

Case Study:
Stream Network
Abstraction



Inside the Code

`r.stream.extract`

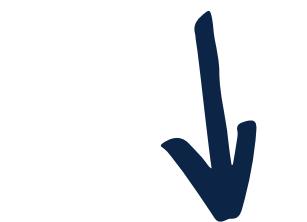


Grass library

extracts streams in both
raster and vector format
from a required
elevation map
optional
accumulation map

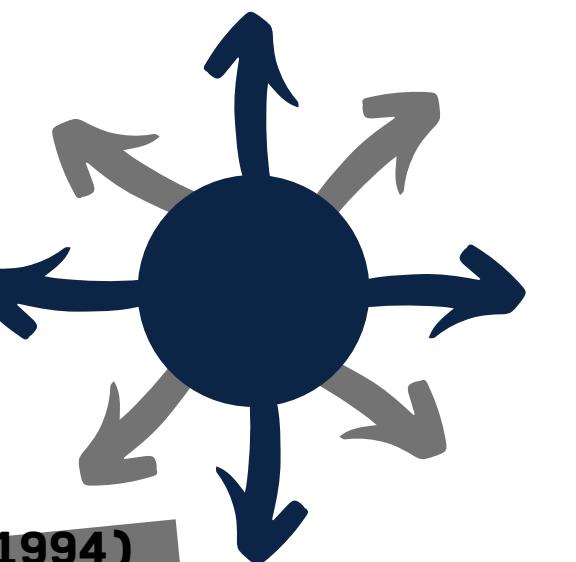
input
and
input

Elements to consider (context variables)

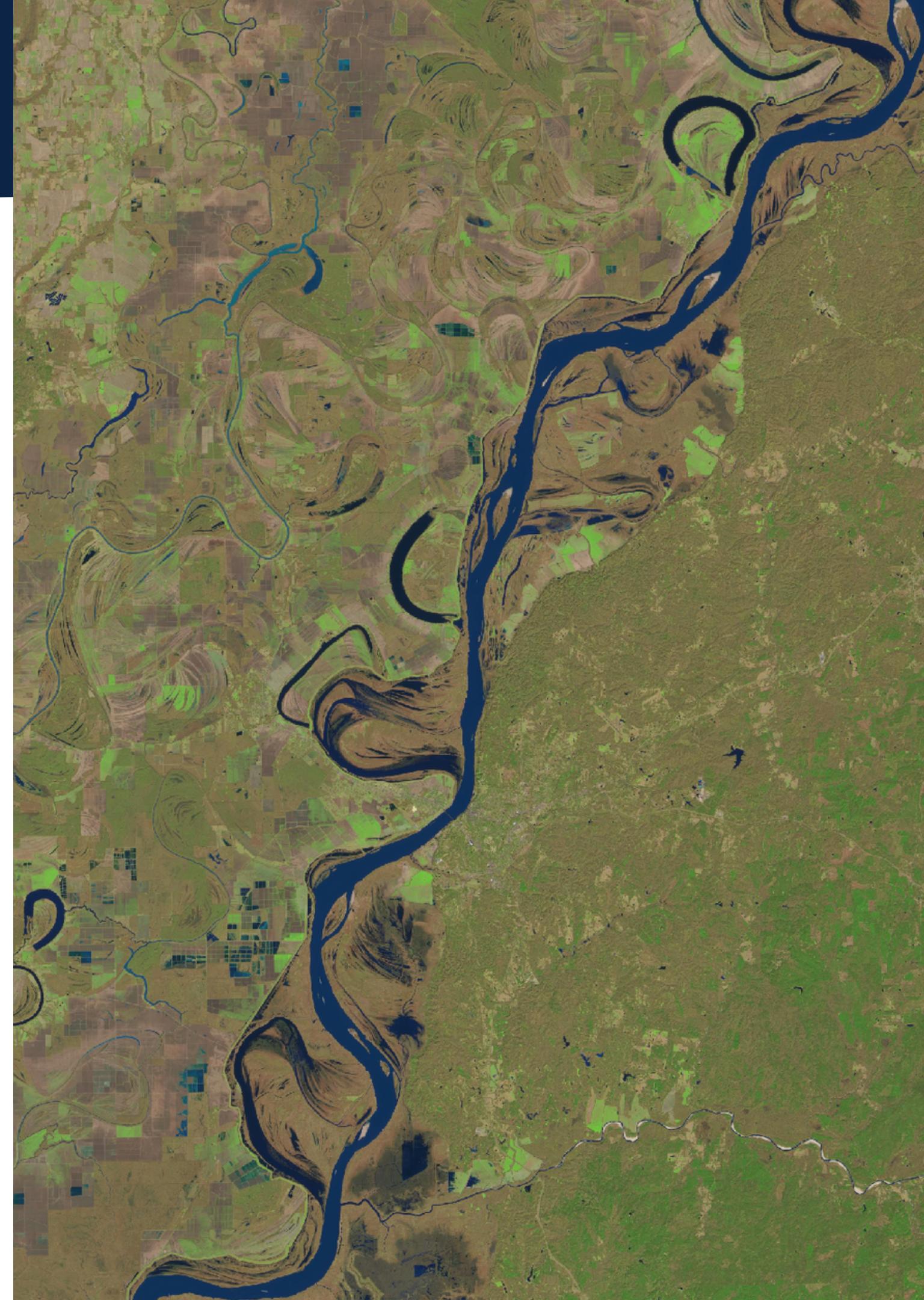


About Grass:

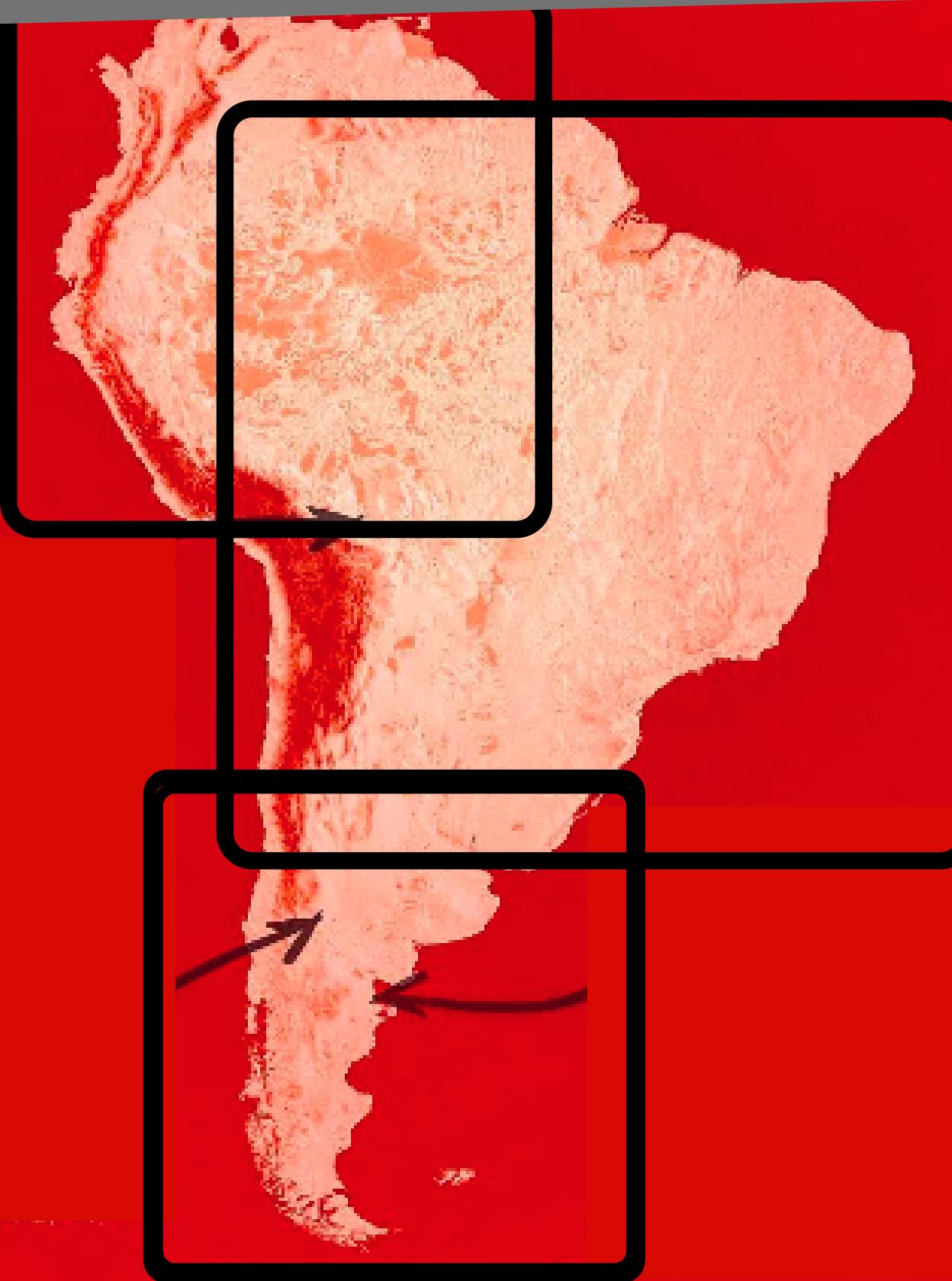
- a) Parallel running of command line and bash
- b) User interface / Command line
- c) Defined Structure Mapset, Location GisBase



Holmgren (1994)



Inside the Code



- a) **r.external**
input = DEM
output=elv --o --q # dem

This function brings the elements we need

- b) **ogrinfo|grep**
Obtains the extends of our shapefiles and stores them in variables.

- c) **crop**
Using S, N, E, W extents croping the DEM making it an easier process to run.

- d) **r.stream.extract (GRASS)**
elevation, accumulation, depression, direction stream_raste

- e) **rasterio.merge**
listing the needed elements, appending and exporting.

Main Challenges

- a) Setting up the environment for GRASS
- b) Setting up the Virtual Machine
- c) Understanding Syntax
- d) Adapting other's code sources into my case
- e) Selecting the case and understanding what's possible.

Given Insights

- a) Troubleshooting resources / RTFM
- b) The computer won't do anything unless it's told to do so (Planning)
- c) Breaking down my procedures into logical steps
- d) The importance of coding for science and replicability. (Pinning down a tought process)
- e) Absorbing knowledge as a skill that needs to be practiced.

