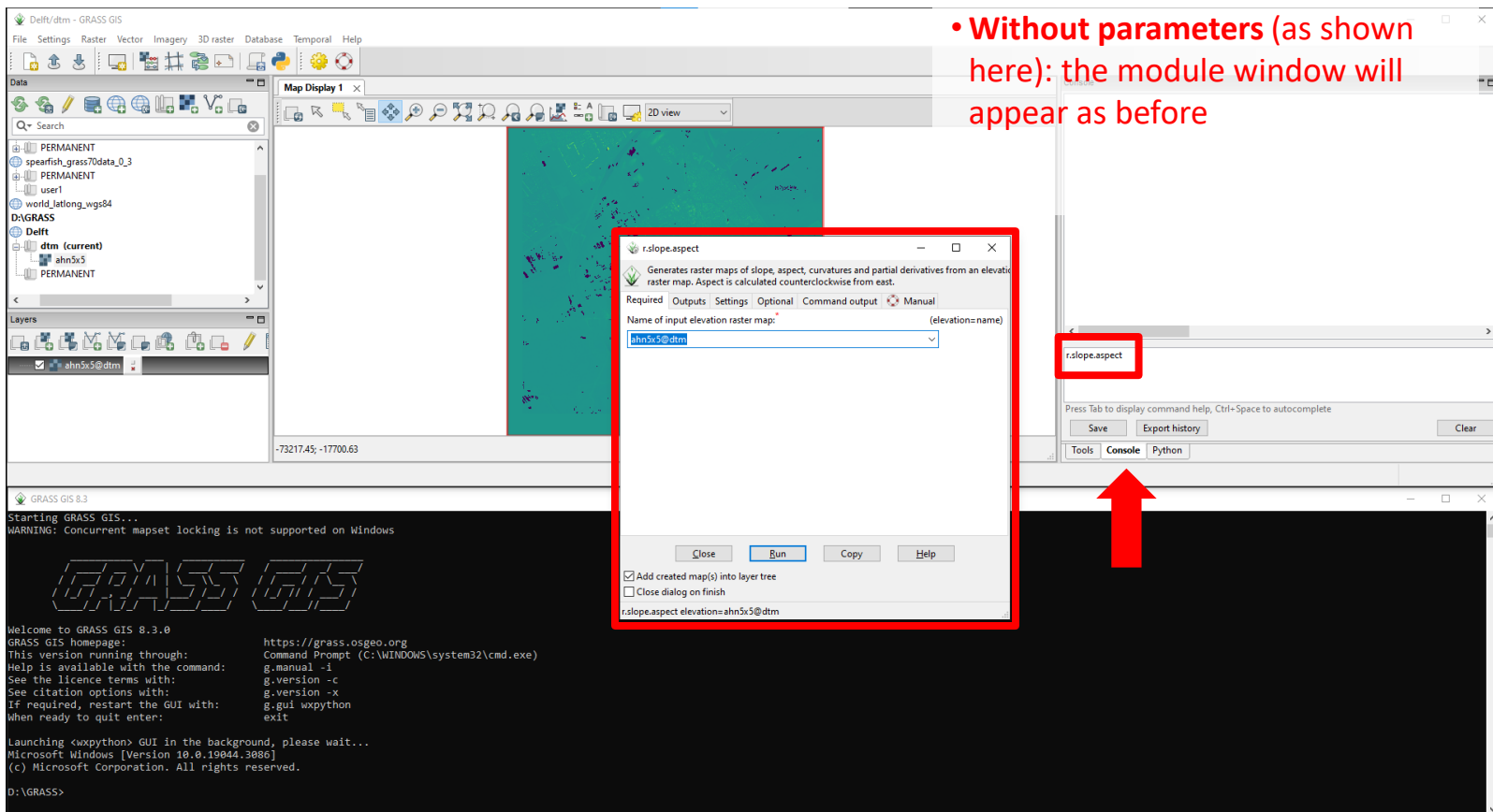
The screenshot displays the GRASS GIS environment. The main window shows a map display with a terrain visualization. The 'Tools' sidebar on the right contains a list of modules, with 'Slope and aspect [r.slope.aspect]' highlighted. The 'r.slope.aspect' dialog box is open in the foreground, showing the 'Required' tab. The 'Name of input elevation raster map:' field is set to 'ahn5x5@dtm'. The 'Run' button is highlighted. The bottom status bar shows the command 'r.slope.aspect elevation=ahn5x5@dtm'.

Working with GRASS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help

c1) You can write the module name in the Console tab. Two options:

- **Without parameters** (as shown here): the module window will appear as before



The screenshot shows the GRASS GIS 8.3.0 interface. The main window displays a map of a terrain. The 'Data' panel on the left shows the project structure, including the 'dtm (current)' layer. The 'Layers' panel at the bottom shows the 'ahns5@dtm' layer. The 'r.slope.aspect' module window is open, showing the 'Name of input elevation raster map:' field with the value 'ahns5@dtm'. The 'Console' tab is selected, showing the command 'r.slope.aspect elevation=ahns5@dtm'. A red arrow points to the 'Console' tab.

GRASS GIS 8.3.0

Starting GRASS GIS...

WARNING: Concurrent mapset locking is not supported on Windows

Welcome to GRASS GIS 8.3.0

GRASS GIS homepage: <https://grass.osgeo.org>

This version running through: Command Prompt (C:\WINDOWS\system32\cmd.exe)

Help is available with the command: g.manual -i

See the licence terms with: g.version -c

See citation options with: g.version -x

If required, restart the GUI with: g.gui wxpython

When ready to quit enter: exit

Launching <wxpython> GUI in the background, please wait...

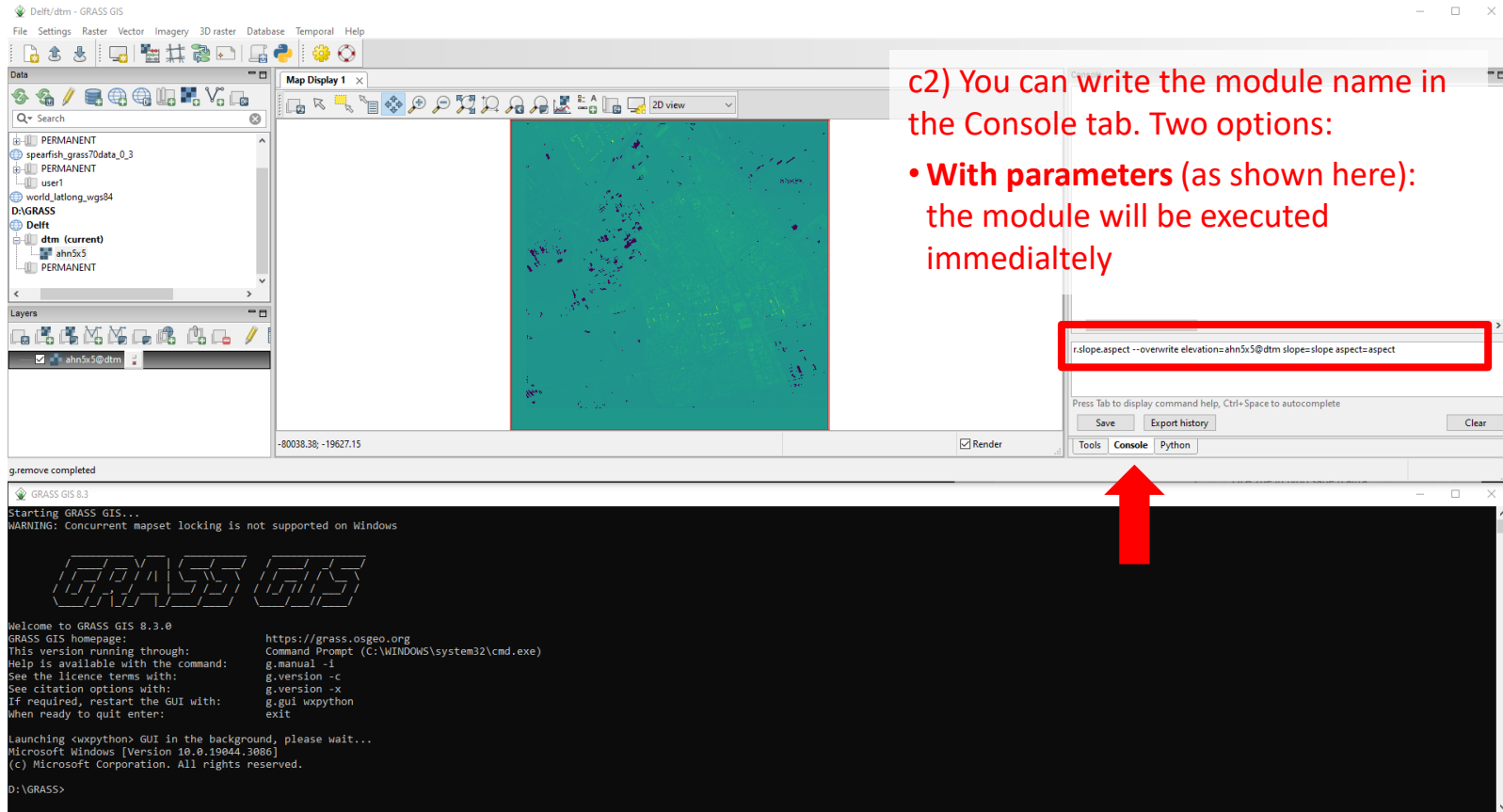
Microsoft Windows [Version 10.0.19044.3086]

(c) Microsoft Corporation. All rights reserved.

D:\GRASS>

Working with GRASS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



The screenshot shows the GRASS GIS 8.3.0 interface. The top menu bar includes File, Settings, Raster, Vector, Imagery, 3D raster, Database, Temporal, and Help. The left sidebar contains a 'Data' panel with a search bar and a list of layers, including 'D:\GRASS' and 'Delt'. The main map display shows a 2D view of a terrain map. The bottom status bar indicates the coordinates -80038.38, -19627.15.

On the right side, a red box highlights the console window. The console displays the command:

```
r.slope.aspect --overwrite elevation=ahn5x5@dtm slope=slope aspect=aspect
```

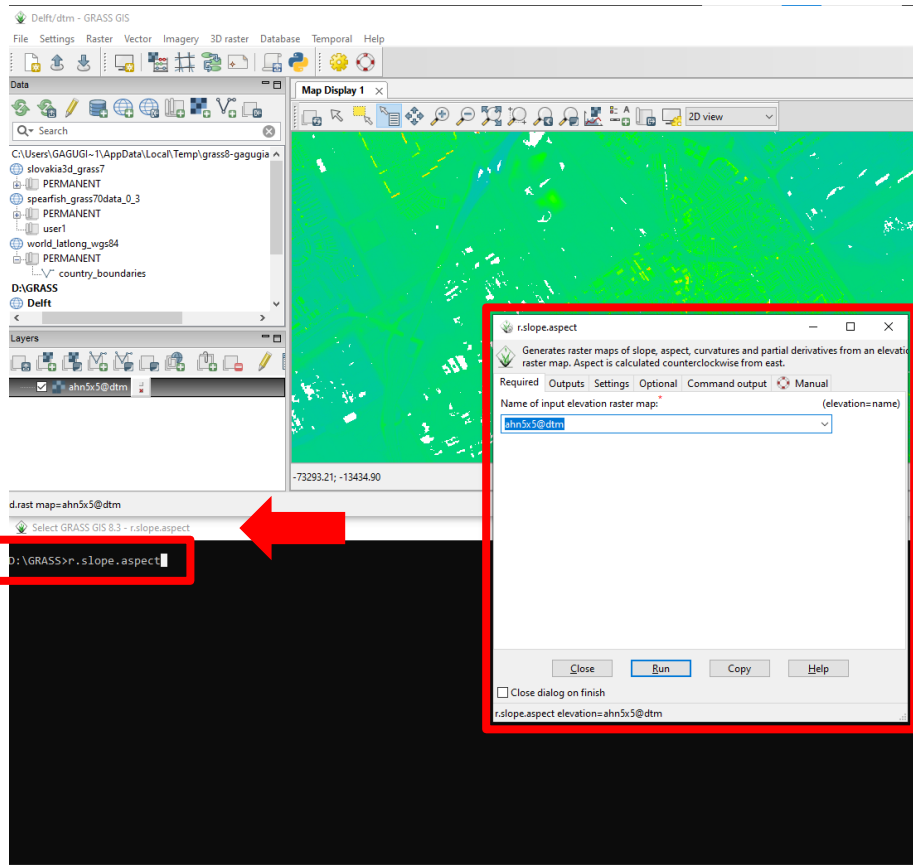
A red arrow points to the 'Console' tab in the bottom right corner of the interface.

c2) You can write the module name in the Console tab. Two options:

- **With parameters** (as shown here): the module will be executed immediately

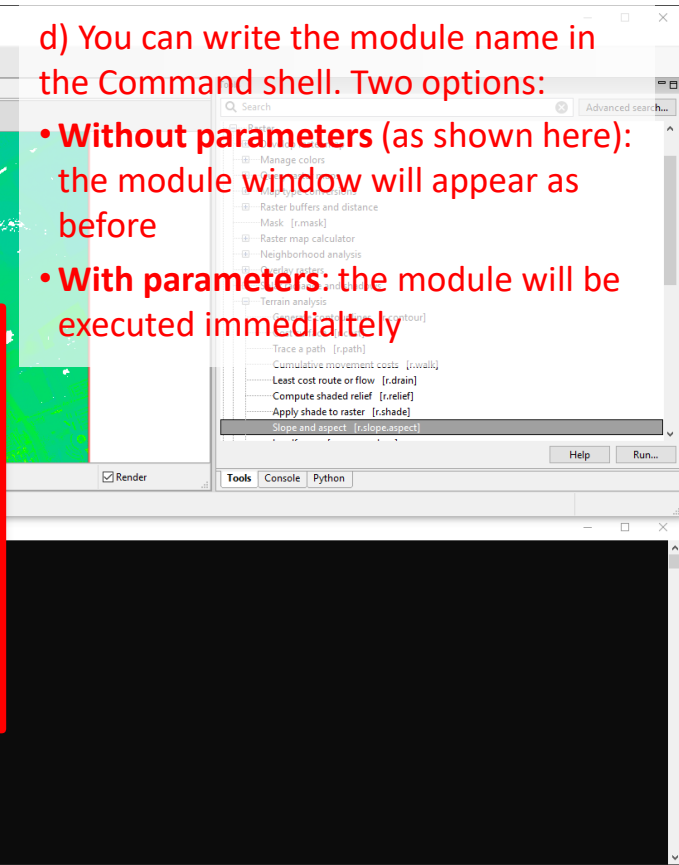
Working with GRASS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



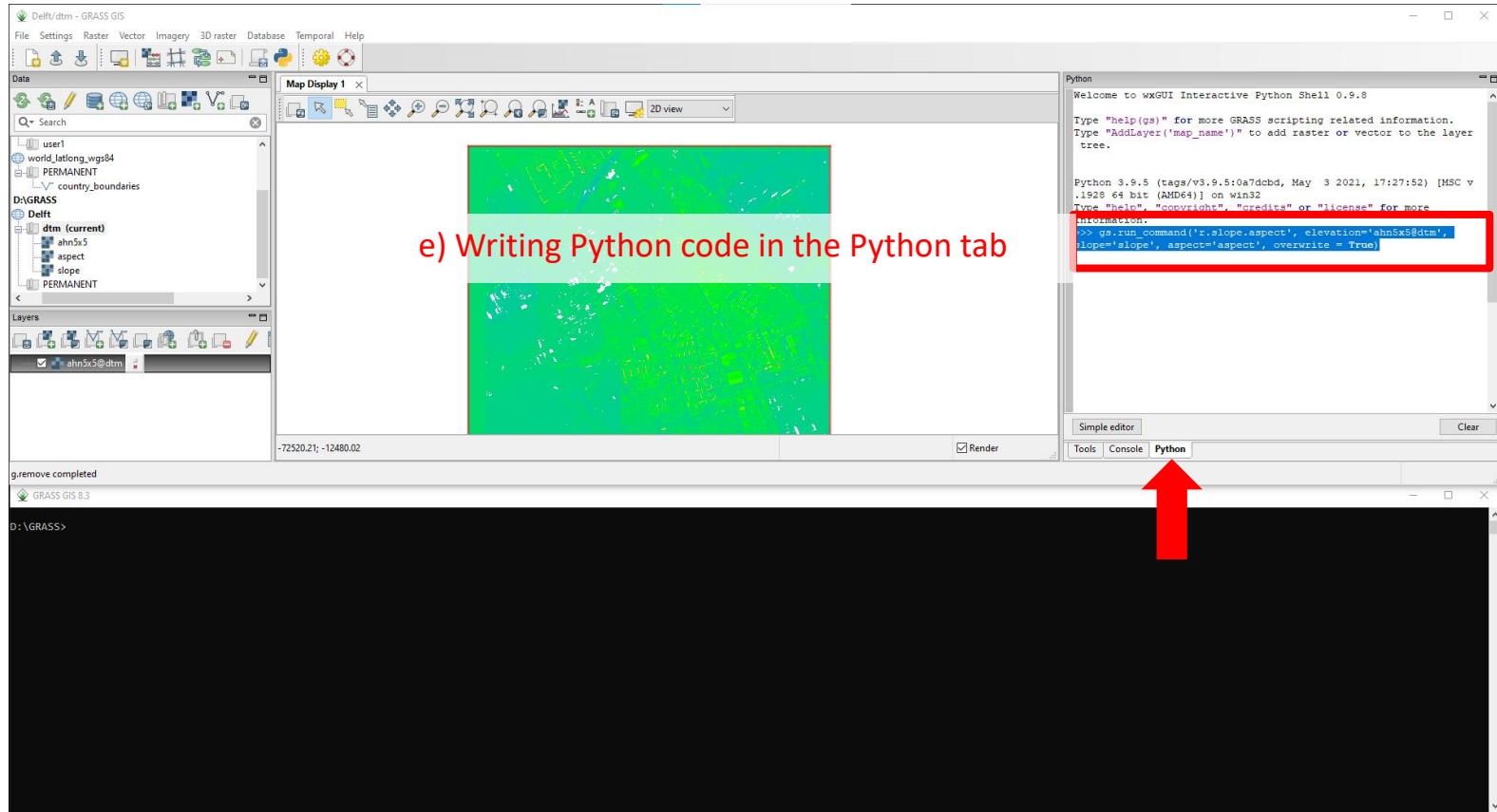
d) You can write the module name in the Command shell. Two options:

- **Without parameters** (as shown here): the module window will appear as before
- **With parameters**: the module will be executed immediately



Working with GRASS

Intro
 GRASS modules
 GRASS DB
 Running GRASS
 New setup
 Running modules
 Test data
 Help



The screenshot shows the GRASS GIS 8.3 interface. The main map display shows a 2D view of a terrain. The left sidebar contains the Data and Layers panels. The right sidebar contains the Python tab, which is highlighted with a red arrow. The Python tab shows the wxGUI Interactive Python Shell 0.9.8. The shell contains the following code:

```

Welcome to wxGUI Interactive Python Shell 0.9.8

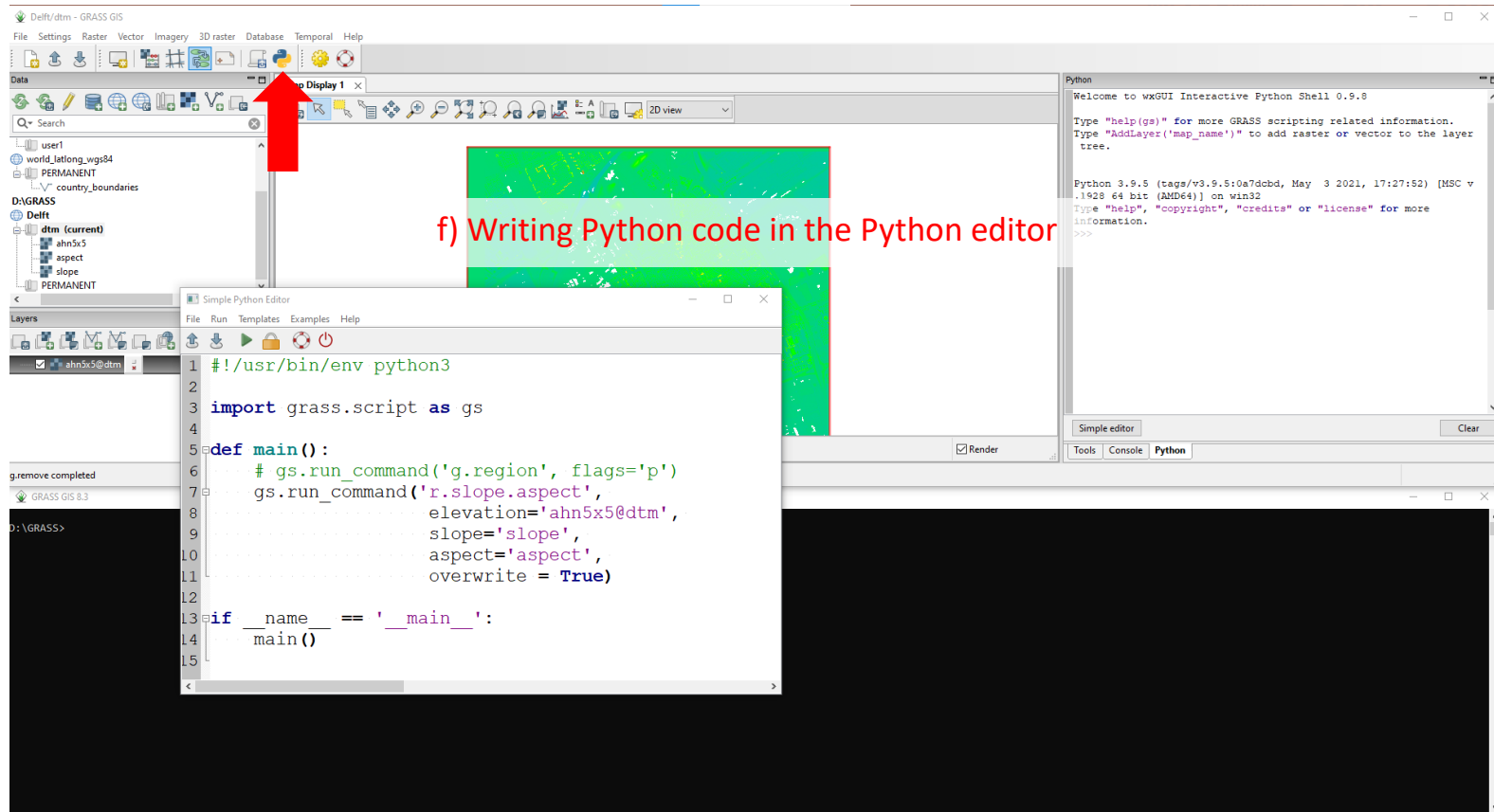
Type "help(gs)" for more GRASS scripting related information.
Type "AddLayer('map_name')" to add raster or vector to the layer
tree.

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v
.1928 64 bit (AMD64)] on win32
Type "help()", "copyright()", "credits()" or "license()" for more
information.
>>> gs.run_command('r.slope.aspect', elevation='ahh5x5@dtm',
slope='slope', aspect='aspect', overwrite = True)
  
```

The code is highlighted with a red box. A red arrow points to the 'Python' tab in the bottom right corner of the interface.

Working with GRASS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



The screenshot shows the GRASS GIS desktop environment. The main window displays a 2D view of a map with a green and blue color scheme. A red arrow points to the 'Python' button in the top toolbar. Below the main window, the 'Simple Python Editor' is open, showing a script that runs the 'r.slope.aspect' module. To the right, the 'Python' shell is open, displaying the 'Welcome to wxGUI Interactive Python Shell 0.9.8' message and some help text.

f) Writing Python code in the Python editor

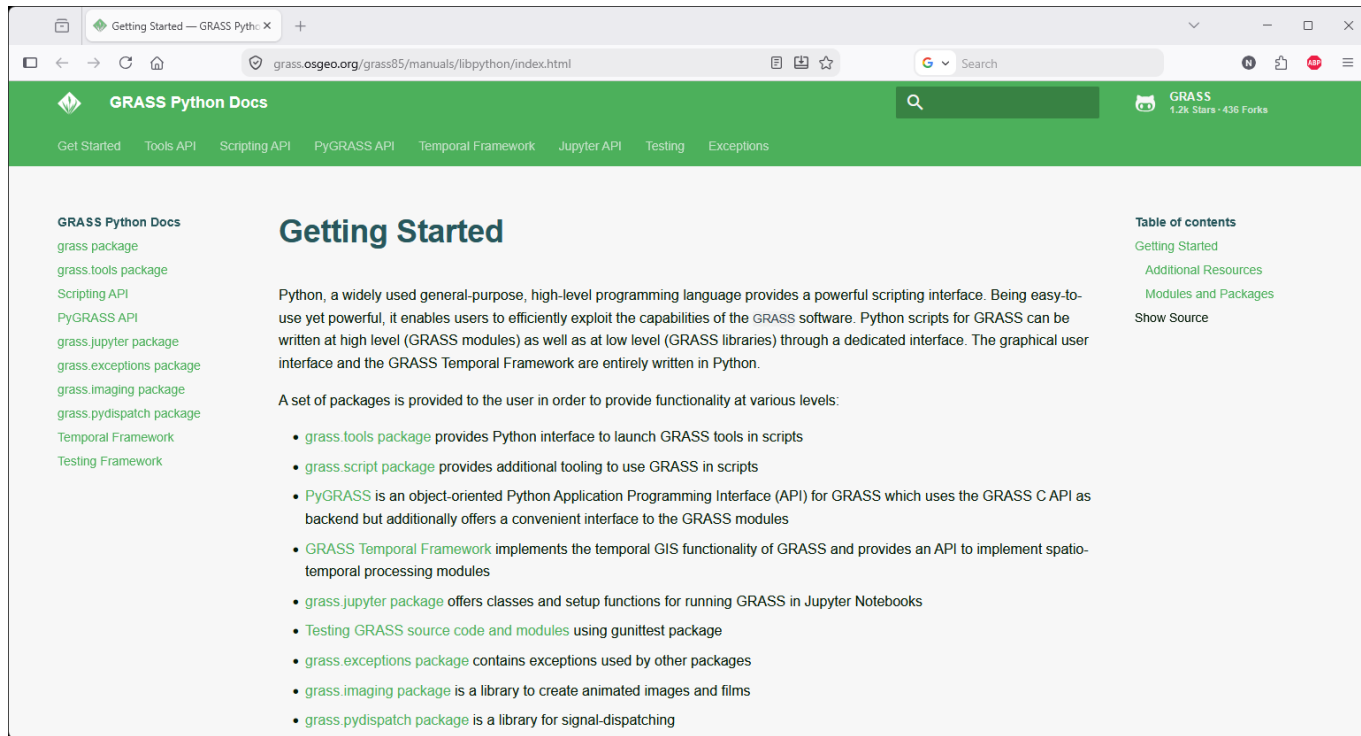
```
#!/usr/bin/env python3
import grass.script as gs

def main():
    # gs.run_command('g.region', flags='p')
    gs.run_command('r.slope.aspect',
                    elevation='ahh5x5@dtm',
                    slope='slope',
                    aspect='aspect',
                    overwrite = True)

if __name__ == '__main__':
    main()
```

GRASS and Python

- There is a useful manual providing all information on scripting GRASS with Python available at: <https://grass.osgeo.org/grass85/manuals/libpython/index.html>
 - Check for the correct GRASS version in the URL: **grass83**, **grass84**, **grass85**, etc.



The screenshot shows the 'Getting Started' page of the GRASS Python Docs. The page has a green header with the title 'GRASS Python Docs' and a search bar. Below the header, there is a navigation menu with links: 'Get Started', 'Tools API', 'Scripting API', 'PyGRASS API', 'Temporal Framework', 'Jupyter API', 'Testing', and 'Exceptions'. The main content area is titled 'Getting Started' and contains a paragraph about Python's role in GRASS, followed by a list of packages provided for users. A sidebar on the left lists various API and package links, and a sidebar on the right shows a 'Table of contents'.

GRASS Python Docs

Get Started Tools API Scripting API PyGRASS API Temporal Framework Jupyter API Testing Exceptions

Getting Started

Python, a widely used general-purpose, high-level programming language provides a powerful scripting interface. Being easy-to-use yet powerful, it enables users to efficiently exploit the capabilities of the GRASS software. Python scripts for GRASS can be written at high level (GRASS modules) as well as at low level (GRASS libraries) through a dedicated interface. The graphical user interface and the GRASS Temporal Framework are entirely written in Python.

A set of packages is provided to the user in order to provide functionality at various levels:

- [grass.tools package](#) provides Python interface to launch GRASS tools in scripts
- [grass.script package](#) provides additional tooling to use GRASS in scripts
- [PyGRASS](#) is an object-oriented Python Application Programming Interface (API) for GRASS which uses the GRASS C API as backend but additionally offers a convenient interface to the GRASS modules
- [GRASS Temporal Framework](#) implements the temporal GIS functionality of GRASS and provides an API to implement spatio-temporal processing modules
- [grass.jupyter package](#) offers classes and setup functions for running GRASS in Jupyter Notebooks
- [Testing GRASS source code and modules](#) using unittest package
- [grass.exceptions package](#) contains exceptions used by other packages
- [grass.imaging package](#) is a library to create animated images and films
- [grass.pydispatch package](#) is a library for signal-dispatching

Table of contents

- [Getting Started](#)
- [Additional Resources](#)
- [Modules and Packages](#)
- [Show Source](#)

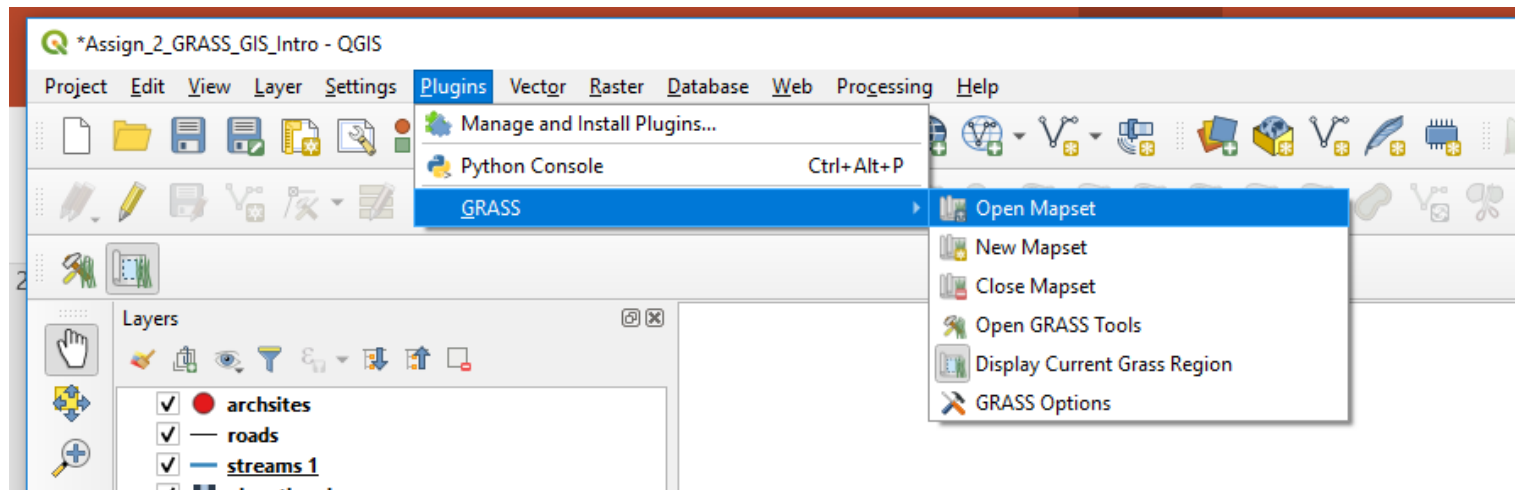
GRASS Python Docs

- [grass package](#)
- [grass.tools package](#)
- [Scripting API](#)
- [PyGRASS API](#)
- [grass.jupyter package](#)
- [grass.exceptions package](#)
- [grass.imaging package](#)
- [grass.pydispatch package](#)
- [Temporal Framework](#)
- [Testing Framework](#)

Working with GRASS: From QGIS

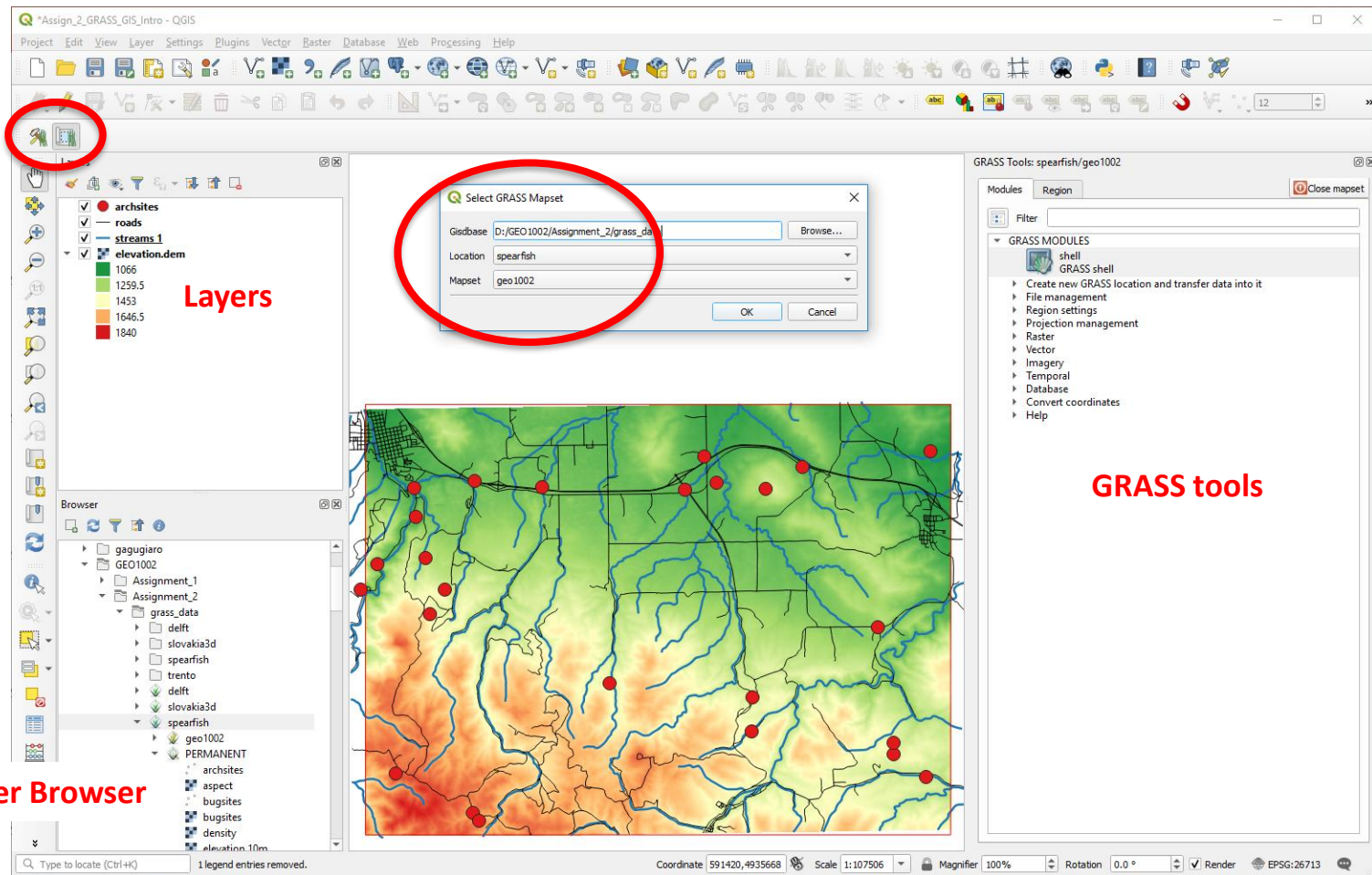
- Check that the GRASS plugin is installed and running!
- Select Plugins\GRASS\Open Mapset to configure GRASS in QGIS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



Working with GRASS: From QGIS

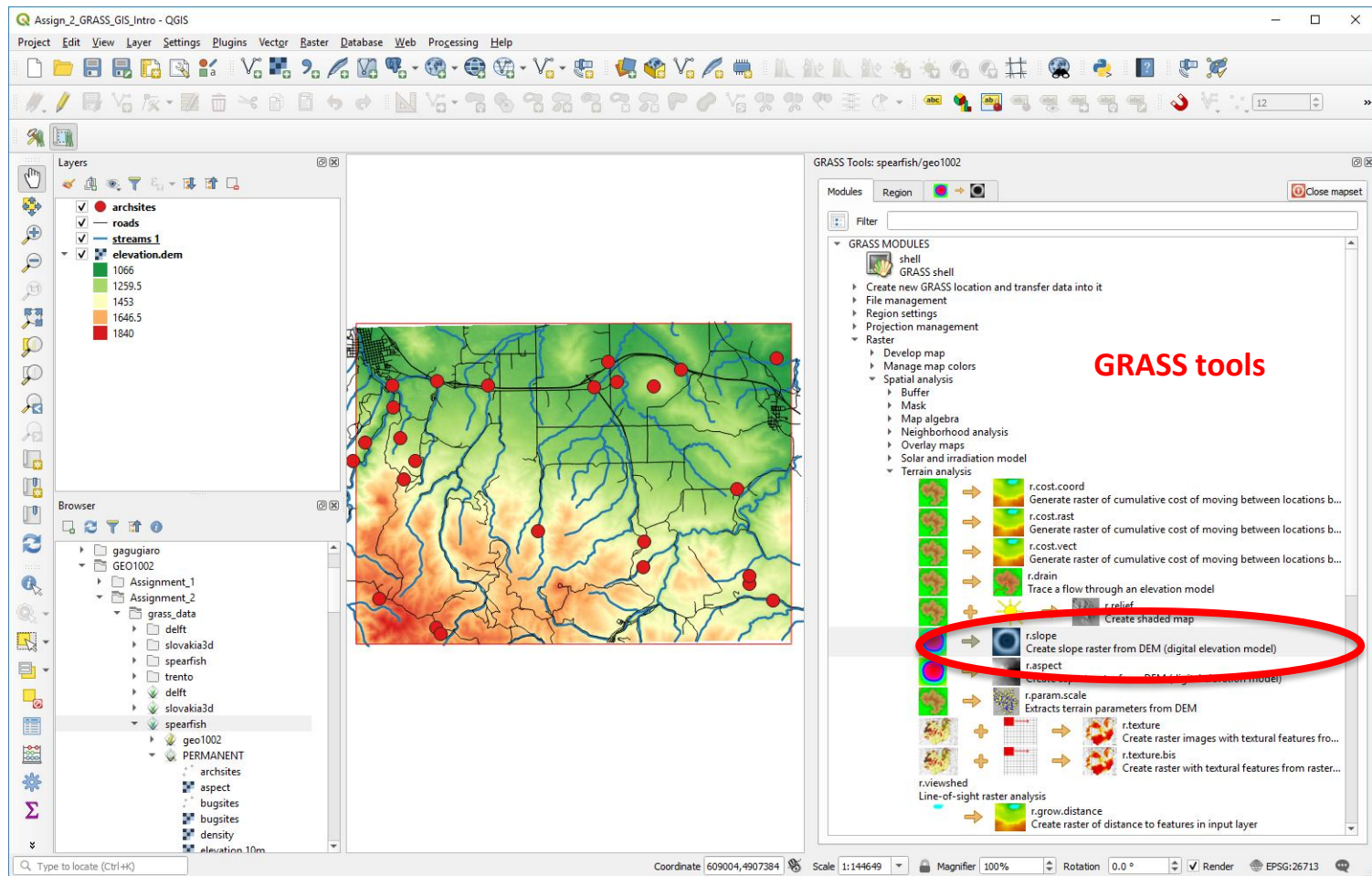
Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



Layer Browser

Working with GRASS: From QGIS

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help



The screenshot shows the QGIS interface with the GRASS Tools panel open. The map displays a terrain elevation model with a network of roads and streams. The GRASS Tools panel is titled "GRASS Tools: spearfish/geo1002" and lists various modules under the "GRASS MODULES" section. The "Create shaded map" module is highlighted with a red circle.

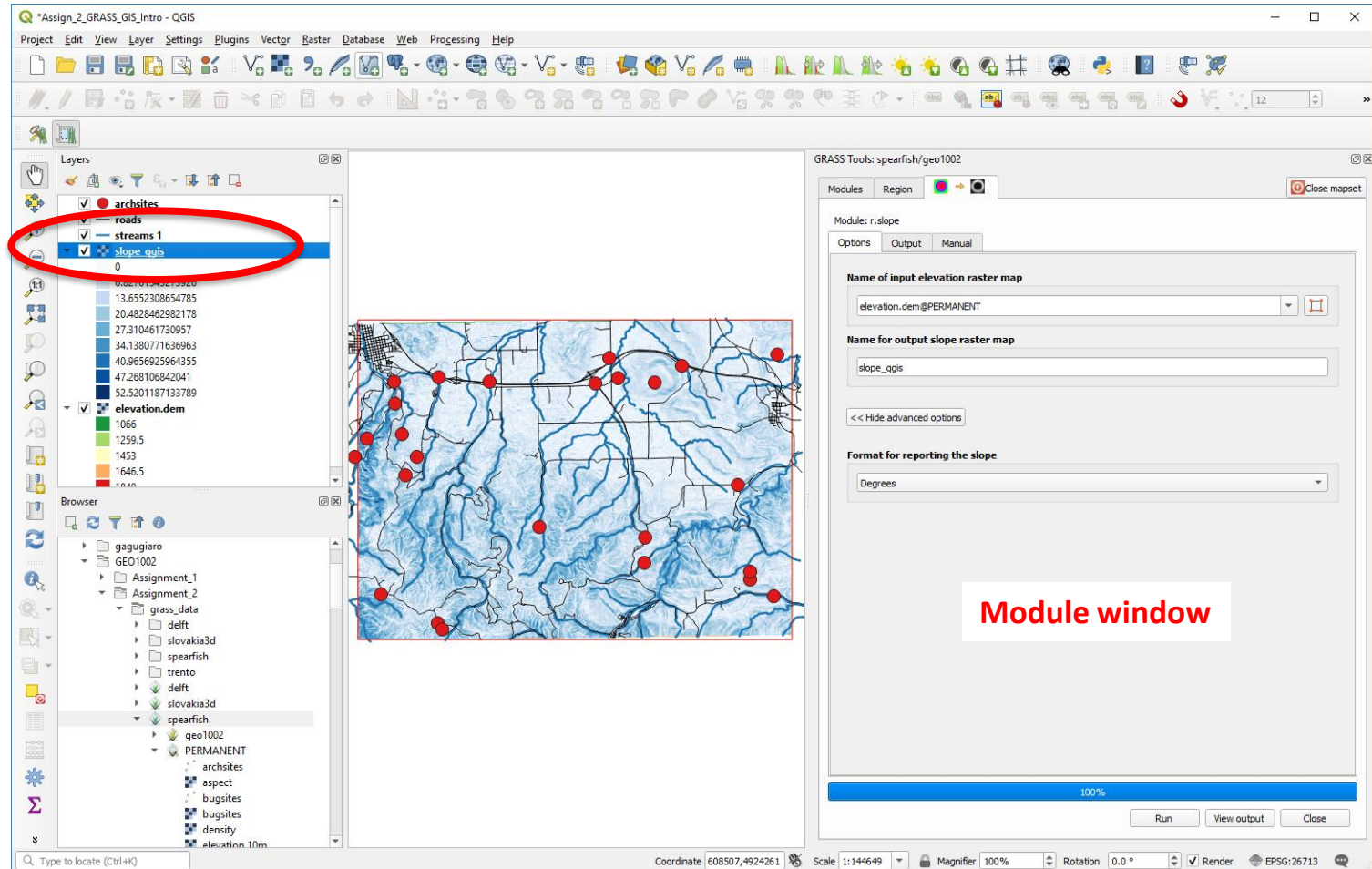
GRASS tools

- shell
 - GRASS shell
 - Create new GRASS location and transfer data into it
 - File management
 - Region settings
 - Projection management
- Raster
 - Develop map
 - Manage map colors
 - Spatial analysis
 - Buffer
 - Mask
 - Map algebra
 - Neighborhood analysis
 - Overlay maps
 - Solar and irradiation model
 - Terrain analysis
 - r.cost.coord: Generate raster of cumulative cost of moving between locations b...
 - r.cost.rast: Generate raster of cumulative cost of moving between locations b...
 - r.cost.vect: Generate raster of cumulative cost of moving between locations b...
 - r.drain: Trace a flow through an elevation model
 - r.relief: Create shaded map
 - r.slope: Create slope raster from DEM (digital elevation model)
 - r.aspect: Create aspect raster from DEM (digital elevation model)
 - r.param.scale: Extracts terrain parameters from DEM
 - r.texture: Create raster images with textural features from...
 - r.texture.bis: Create raster with textural features from raster...

The map shows a terrain elevation model with a network of roads and streams. The GRASS Tools panel is open, showing the "GRASS MODULES" section. The "Create shaded map" module is highlighted with a red circle.

Working with GRASS: From QGIS

Intro
 GRASS modules
 GRASS DB
 Running GRASS
 New setup
Running modules
 Test data
 Help

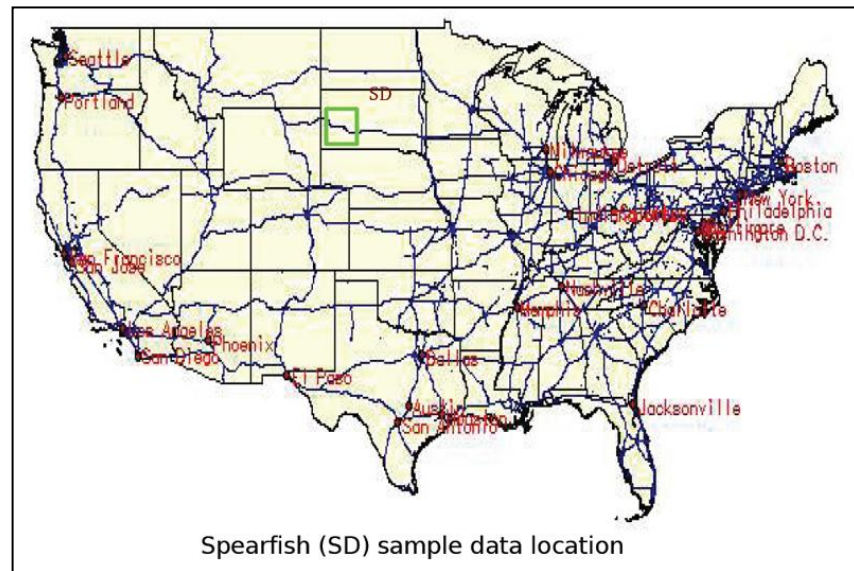


Module window

GRASS test data

There are many sample datasets to learn GRASS, to “play around” and get used to how the program works. For example:

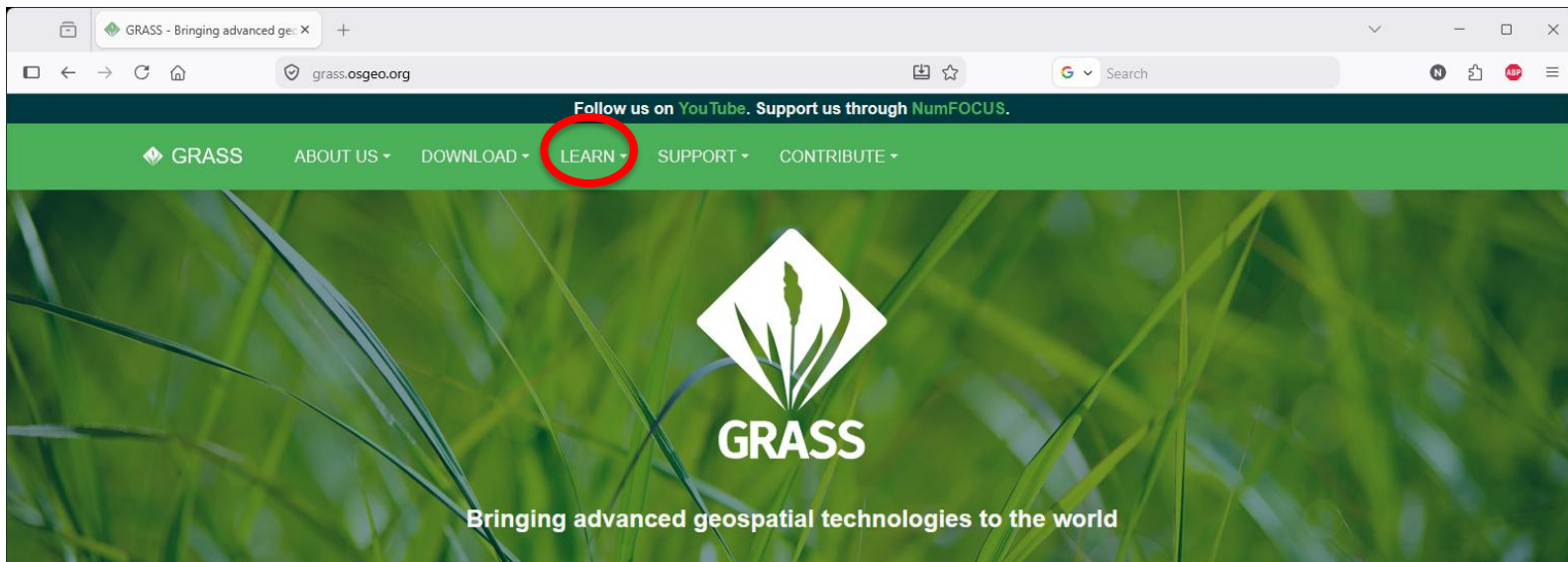
- **North Carolina**
 - LOTS of data in all formats
- **Spearfish (South Dakota)**
 - Contains raster e vector data
- **Spearfish imagery**
 - contains satellite imagery
- **Slovakia3d**
 - contains grid3d data (voxels)



Download from <https://grass.osgeo.org/download/data/>

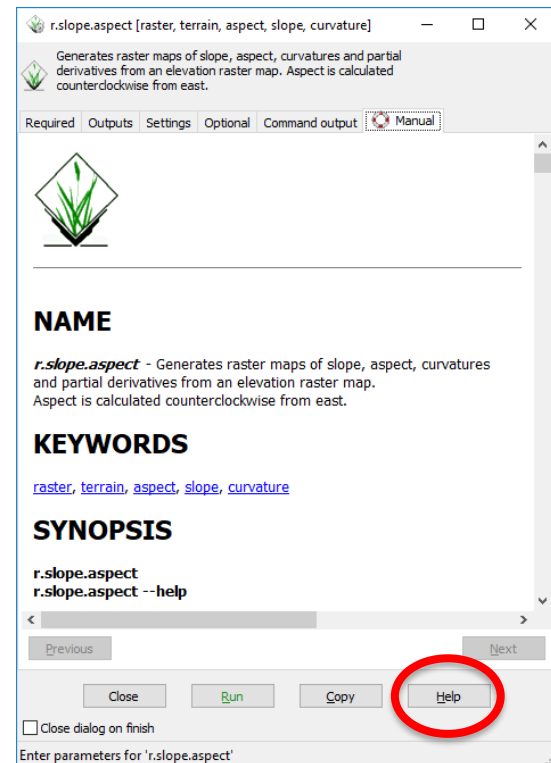
Help with GRASS

- GRASS **manual**: <https://grass.osgeo.org/learn/manuals/>
- GRASS **wiki**: <https://grasswiki.osgeo.org/wiki/GRASS-Wiki>
- GRASS **tutorials**: <https://grass.osgeo.org/learn/tutorials/>



Help with GRASS modules

- Run the module name with "--help" or "--h" in the Console window or in the Command shell
 - Example: `r.slope.aspect --help`
- From the module window, click on help button
- Refer to the GRASS online manual, where you find help for each module (see next slide)



Help with GRASS modules

Intro
GRASS modules
GRASS DB
Running GRASS
New setup
Running modules
Test data
Help

Quick Introduction

- [How to start with GRASS GIS](#)
- [Index of topics and keywords](#)
- [Intro: projections and spatial transformations](#)
- [FAQ - Frequently Asked Questions \(Wiki\)](#)
- [Graphical index of functionality](#)

Graphical User Interface

- [wxGUI](#)
- [wxGUI components](#)
- [wxGUI toolboxes](#)
- [GUI commands](#)

Display

- [Display commands manual](#)
- [Display drivers](#)

General

- [GRASS GIS startup manual](#)
- [General commands manual](#)

Addons

- [Addons manual pages](#)

Programmer's Manual

- [Programmer's Manual](#)

Raster processing

- [Intro: 2D raster map processing](#)
- [Raster commands manual](#)

3D raster processing

- [Intro: 3D raster map \(voxel\) processing](#)
- [3D raster \(voxel\) commands manual](#)

Image processing

- [Intro: image processing](#)
- [Imagery commands manual](#)

Vector processing

- [Intro: vector map processing and network analysis](#)
- [Vector commands manual](#)
- [GRASS ASCII vector format specification](#)

Database

- [Intro: database management](#)
- [SQL support in GRASS GIS](#)
- [Database commands manual](#)

Temporal processing

- [Intro: temporal data processing](#)
- [Temporal commands manual](#)

Cartography

- [PostScript commands manual](#)
- [wxGUI Cartographic Composer](#)

Miscellaneous & Variables

- [Miscellaneous commands manual](#)
- [GRASS variables and environment variables](#)

Python

- [GRASS GIS Python library documentation](#)
- [PyGRASS documentation](#)
- [GRASS GIS in Jupyter Notebooks](#)



Dr. Giorgio Agugiaro

g.agugiaro@tudelft.nl

3D Geoinformation Group

TU Delft

The Netherlands

<https://3d.bk.tudelft.nl/gagugiaro>