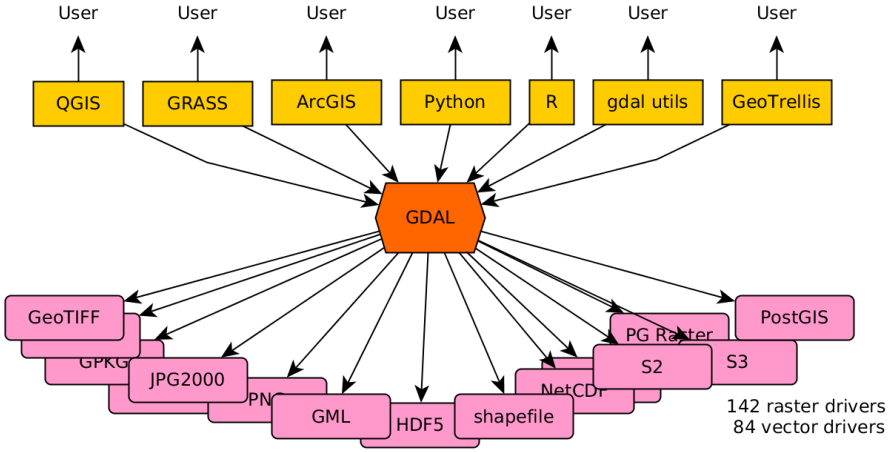
```
825  #' @keywords spatial interface
826  #' @export
827  rsaga.geoprocessor = function(
828      lib, module = NULL, param = list(),
829      show.output.on.console = TRUE, invisible = TRUE, intern = TRUE,
830      prefix = NULL, flags = ifelse(show.output.on.console,"q","s"), cores,
831      env = rsaga.env(), display.command = FALSE, reduce.intern = TRUE,
```

Integrating R with GIS for innovative geocomputing – the examples of RQGIS and RSAGA

Jannes Muenchow, Patrick Schratz,
Donovan Bangs, Alexander Brenning

R as a GIS

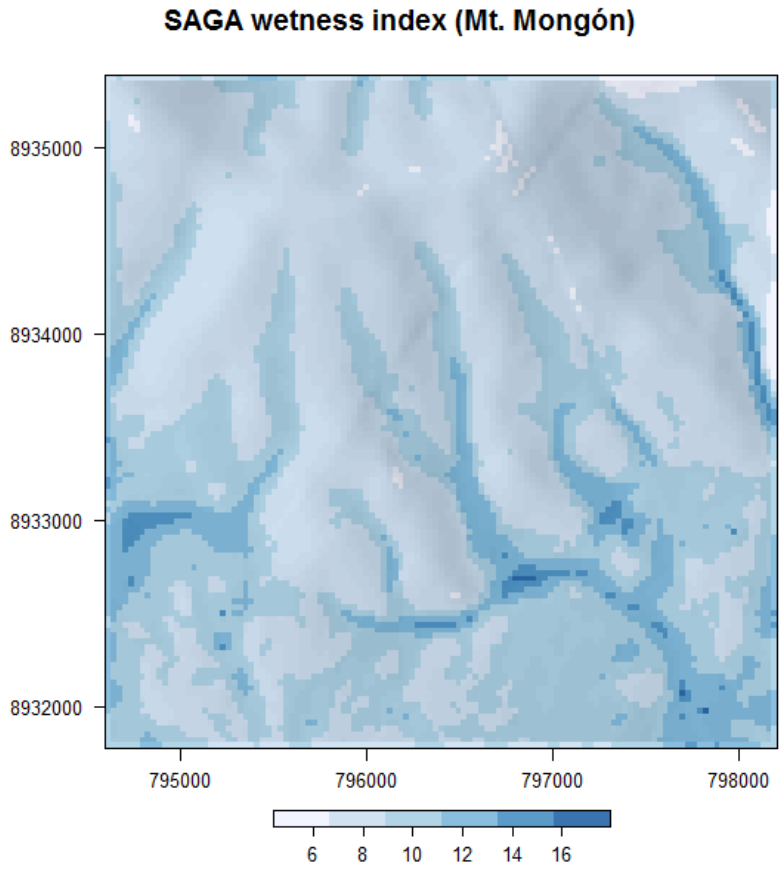
- More than 100 geo-related R packages (https://cran.r-project.org/web/views/Spatial.html)



http://r-spatial.org//2016/11/29/openeo.html

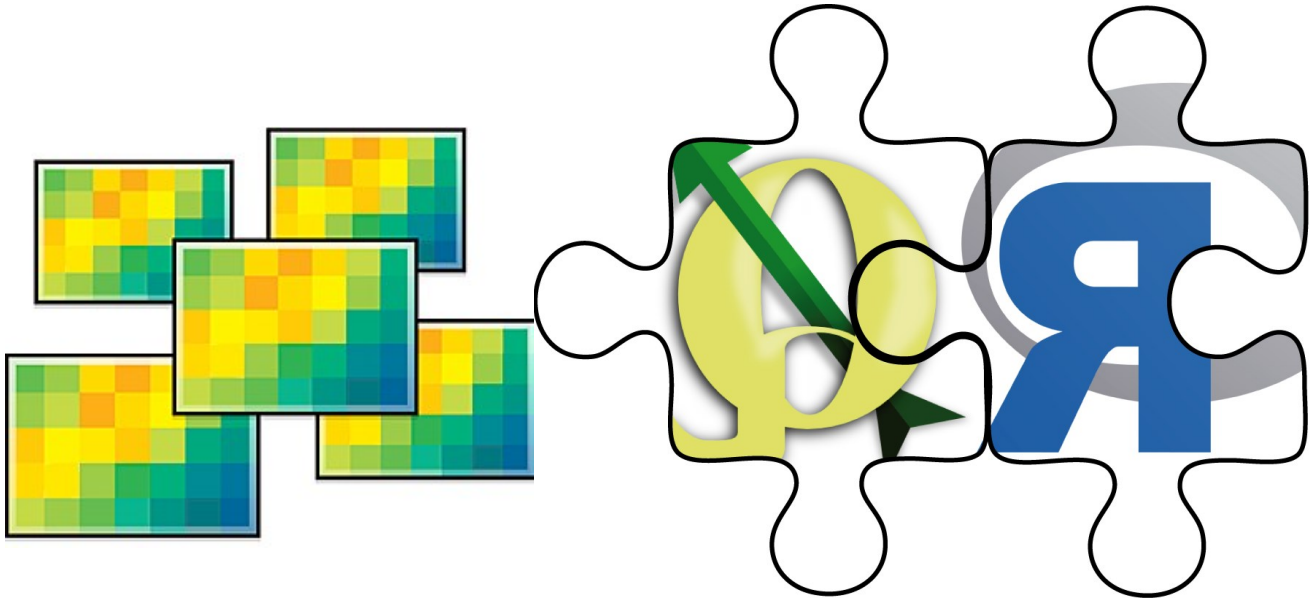
But:

- R was never designed as a GIS
- Computationally demanding operations
- Missing geoalgorithms



Combining the best of two worlds

seit 1558



http://www.esri.com/

https://7segments.com/

Navigation

1. R as a GIS

2. R-GIS integration

3. R-GIS examples

4. RQGIS vs. RSAGA

(5. RQGIS usage)

RSAGA

RQGIS

Navigation
Click buttons to jump to content

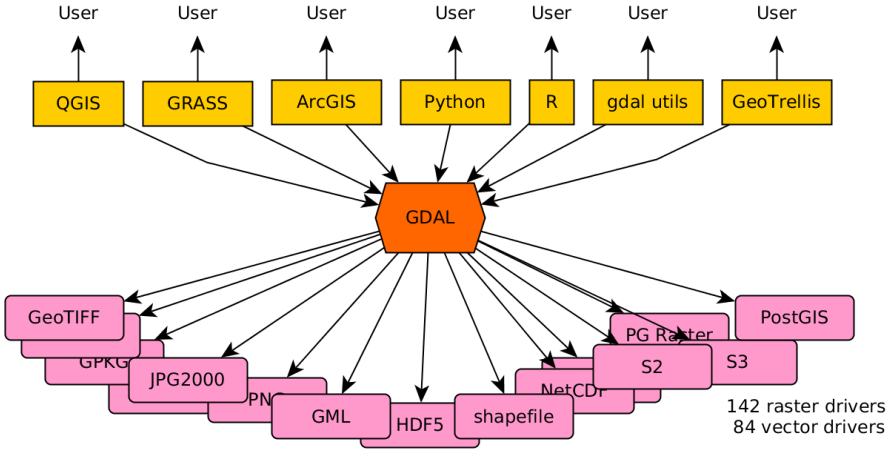
seit 1558

R AS A GIS

Jump to Navigation

R as a GIS

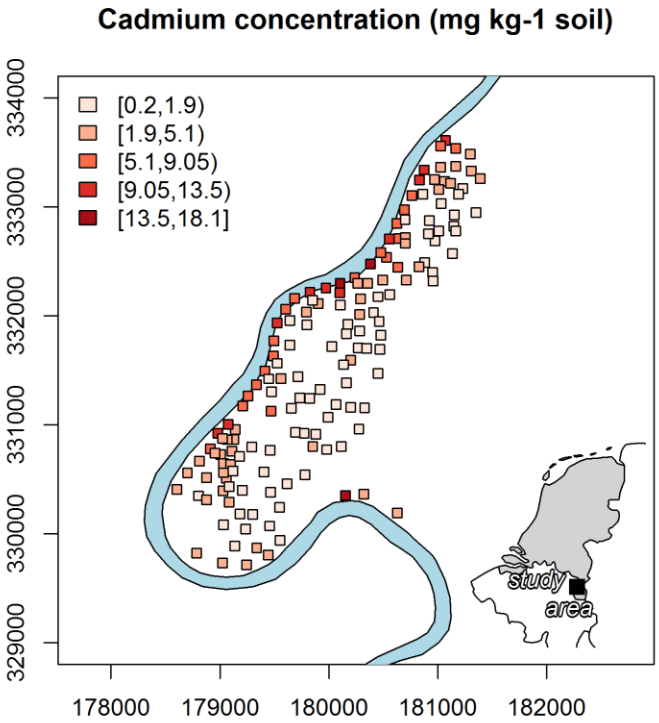
- More than 100 geo-related R packages (https://cran.r-project.org/web/views/Spatial.html)
- Package **`rgdal`** for importing and exporting geodata



http://r-spatial.org//2016/11/29/openeo.html

R as a GIS

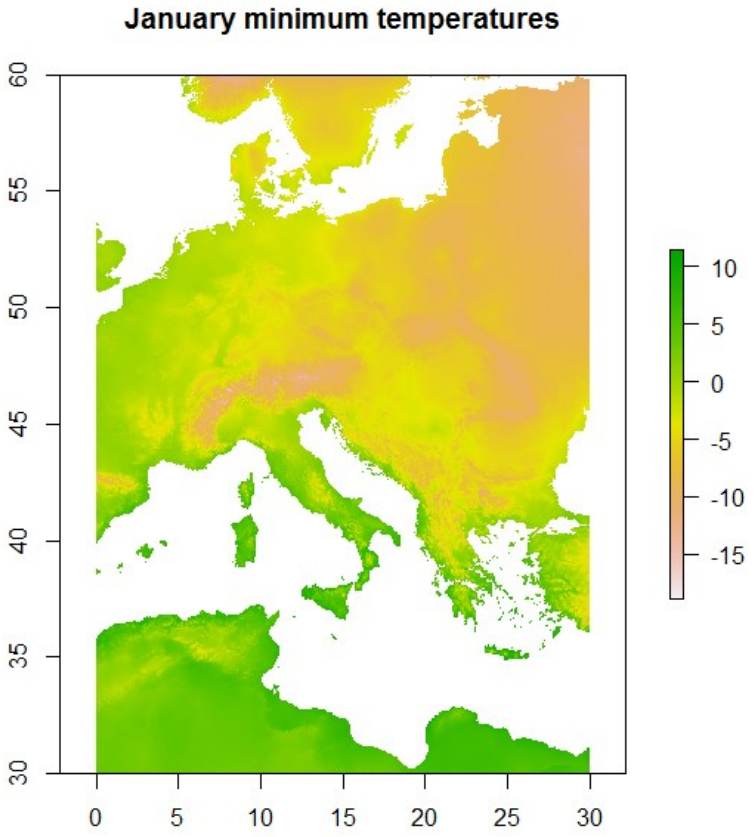
- More than 100 geo-related R packages (https://cran.r-project.org/web/views/Spatial.html)
- Package **`rgdal`** for importing and exporting geodata
- Packages **`sp`** and **`sf`** for vector geodata



Data: Rikken, M.G.J & Van Rijn, R.P.G. (1993).

R as a GIS

- More than 100 geo-related R packages (https://cran.r-project.org/web/views/Spatial.html)
- Package **`rgdal`** for importing and exporting geodata
- Packages **`sp`** and **`sf`** for vector geodata
- Package **`raster`** for raster geodata



Data: http://www.worldclim.org/.

R as a GIS

Defining a GIS as a system for the analysis, manipulation and visualization of geographical data (Longley, Goodchild, Maguire, and Rhind 2011), one could argue that R has become a GIS

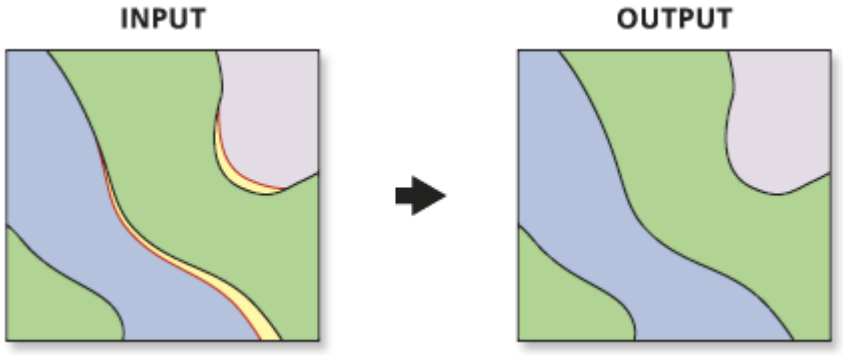
But what about…

seit 1558

(digitizing)

(Geodatabase-functionality and topology rules)

http://www.unioneag.org

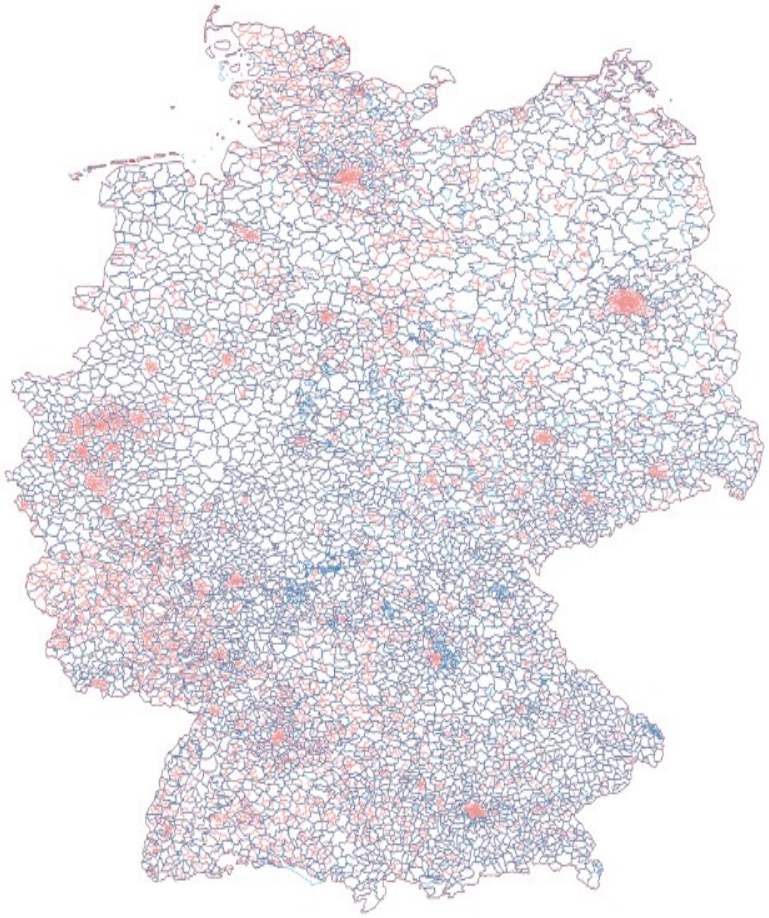


http://help.arcgis.com/

Computationally demanding operations

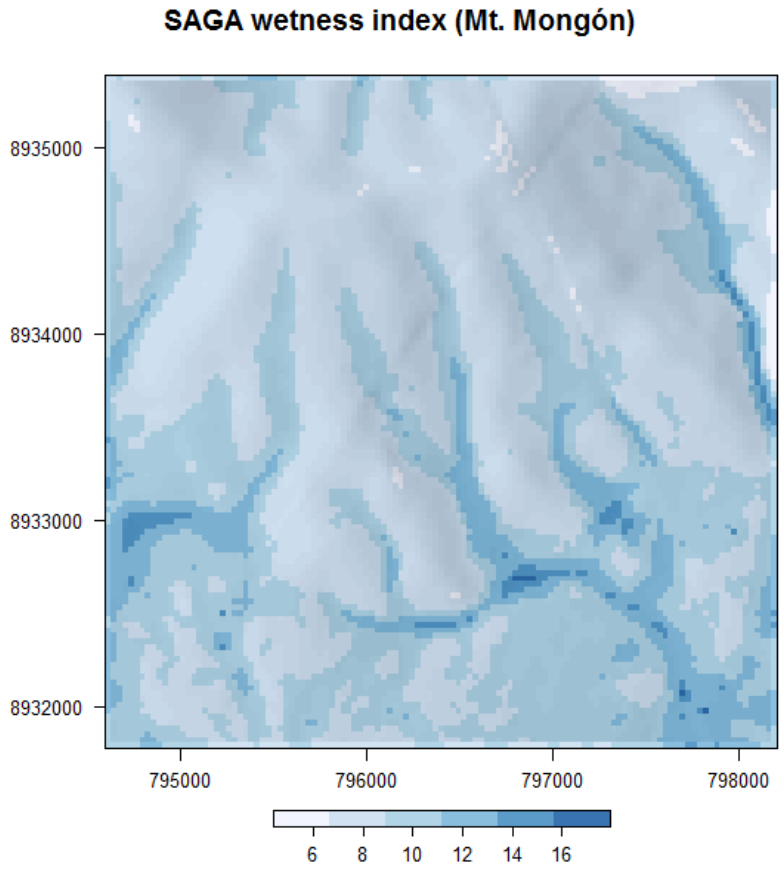
seit 1558

- Computationally demanding operations



Missing geoalgorithms

- Catchment area
- Catchment slope
- Saga Wetness Index
- Lidar processing
- …



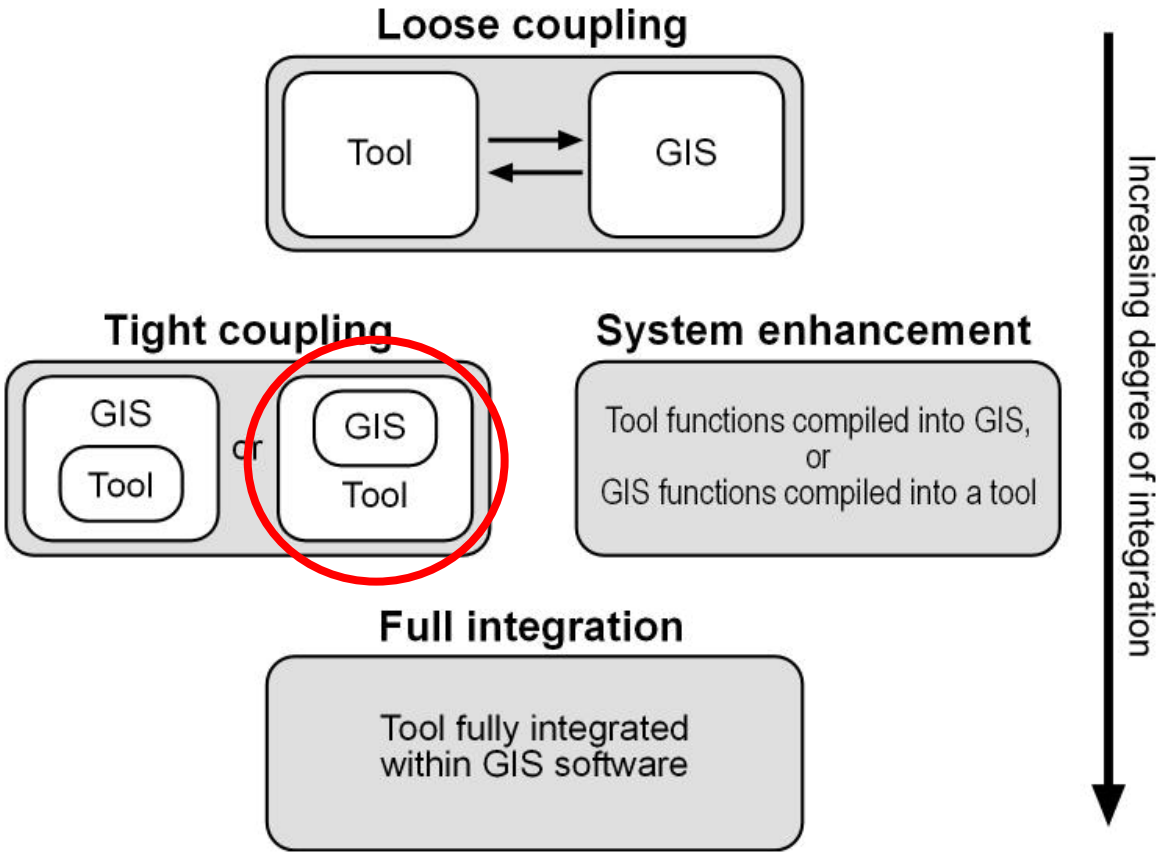
Interface

R has been designed from the beginning as an interactive interface to other software packages (Chambers, 2016).

R-GIS INTEGRATION - RQGIS AND RSAGA

Jump to Navigation

GIS interfaces



http://www.geocomputation.org/2000/GC009/Gc009.htm

R-GIS integration

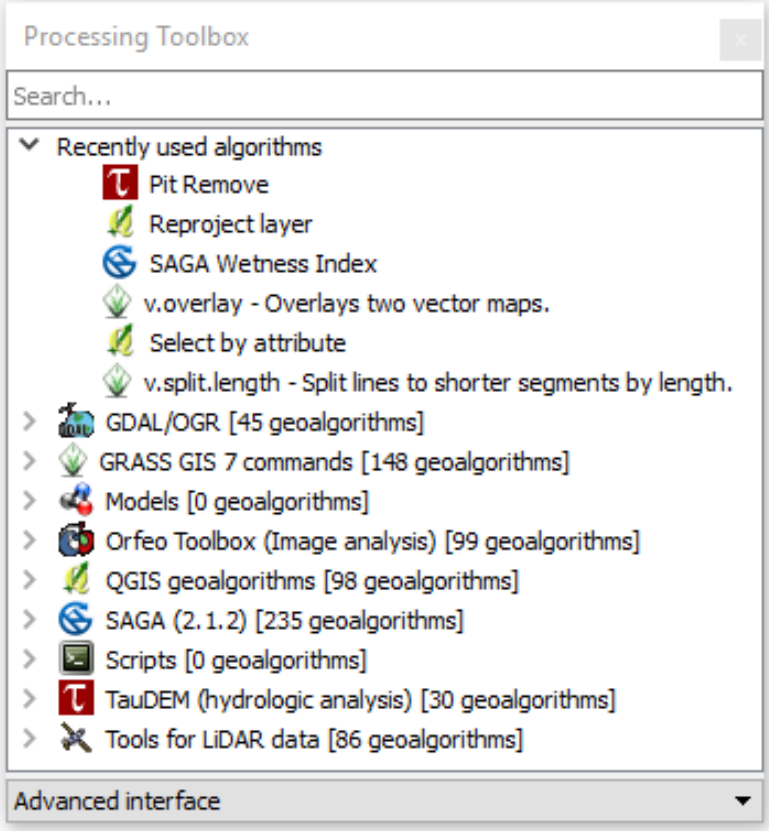
RSAGA

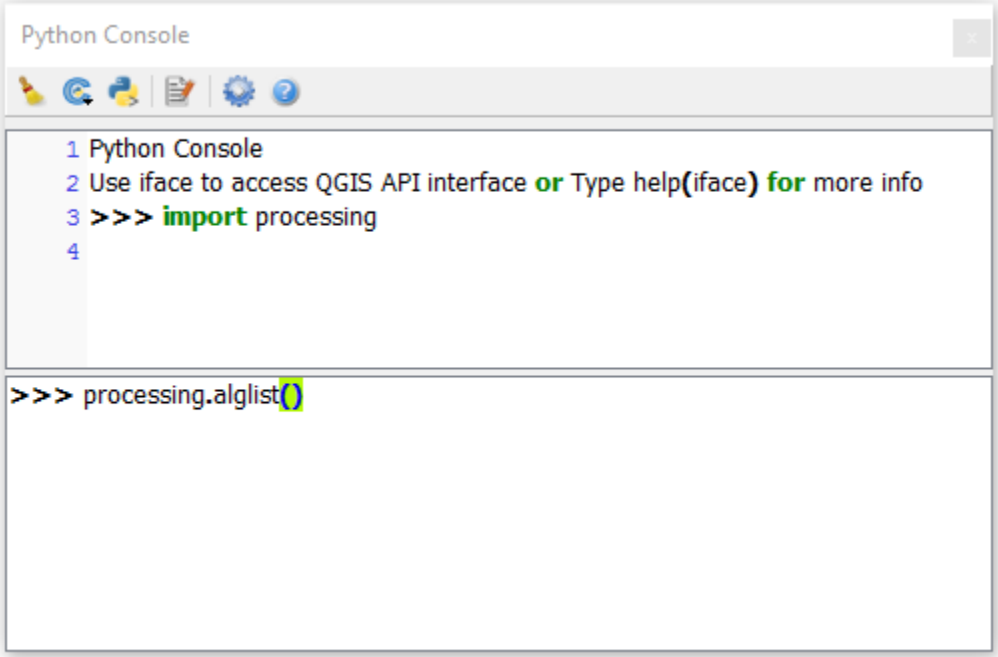
RQGIS

rgrass7

RPyGeo

QGIS – Python API



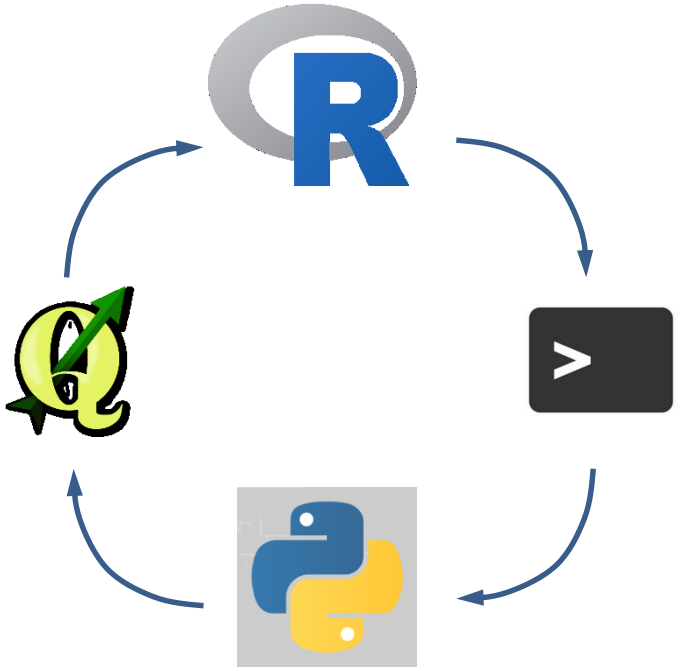


How does it work?

seit 1558

- Access QGIS Python API via command line (old interface)
- The new interface (still work in progress) uses the reticulate package to establish a QGIS Python-tunnel, please refer to:

  https://github.com/jannes-m/RQGIS/tree/rPython

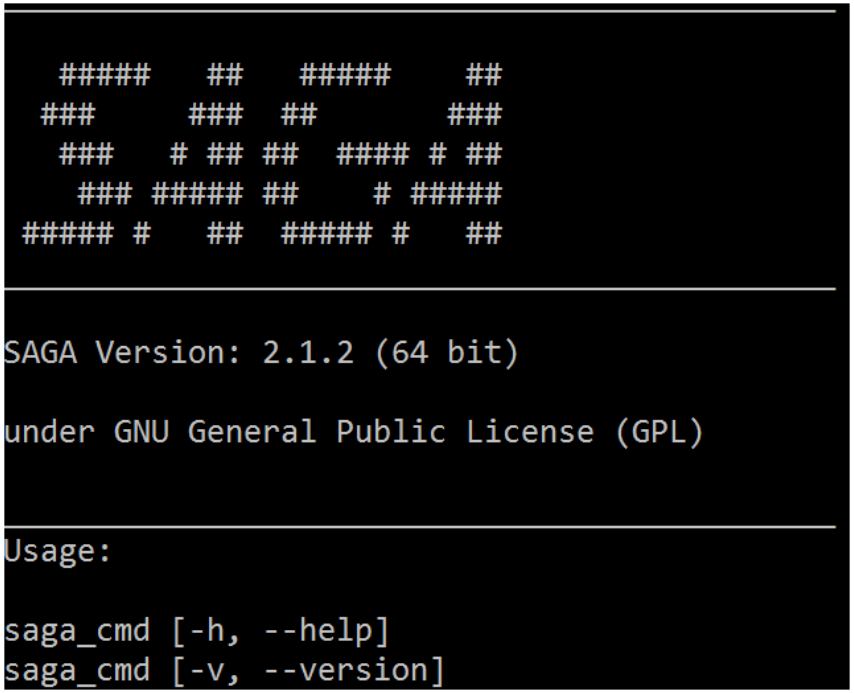


Most notable features of RQGIS

- QGIS geoalgorithms
- Access to hundreds of further geoalgorithms, especially SAGA- and GRASS-geoalgorithms
- R users can stay in their environment without having to touch Python
- Convenience functions `open_help`, `get_args_man` and `run_qgis`

RSAGA interface

- The RSAGA package provides R geocomputing functions that make use of the command line interface of SAGA GIS, `saga_cmd.exe`, to execute SAGA GIS modules.



RSAGA structure

Geoprocessing environment

- List data structure with information on working directory, location of SAGA GIS binaries, etc.

Geoprocessor (using SAGA GIS)

- Workhorse that calls SAGA GIS and provides low-level access to all SAGA GIS modules

User-level interface functions (using SAGA GIS):

- e.g., `rsaga.local.morphometry`, `rsaga.hillshade`

Local and focal functions (written in R):

- e.g., `multi.focal.function`, `grid.predict`

Utility functions (written in R):

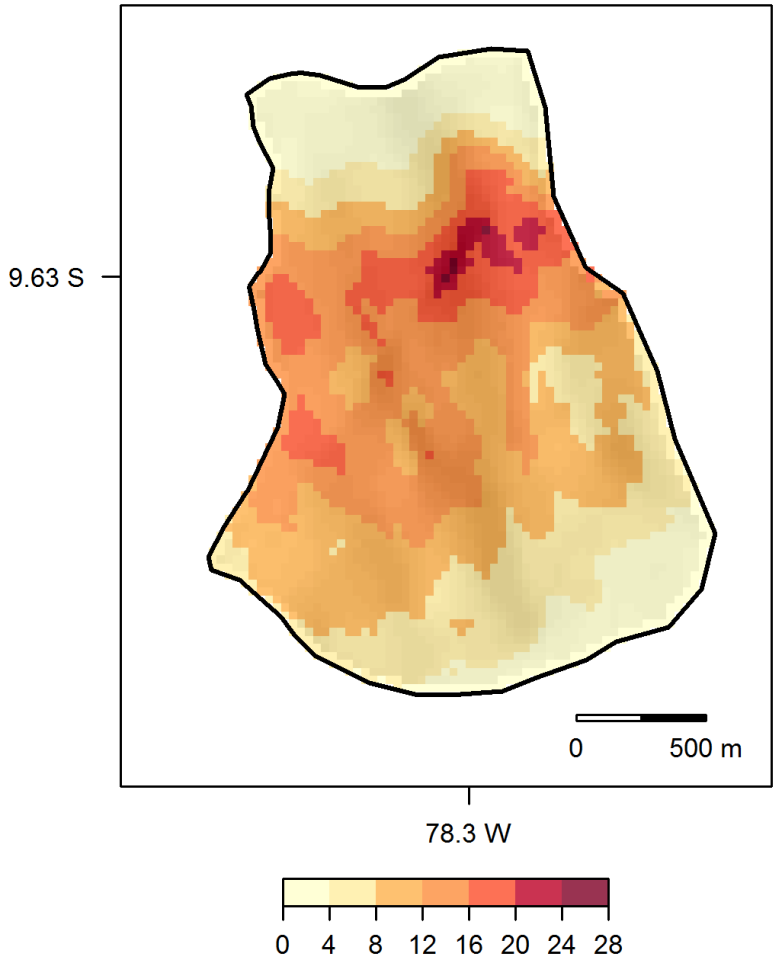
- e.g., `pick.from.ascii.grid`

R-GIS EXAMPLES

Jump to Navigation

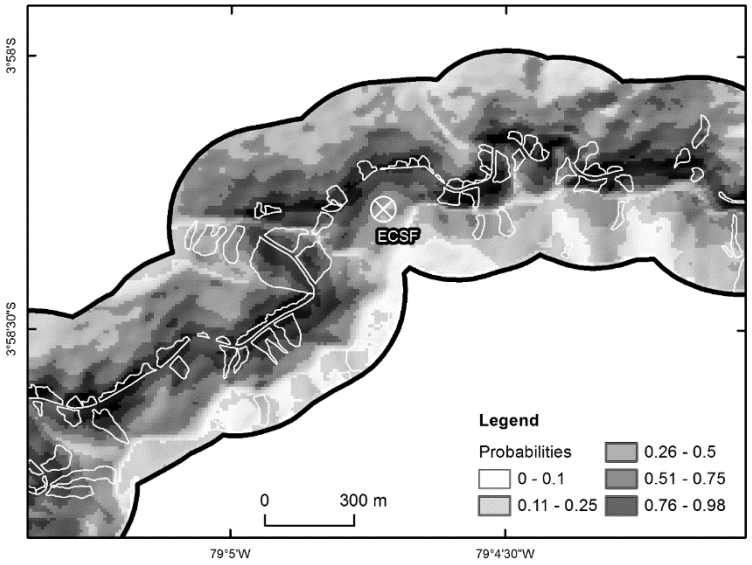
Spatial prediction of alpha diversity

seit 1558



Muenchow et al. (2013): Predictive mapping of species richness and plant species' distributions.

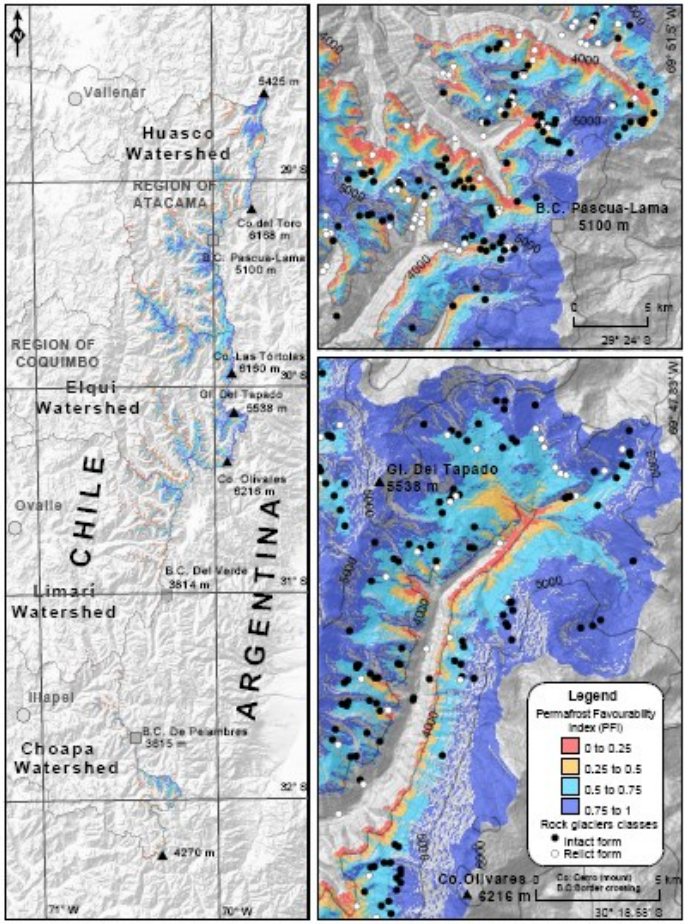
Landslide susceptibility



Brenning et al. (2015): Landslide susceptibility near highways.

Rock glaciers/permafrost

- Computation of direct and diffuse incoming solar radiation using RSAGA

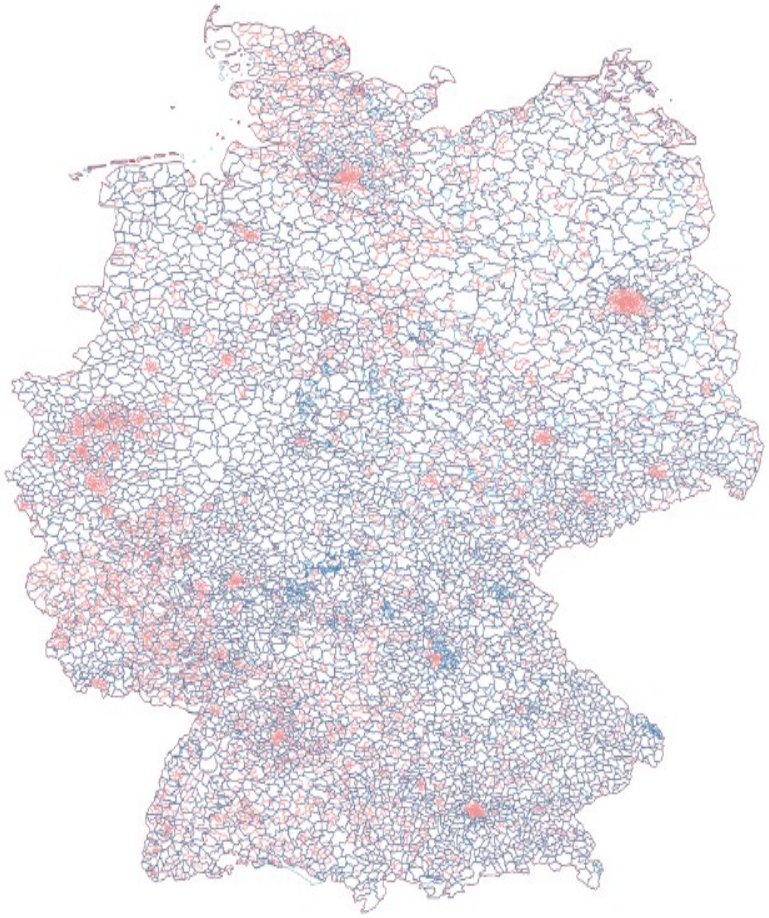


Azócar et al. (2017): Permafrost distribution modeling.

Geomarketing

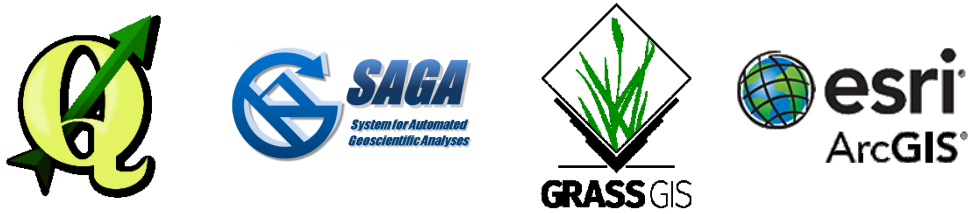
seit 1558

- Unioning postal code with municipality layers



Further applications

- Soil classes and mapping (e.g., Brungard et al. 2015)
- Stream networks (e.g., Hengl et al. 2010)
- Climatology (rainfall prediction; e.g., Hengl et al. 2010)
- Archeology (e.g., Borck 2016)
- Socio-demography(population index prediction; e.g., Bajat et al. 2012)
- …



COMPARING RQGIS AND RSAGA

Jump to Navigation

RQGIS vs. RSAGA/rgrass7

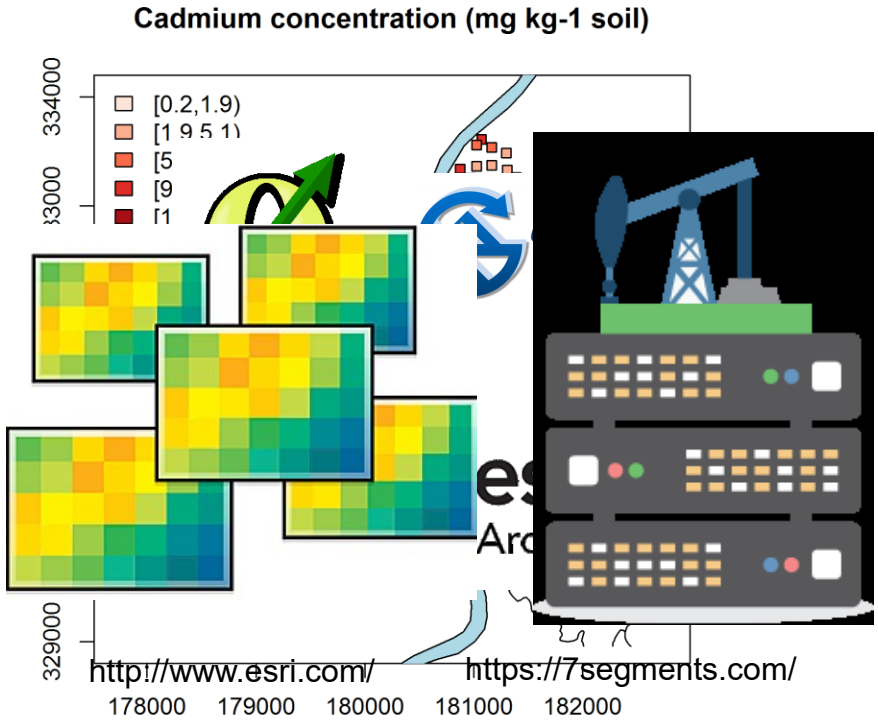
- Unified interface to SAGA and GRASS
- Interface to further 3rd-party providers
- Easier-to-use
  - `get_args_man`, `open_help`
  - On-the-fly import/export of spatial objects (`run_qgis`)
  - Automatic data conversions (e.g., asc, tif, etc.)

But:

- QGIS does not support the most recent SAGA versions
- QGIS does not provide access to all SAGA and GRASS functionalities
- RSAGA has special geocomputing functions (written in R)

Wrap-up

- We can use R as a GIS
- Geoprocessing is better done in a GIS
- R-GIS integration combines the best of two worlds
- RQGIS, RSAGA, rgrass7 are all great



Data: Rikken, M.G.J & Van Rijn, R.P.G. (1993).

RQGIS USAGE

Jump to Navigation

RQGIS on github

Find out how to use RQGIS on github:

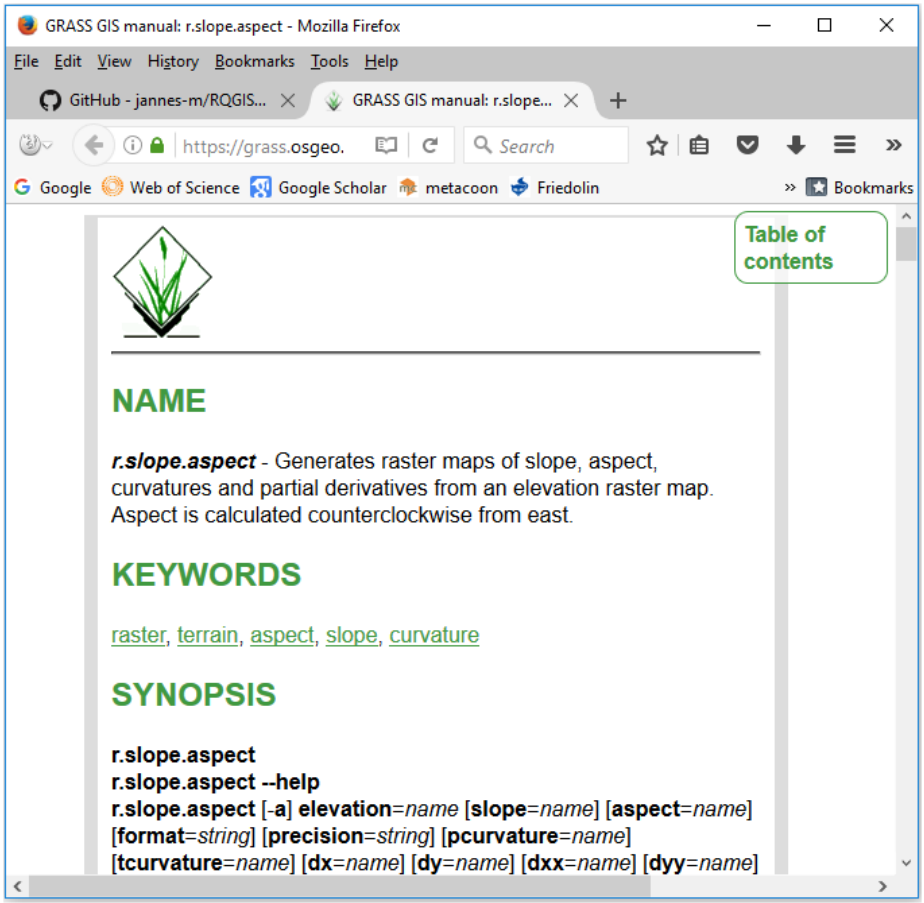
http://jannes-m.github.io/RQGIS/index.html

Access to the online help

seit 1558

Let‘s run:

```
library("RQGIS")
qgis_env <-
set_env("C:/OSGeo4W64/")
open_help(alg =
"grass7:r.slope.aspect",
 qgis_env = qgis_env)
```



Automatic argument retrieval

The QGIS Python API does not provide a way to collect function arguments of a geoalgorithm. You can only do that:

```
processing.alghelp("grass7:r.slope.aspect")
```

```
ALGORITHM: r.slope.aspect - Generates raster layers of slope, aspect,
curvatures and partial derivatives from a elevation raster layer.
        elevation <ParameterRaster>
        format <ParameterSelection>
        precision <ParameterSelection>
        -a <ParameterBoolean>
        zscale <ParameterNumber>
        min_slope <ParameterNumber>
        GRASS_REGION_PARAMETER <ParameterExtent>
        GRASS_REGION_CELLSIZE_PARAMETER <ParameterNumber>
        slope <OutputRaster>
        aspect <OutputRaster>
        pcurvature <OutputRaster>
        tcurvature <OutputRaster>
        dx <OutputRaster>
        dy <OutputRaster>
        dxx <OutputRaster>
        dyy <OutputRaster>
        dxy <OutputRaster>


format(Format for reporting the slope)
        0 - degrees
        1 - percent
precision(Type of output aspect and slope layer)
        0 - FCELL
        1 - CELL
        2 - DCELL
```

Convenience function `get_args_man`

However, `get_args_man` let‘s you automatically collect all function arguments and default values of a specific geoalgorithm.

```
R> params <- get_args_man(alg = "grass7:r.slope.aspect",
R+   options = TRUE, qgis_env =  qgis_env)
R+ params

$elevation
[1] "None"

$format
[1] "0"
```

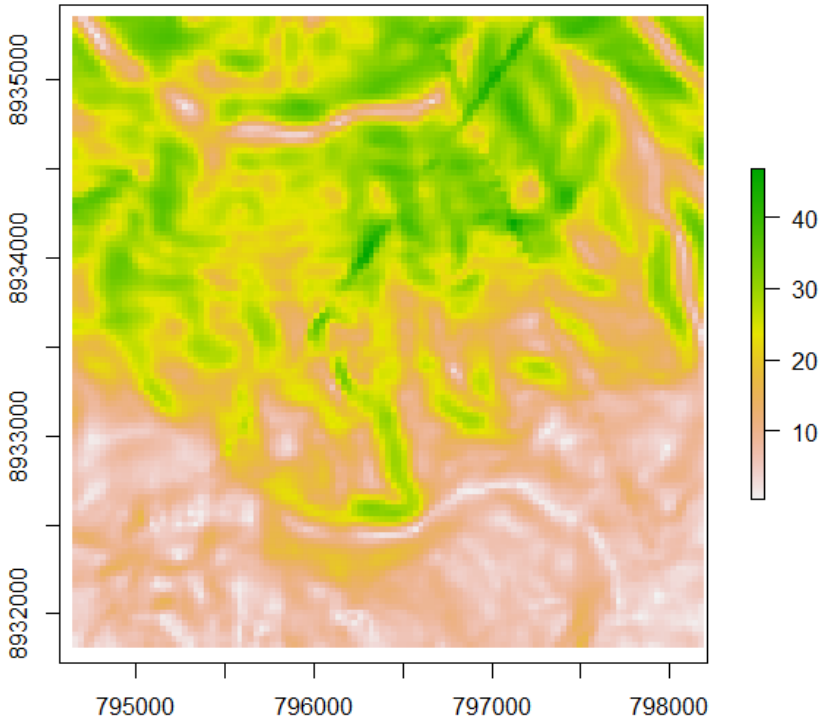
…

Let's run_qgis

Spatial object residing in R

```
data("dem")
params$elevation <- dem
params$slope <-
file.path(tempdir(),
"slope.tif")

slope <- run_qgis(
"grass7:r.slope.aspect",
params = params,
qgis_env = qgis_env,
load_output = TRUE)
```



Loads automatically the QGIS ouptut back into R