

Introduction to spatial mapping

Spatial Statistics

Madlene Nussbaum

m.nussbaum@uu.nl

Utrecht University, Department of Physical Geography

1 What is spatial mapping?

- Spatial mapping: predicting or visualizing how a variable of interest varies across space.
- Main goal: predict values at **unsampled locations**.
- Common in:
 - Environmental science, especially soil
 - Ecology & conservation
 - Public health & epidemiology
 - Urban & socio-economic studies

2 Ingredients for spatial mapping

1. **Point data** (outcome, response, target variable)

- Measurements or observations at known sampled locations
- Examples: soil samples, bird sightings, hospital admissions

2. Optional: **Covariates** (predictors, explanatory variables)

- Continuous spatial layers available for the whole area of interest
- Often available as gridded / spatial raster, or can be transformed into raster
- Examples: elevation, climate, land use, population density, remote sensing indices

3. **Prediction function**

- A model with a pre-defined structure or an algorithm that establishes a structure from the relationships of the response and covariates (machine learning)
- Examples: linear regression, ordinary kriging, random forest

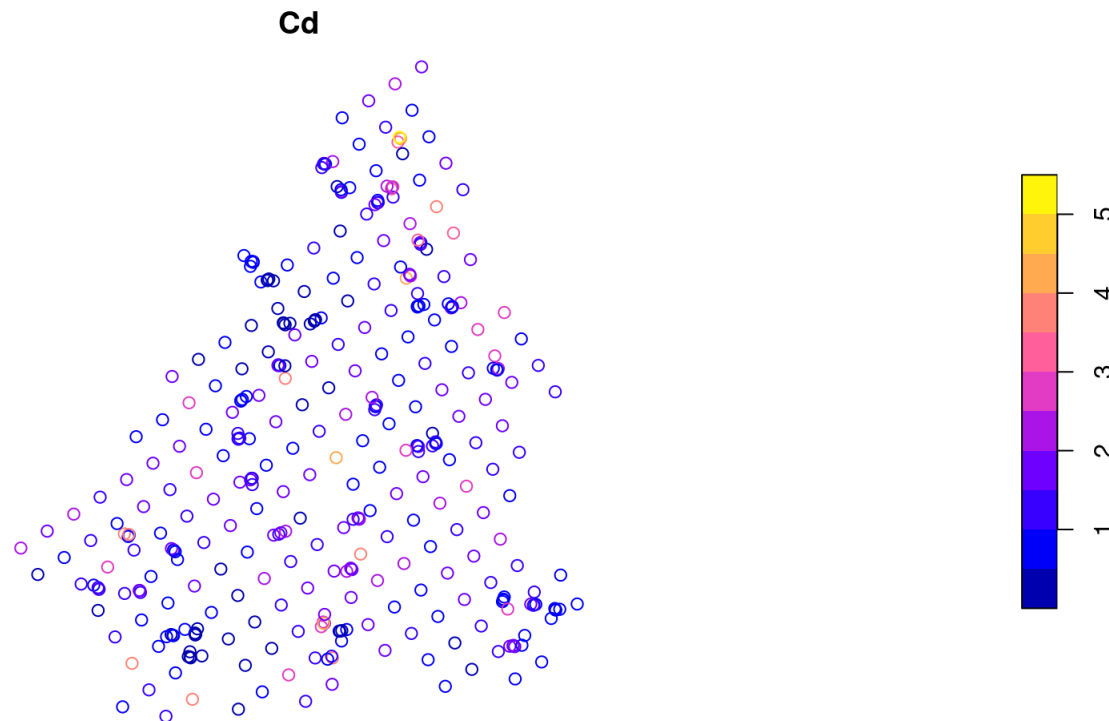
Spatial mapping

Spatial mapping links **point data** to **raster covariates** using a **prediction function**. Mapping can be done without raster covariates, then only the **neighborhood relationships** of the point data are used.

3 Examples of point data

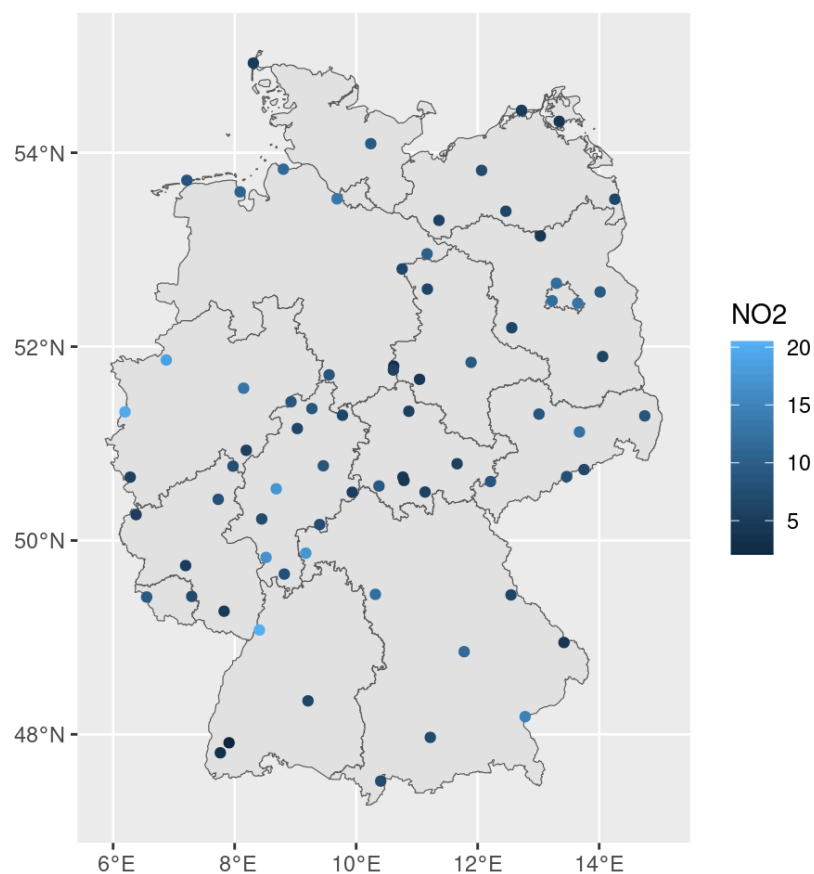
Example of geostatistical data: Soil pollution

```
1 library(sf)
2 library(compositions)
3 data(juraset)
4 d <- st_as_sf(juraset, coords = c("X", "Y"))
5 plot(d[3])
```



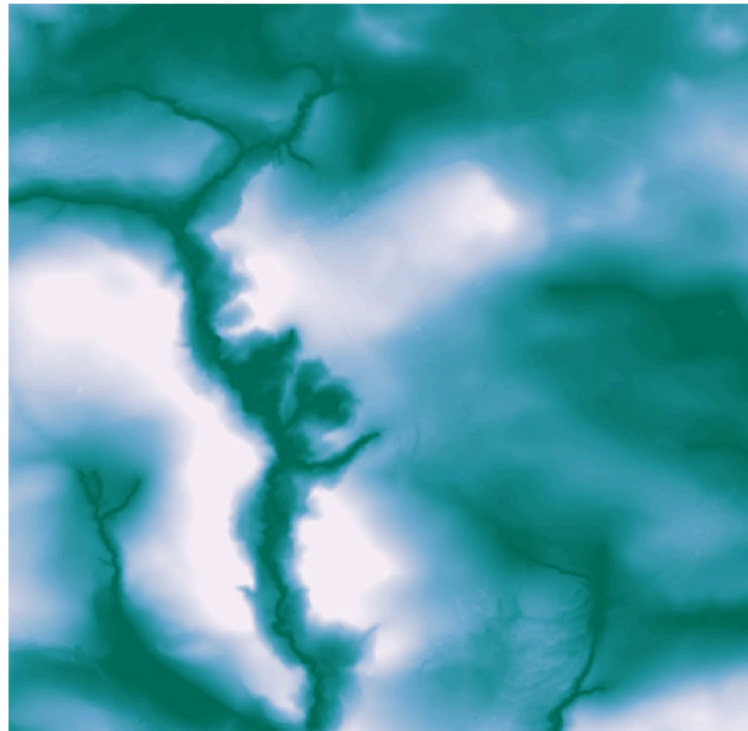
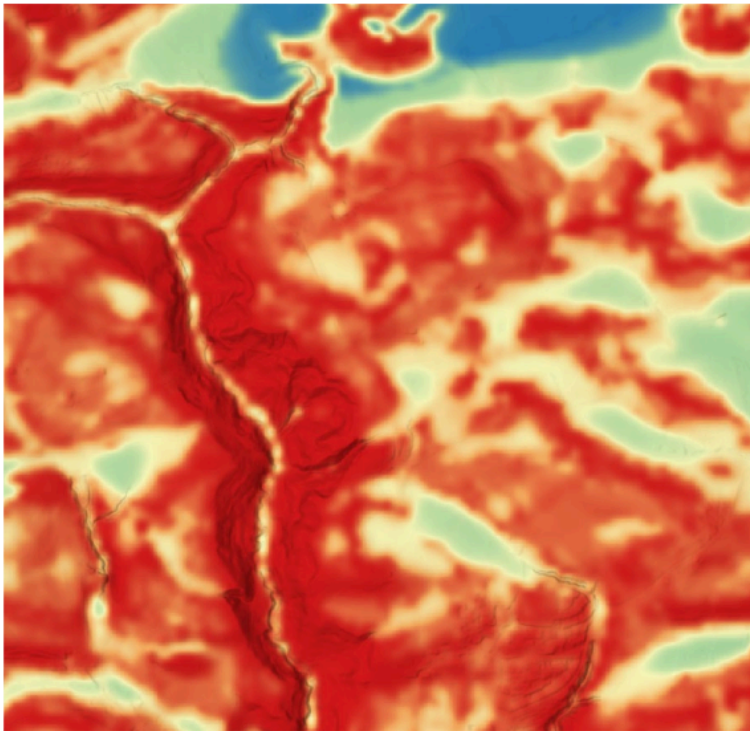
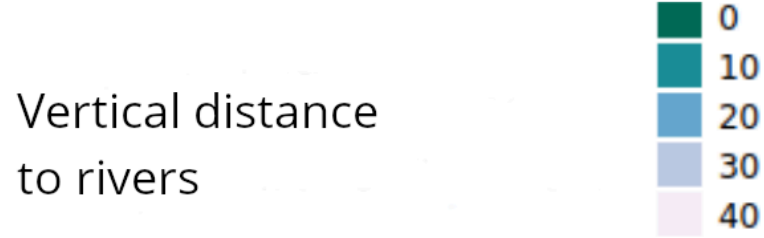
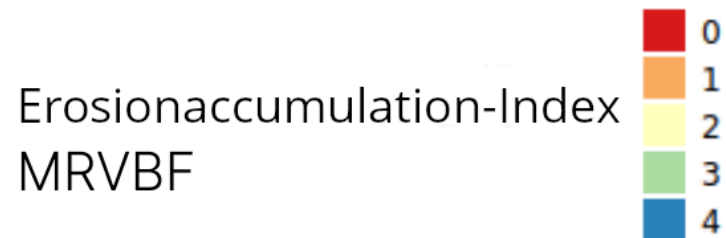
Example of geostatistical data: Air pollution

```
1 library(gstat)
2 library(ggplot2)
3 no2 <- read.csv(system.file("external/no2.csv", package = "gstat"))
4 v.no2 <- st_as_sf(no2, crs = "OGC:CRS84", coords = c("station_longitude_deg", "station_latitude_deg"))
5 v.de <- read_sf("https://github.com/edzer/sdsr/raw/main/data/de_nuts1.gpkg")
6 ggplot() + geom_sf(data = v.de) + geom_sf(data = v.no2, mapping = aes(col = NO2))
```



4 Examples of predictor data

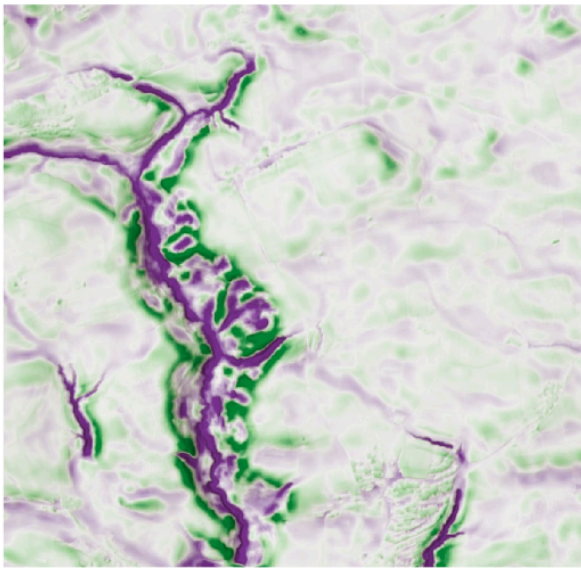
Predictor engineering from elevation model



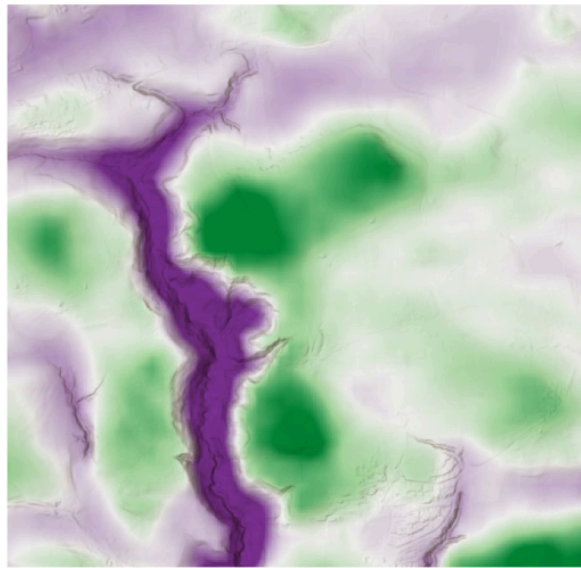
Predictor engineering using different scales

Topographic Position Index (TPI)

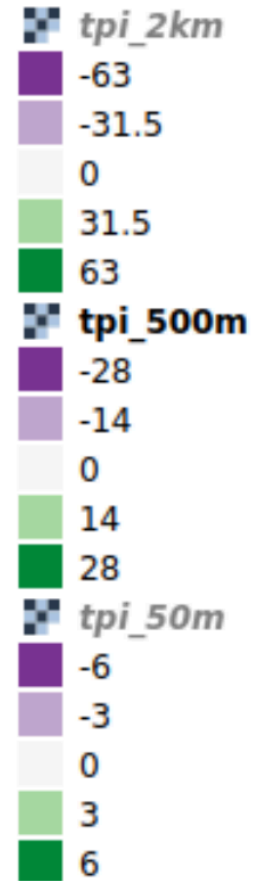
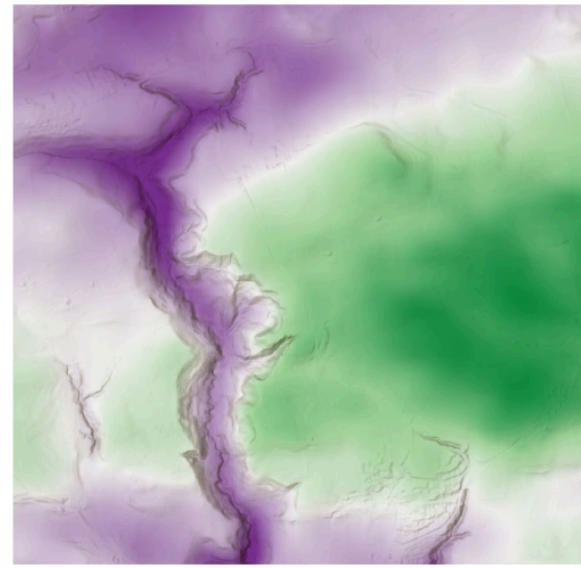
TPI Radius 50 m



TPI Radius 500 m



TPI Radius 2 km

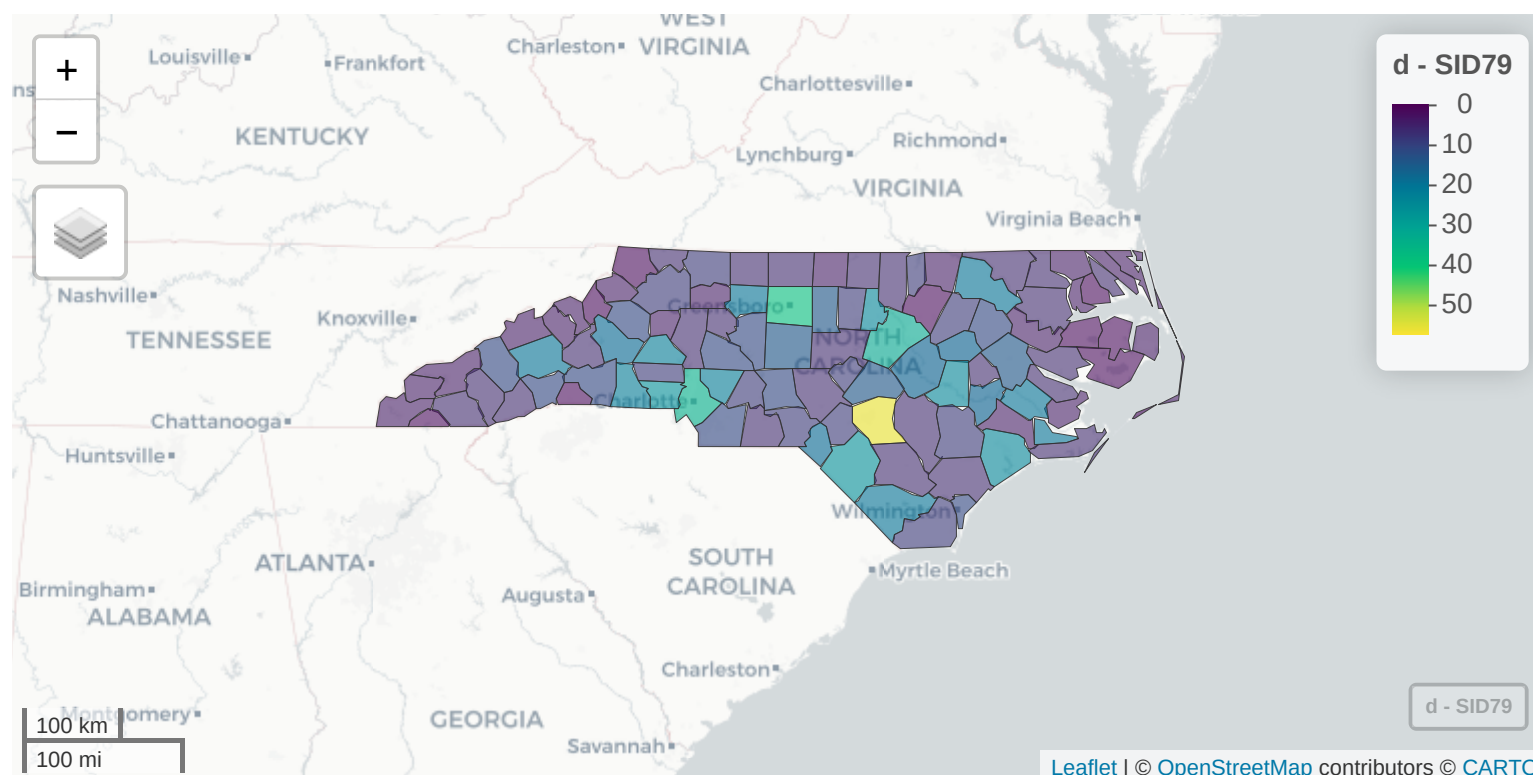


Predictors from vector data

Also areal or lattice data can be possible predictors. Spatial distribution within areas is unknown and we have crisp boundaries, reflecting large uncertainty in the spatial allocation of the data.

A practical approach: transform to raster and apply smoothing.

```
1 library(sf)
2 library(mapview)
3 d <- st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
4 mapview(d, zcol = "SID79")
```



Number of sudden infant deaths per counties of North Carolina (USA) in 1979.

5 Prediction functions

Geostatistical model (kriging approaches)

$$Y = \mathbf{X}\beta + E + Z$$

where

Y : observed point data at location i

$\mathbf{X}\beta$: general trend, e.g. a linear regression with covariates $\mathbf{X}$ (raster data)

E : small scale fluctuations, spatial autocorrelation specified by a variogram

Z : random error (e.g. measurement error, part we cannot explain with model)

- Explicit model for:
Points closer to each other are more similar, than points further away.
- Yields prediction uncertainty directly, many special cases solved.
- Classical statistics, relies on e.g. normally distributed residuals.
- Selecting a trend model is often challenging.

(Spatial) machine learning

$$Y = f(\mathbf{X}) + Z \quad \text{or} \quad Y = f(\mathbf{X}, I) + Z \quad \text{or} \quad Y = f_i(\mathbf{X}) + Z$$

where

Y : observed point data at locations i

$f(X)$: complex function, established by an algorithm using covariates $\mathbf{X}$

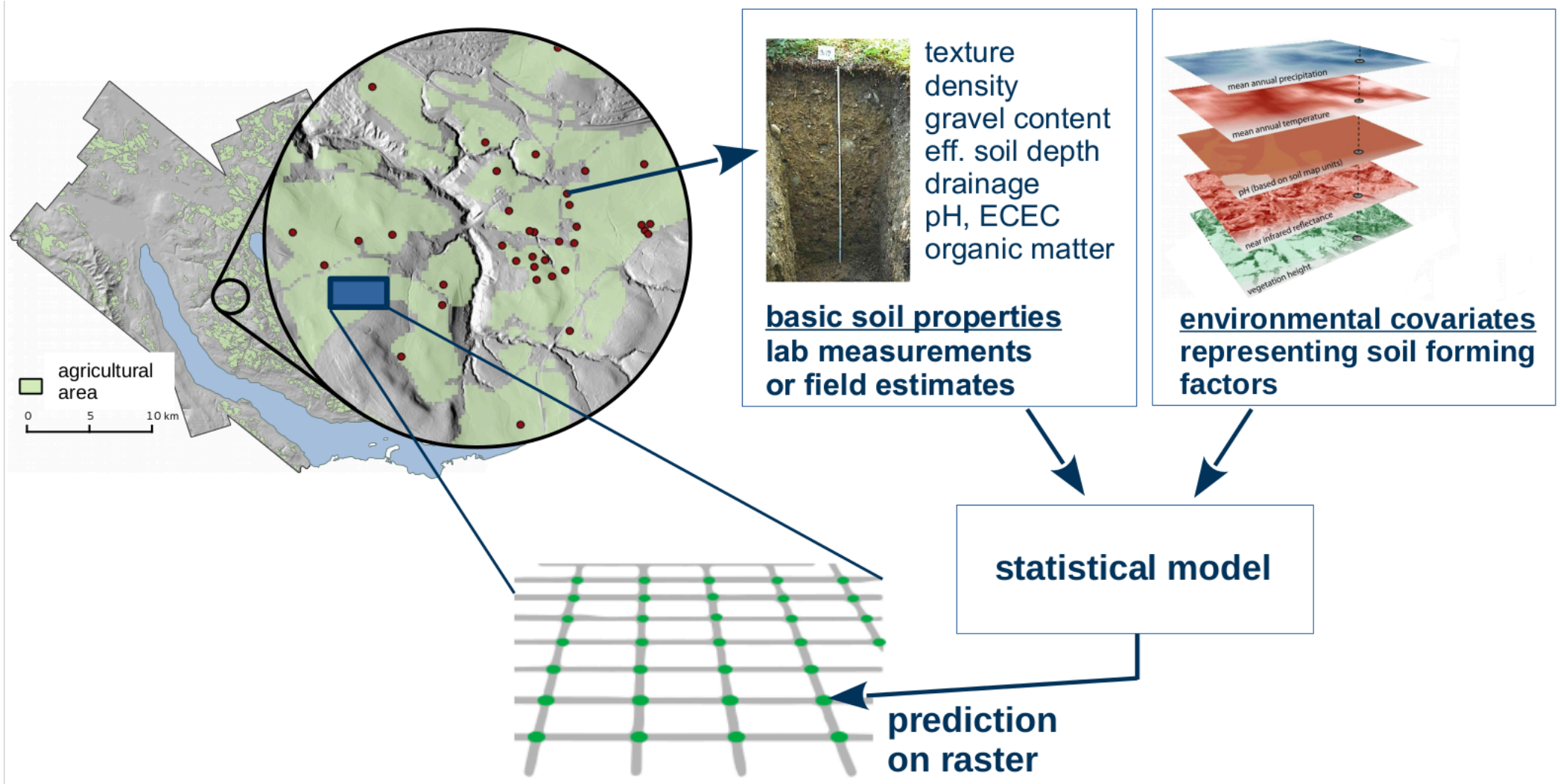
I : location i of the data points

Z : random error (e.g. measurement error, part we cannot explain with model)

- Uses non-spatial prediction approach, sometimes integrates spatial position somehow.
- Easy to find a model, relaxed statistical assumptions.
- Spatial autocorrelation is less explicitly handled, mostly ad-hoc.
- Drawback for special cases (e.g. uncertainty for a mean prediction of a region)

6 Predictive mapping example

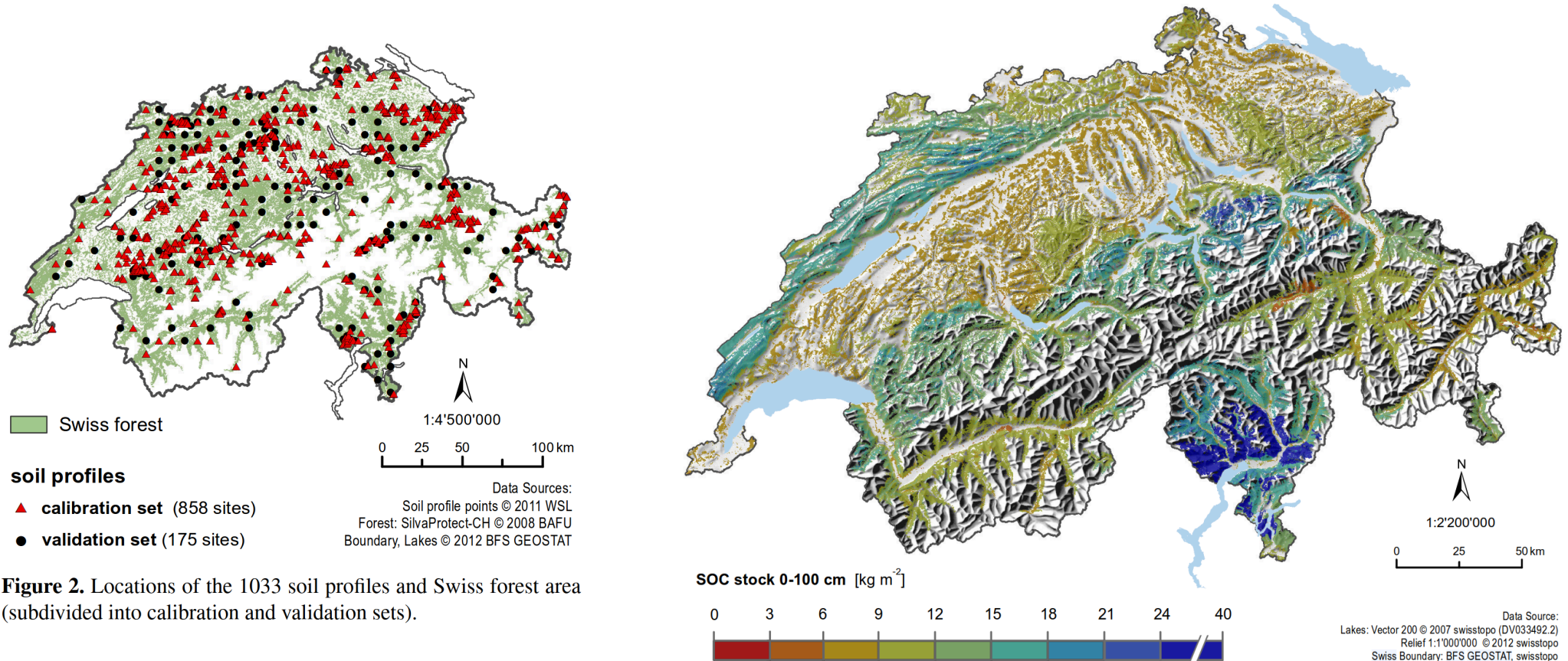
Workflow digital soil mapping



Mapping of soil organic carbon stocks

Swiss government needed data for greenhouse gas inventory.

→ Mapping of carbon stocks in forest soils using kriging with a linear trend model

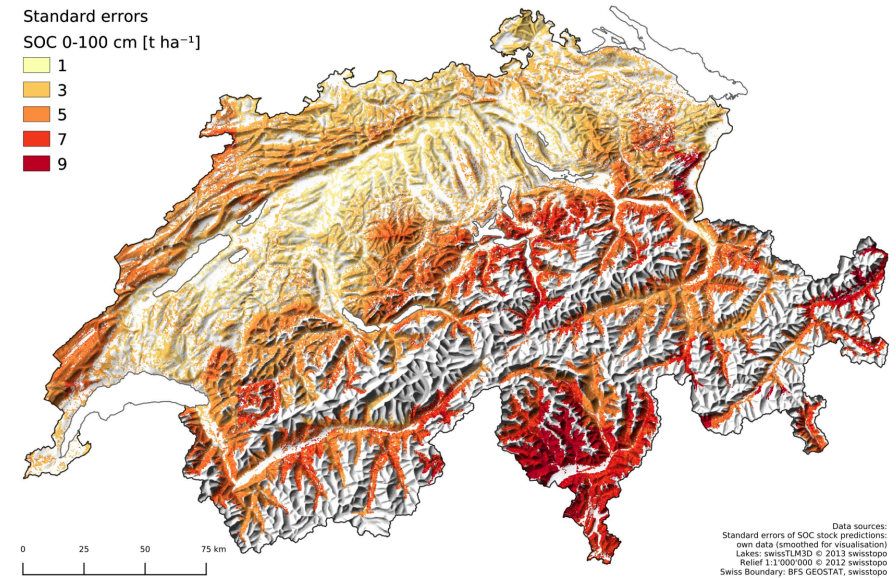
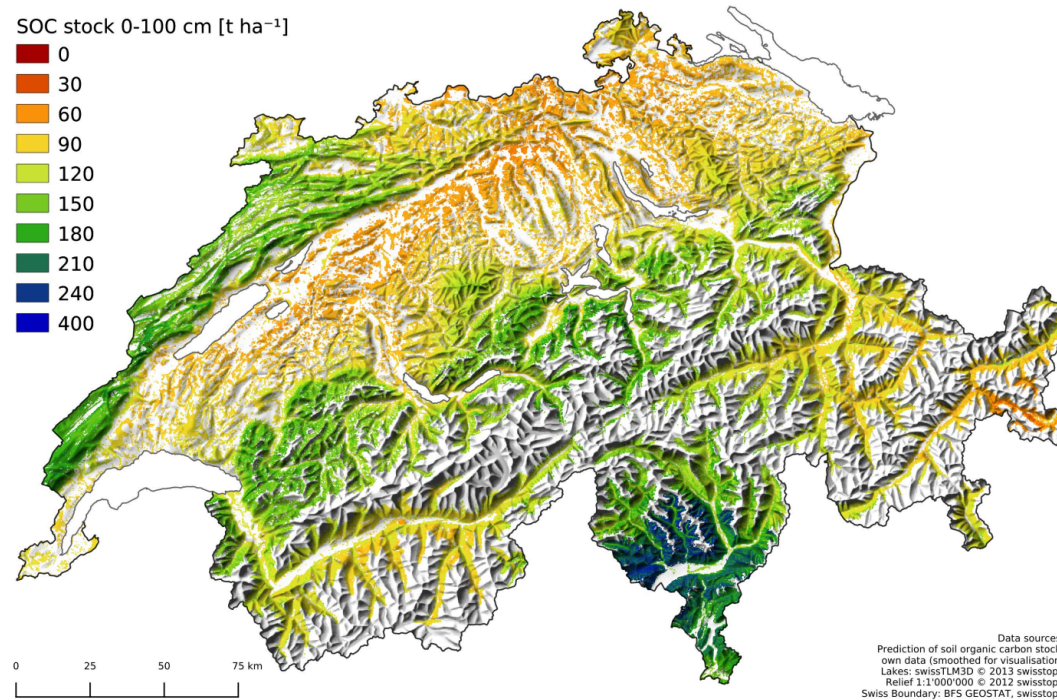


Soil organic carbon stock predictions in soil depth 0-100 cm for forests in Switzerland (Nussbaum et al., 2014)

Map update

Some years later, 850 new soil point data became available.

→ Mapping with random forest



Map showing uncertainty of predictions (Nussbaum and Burgos, 2021)

Same mapped outcome with more point data, updated covariates and different prediction function (caution: colors are different).

7 Goal of spatial predictive analysis

- Point sampling is **sparse & expensive**.
- End-users need actionable spatial information, not just point samples.
- Supports decision-making, environmental assessment, resource or risk management.

Our objective is to

1. Compute spatial **predictions** of response variable for
 - a location without measurement, i.e. creating spatially continuous map,
 - spatial means or totals for pre-defined area (e.g. province),
2. Quantify **uncertainty** of these predictions,
3. Estimate **relationships** of spatial response variable and covariates.

8 Outlook

In short, you will learn:

- Handling spatial data in R
- Find a prediction function / model
- Investigate the spatial structure / auto-correlation of the data
- Interpretation of the prediction model
- Assess the performance of the predictions



**Utrecht
University**

Sharing science,
shaping tomorrow