Data preparation for digital soil mapping

Bas Kempen

Content

- Processing soil point data
- Creating a prediction mask
- Processing covariate layers and creating a stack
- Generating a regression matrix

Data preparation

- Preparing input data often is the most **time-consuming** activity in a digital soil mapping exercise.

- In this lecture: tips and guidelines for preparing input data for DSM.

- Two data sources:
  - **point** data (soil sample data)
  - **covariate** layers (nowadays there is so much data available, e.g. MODIS imagery, SRTM DEM that it is easy to get lost).

- Point data types:
  - soil profile data
  - sampled layers: fixed depths, e.g. the 0-20 cm layer.

Compile and organize point data

- Soil profile data: 2 tables
  - 1 table with properties of sampling sites (e.g. coordinates, soil classification)
  - 1 table with soil profile description (e.g. horizons, depths, soil properties)
- Sampled layers: 1 table.

- 1 variable in each column.
- Each observation (layer) in a row.
- Leave cells with missing data empty: **do not put a 0**!
- Use sensible and short column names; **avoid** white spaces.
- When compiling (legacy) data from different sources make sure the unit of measurement is the same.
- Try to trace the analytical method (standardize/harmonize).

Derive soil property values

- In DSM we typically map the soil property of interest for a specific depth interval.

- Compute soil property values for the target depth layer (e.g. 0–30 cm): **weighted averaging** or **spline**.

- **Weighted averaging** of horizons with weights defined by relative depth contributions

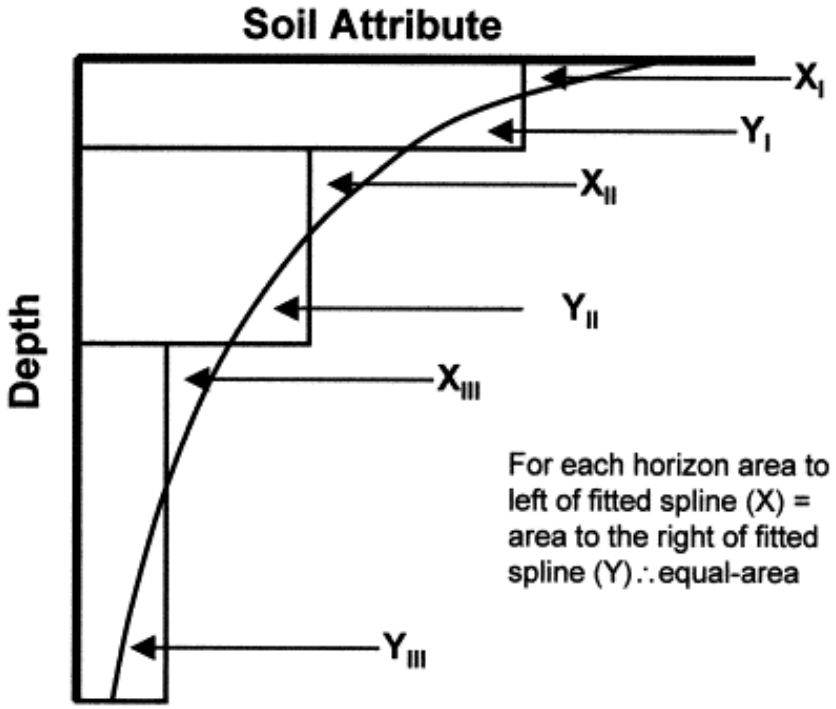
  0 – 20 cm, 6% SOC

  20 – 30 cm, 4% SOC

SOC 0-30 cm: $\frac{20}{30} \times 6\% + \frac{10}{30} \times 4\% = 5.3\%$

Derive soil property values

- **Equal area smoothing splines**:
  - fit a **continuous** depth function to horizon estimates.
  - derives soil property value at a specific depth from the fitted function (e.g. the **midpoint** of the target depth layer).
  - Needs at least two observations.
  - Ideal in case of missing data.



Source: Bishop et al., 1999. Geoderma 91(1-2), 27-45

Equal area splines

- Can be fitted with the **mpspline** function of the **GSIF** package.

- Function should be applied to an object of class **SoilProfileCollection** (**aqp** package).

```
mps <- mpspline(edgeroi.spc, var.name="CLYPPT", vlow=0, vhigh=100)
```

Processing of point data

- In case of **soil profile data**: create 1 table by **joining** the **site** information to the **layer** information based on a common identifier.
- Clean-up: remove unnecessary data

Table with site properties

OBJECTID_1	OBJECTID	ProfileNo	X_coord	Y_coord	**ProfID**
1	4318	43	0	0	P6517
2	4319	13	0	0	P5841
3	4320	35	0	0	P5842
4	4321	26	0	0	P5843
5	4322	20	0	0	P5844
6	4323	36	0	0	P5845
7	4324	18	0	0	P5846

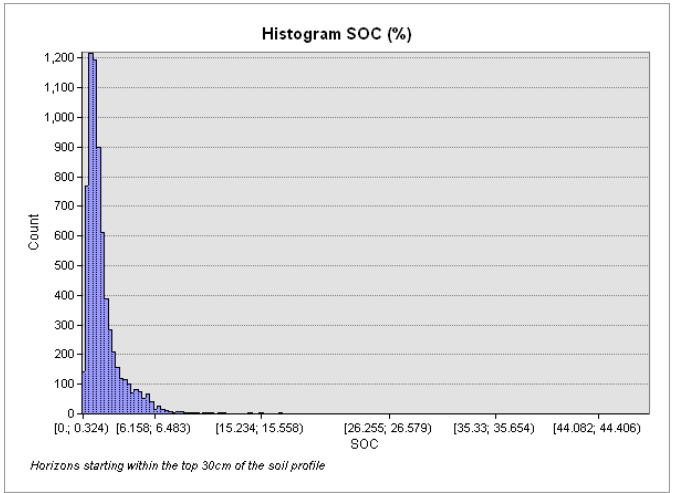
Table with soil profile description

OBJECTID_1	OBJECTID	HorNO	HorID	Depth From	Depth To	Code	SOC	**ProfID**
1	22638	4	P2275H04	106	131	0	-9999	P2275
2	22689	1	P2307H01	0	41	0	-9999	P2307
3	22758	1	P2349H01	0	30	0	-9999	P2349
4	22825	1	P2391H01	0	39	0	-9999	P2391
5	22826	2	P2391H02	39	77	0	-9999	P2391
6	22827	3	P2391H03	98	115	0	-9999	P2391
7	22828	1	P2394H01	0	29	0	-9999	P2394

Processing of point data

- In case of **soil profile data**: create 1 table by **joining** the **site** information to the **layer** information based on a common identifier.
- Clean-up: remove unnecessary data
- Look at summary statistics -> need to transform data?

Min	Q1	Mean	Median	Q3	Max
0	0.87	**1.86**	**1.31**	2.10	48.62



Processing of point data

- In case of **soil profile data**: create 1 table by **joining** the **site** information to the **layer** information based on a common identifier.
- Clean-up: remove unnecessary data.
- Look at summary statistics -> need to transform data?
- Make a spatial plot of the data points.

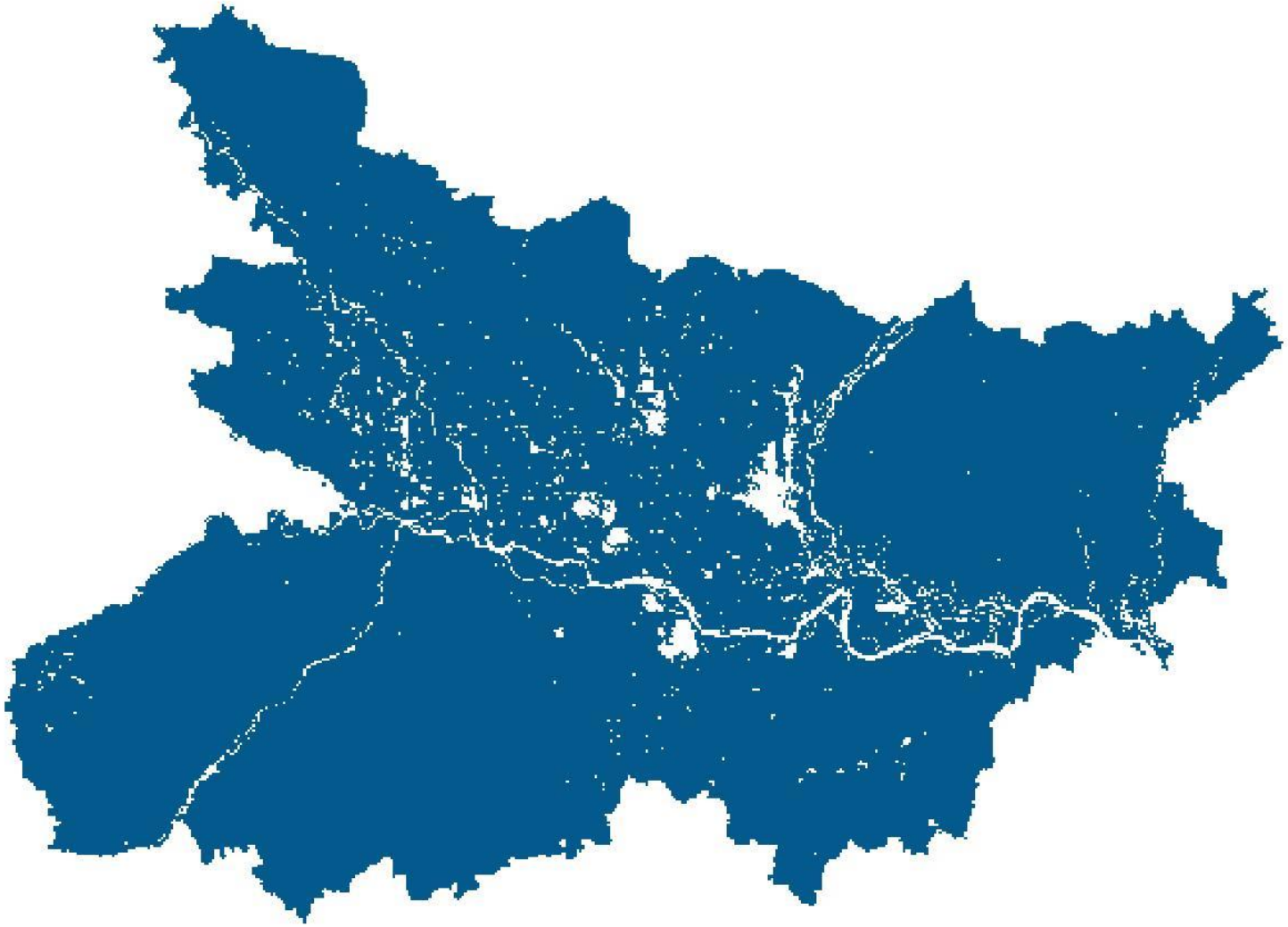
Processing of point data

- In case of **soil profile data**: create 1 table by **joining** the **site** information to the **layer** information based on a common identifier.
- Clean-up: remove unnecessary data.
- Look at summary statistics -> need to transform data?
- Make a spatial plot of the data points.
- Check if data points have identical coordinates (gives problems with kriging).
- Remove locations with missing data from the data set.

Prediction mask

- Mask: area of interest for which we want to predict.
- Mapping area with **non-soil areas** are masked out, e.g. water bodies, rivers, built-up areas, bare rock areas.
- Specifications:
  - Raster format (GeoTiff)
  - Decide on spatial resolution and coordinate system
- Creating a mask:
  - Rasterize an administrative boundary map
  - Use a covariate layer: clip the layer with a admin boundary layer and replace original values with a constant (e.g. 1)
  - Use a land cover map to mask out all non-soil pixels

Mask Bihar State (India)



Covariate sources

- Inventory of covariate sources:
  - Satellite imagery: MODIS, Sentinel, Landsat, AfSIS
  - SRTM Digital Elevation Database
  - Land cover maps (global, regional)
  - Soil maps (national, SOTER databases)
  - ftp://ftp.isric.org/: 171 GeoTiff layers clipped from layers with global coverage at 1 km resolution for each territory in the world, including:
    - MODIS imagery
    - terrain parameters
    - land cover

    username: gsp **password**: gspisric

- Downloading from an FTP server can be done in R.

Covariate processing

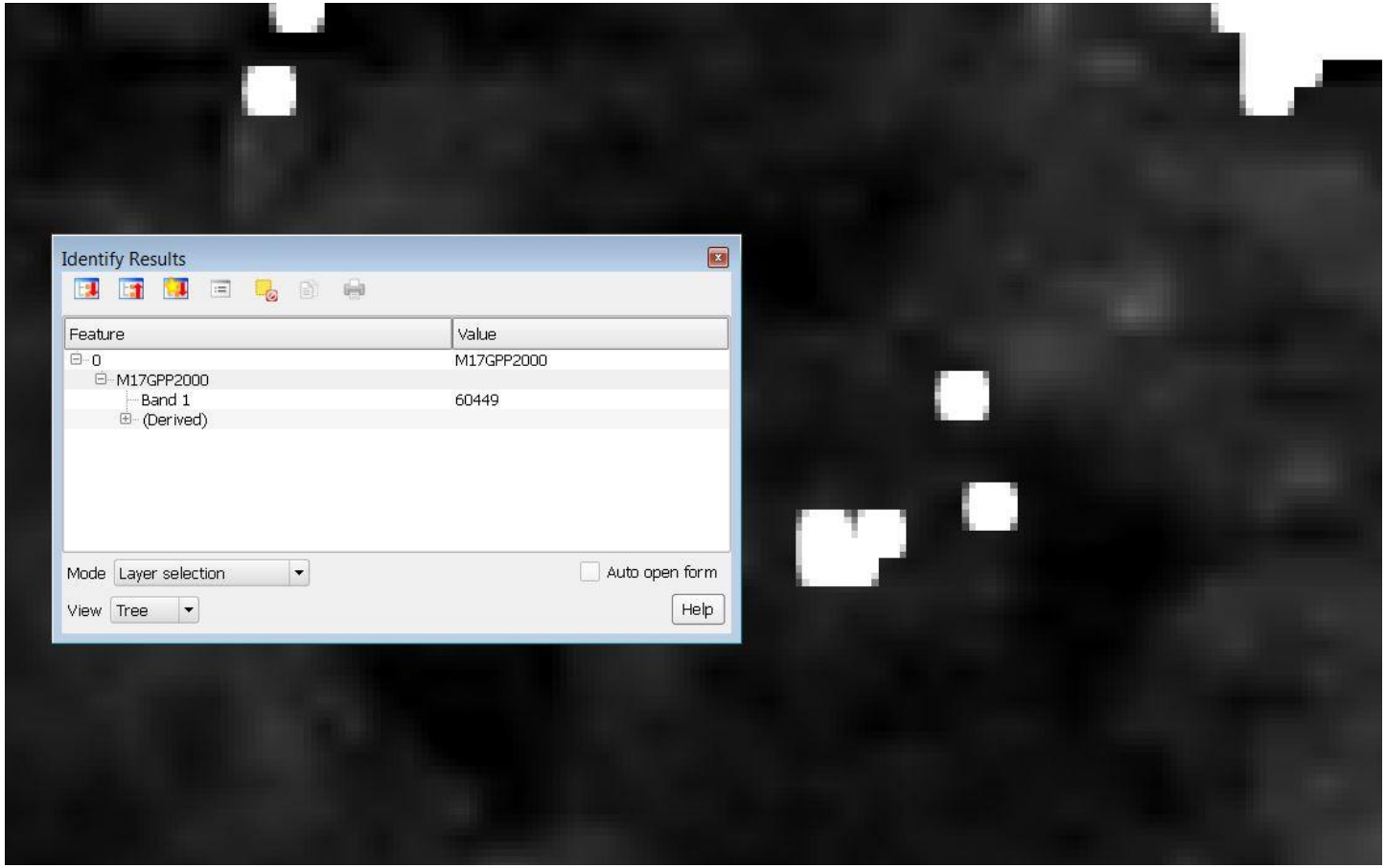
- Re-project if necessary; make sure all data layers have the same geographic reference
- R can work with EPSG codes and character string reference definitions (handy: http://spatialreference.org)
- EPSG projection codes :
  - http://spatialreference.org/ref/epsg/32645/
  - lat-lon: WGS84 = EPSG code 4326
  - UTM Zone 45N = EPSG code 32645
- Reference definitions as character strings (**proj4**):
  - http://spatialreference.org/ref/epsg/32645/proj4/
  - "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
  - "+proj=utm +zone=45 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"

Covariate processing

- (Convert vector data layers to raster)
- (Resample to a common resolution -> mask)
- (Make sure the grid cell exactly align with the cells of the prediction mask -> i.e. have the same origin and resolution)
- Save the raster layers in GeoTIFF format.
- Check the layers!

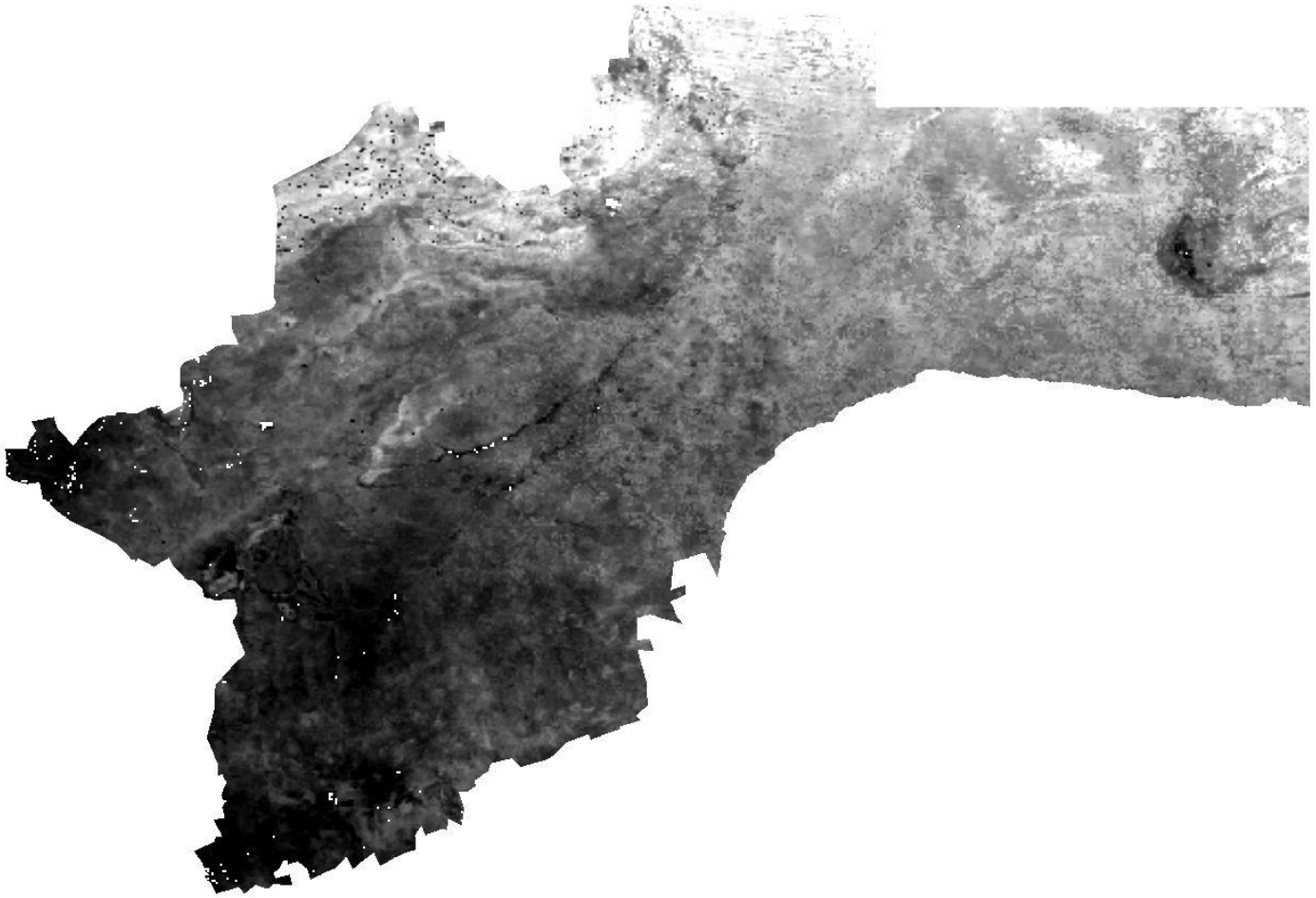
Covariate issues

- Artefacts in covariates



Covariate issues

- NoData values: cannot predict for these locations



Covariate issues

- Constant value: no spatial variation

Creating a covariate stack

- Two ways to create a stack:
  - Stacking layers (**stack** function):
    - Read all covariate layers simultaneously.
    - All layers need to have the **same** extent, resolution, origin, projection as the mask.
    - Clip the stack with the mask to extract the covariate values.
  - Sampling layers (**over** function):
    - Extract covariate values at the grid cell centres of the mask using a spatial overlay (**over** function),
    - One layer at a time, **using a loop**.
    - Layers do not need to have same extent resolution, origin, format (but need to have the same projection).

Creating a covariate stack

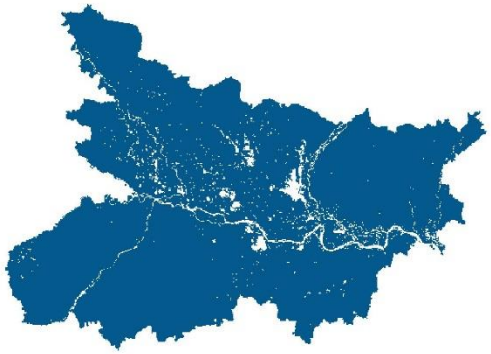
- Store stack as a **SpatialGridDataFrame** or **RasterStack** object (**stack** function).
- Process the stack:
  - **categorical variables**: should be converted to **factor**.
  - **NoData** values (remove pixels).
  - zero or near-zero variance variables (**nearZeroVar** function of caret package).

```
> summary(gridStack@data)
    B02CHE3          B04CHE3          B07CHE3          B13CHE3          B14CHE3
 Min.   :13.46   Min.   :367.0   Min.   :24.94   Min.   : 44.55   Min.   : 0.098
 1st Qu.:16.19   1st Qu.:551.1   1st Qu.:32.80   1st Qu.:188.24   1st Qu.: 4.366
 Median :17.01   Median :604.6   Median :35.26   Median :351.17   Median : 6.349
 Mean   :17.14   Mean   :625.6   Mean   :35.12   Mean   :330.14   Mean   : 7.185
 3rd Qu.:18.45   3rd Qu.:723.7   3rd Qu.:38.13   3rd Qu.:425.13   3rd Qu.:10.195
    C01MCF5          C02MCF5          C03MCF5          C04MCF5          C05MCF5
 Min.   :  902   Min.   : 1320   Min.   : 1078   Min.   : 1036   Min.   : 1401
 1st Qu.: 2312   1st Qu.: 2411   1st Qu.: 1963   1st Qu.: 2232   1st Qu.: 3235
 Median : 3127   Median : 2789   Median : 2774   Median : 3503   Median : 4116
 Mean   : 3229   Mean   : 3236   Mean   : 3306   Mean   : 3978   Mean   : 4651
 3rd Qu.: 4085   3rd Qu.: 3815   3rd Qu.: 4266   3rd Qu.: 5364   3rd Qu.: 5631
```

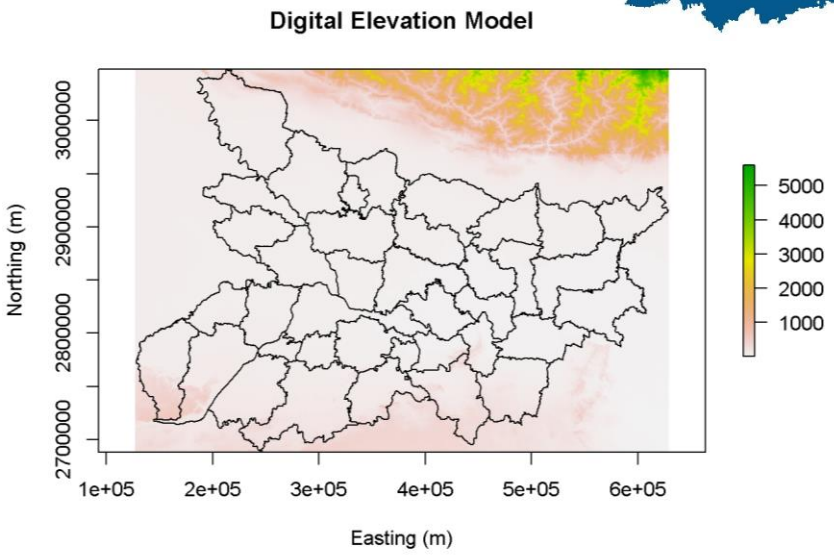
- Save stack as an '.RDATA' or '.rda' file

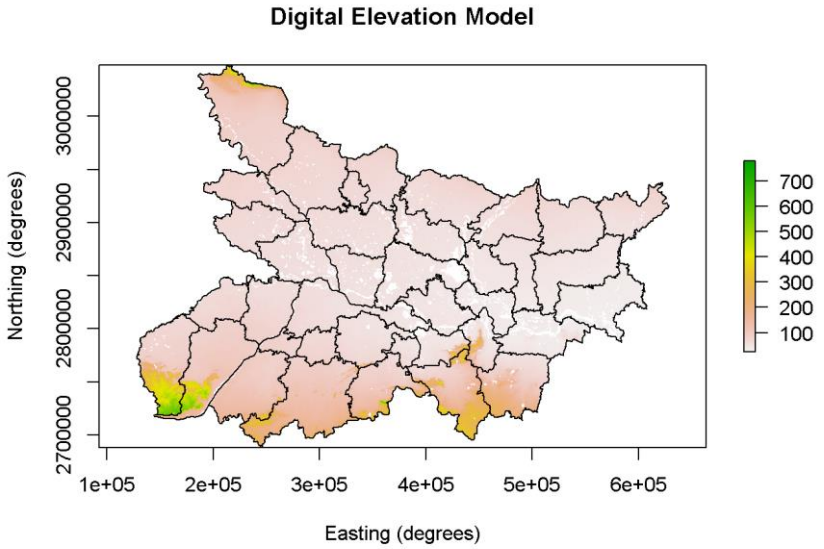
Clipping the covariate stack

- Extract the area of interest using the mask layer (**mask** function).



mask





The regression matrix

- Regression matrix: **table** that contains the values of the covariate layers for soil each sampling site.
- Content (columns):
  - Sample id
  - Coordinates
  - Target soil property
  - Covariate values
- Number of rows equal to the number of sampling sites.
- Input for the statistical DSM model to derive predictive relationships.

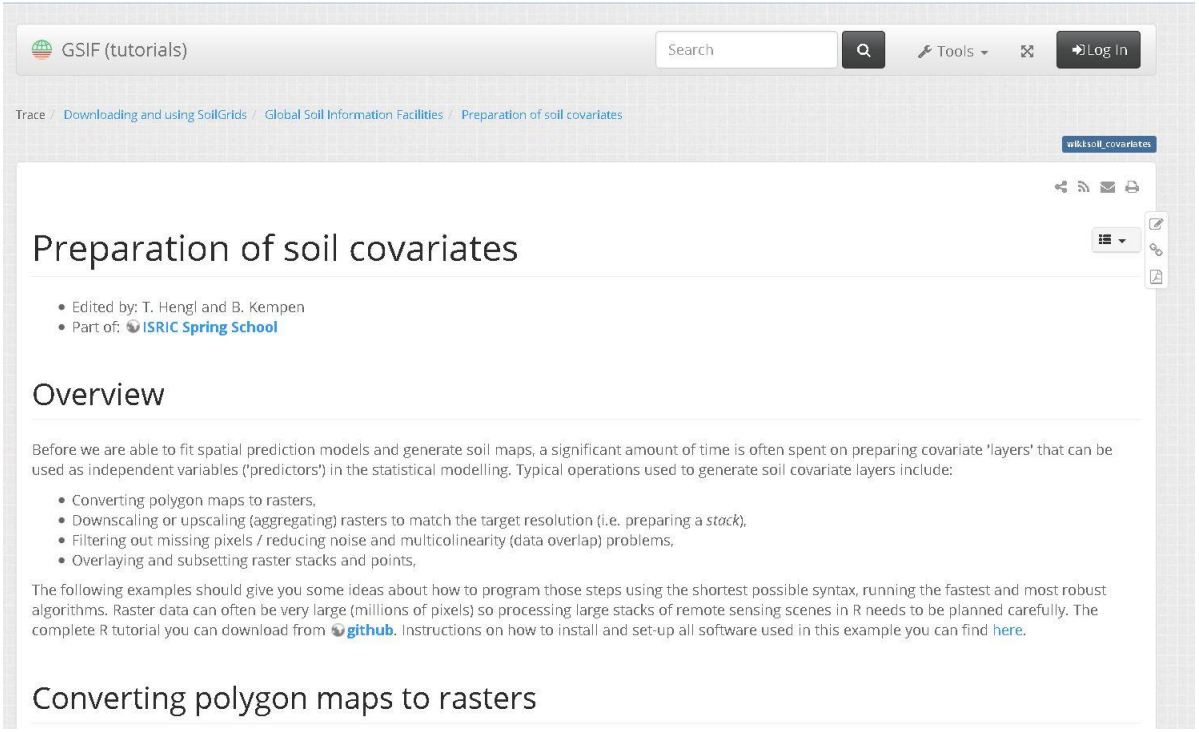
Generating the regression matrix

- spatial overlay:
  - **over** function of the **sp** package
  - **extract** function of the **raster** package
- Remove data points with no covariate data with the **na.omit** or **complete.cases** functions.
- When using **categorical covariates** check if there is at least **one observation** for each category. If not:
  - combine classes in a sensible way
  - convert the categorical covariate with *n* classes to *n* [0,1] **binary variables** indicating presence/absence of each class

  (apply to the covariate layer).

Resources

- http://gsif.isric.org/doku.php/wiki:soil_covariates
  requires installation of gdal and SAGA-GIS
- https://geoscripting-wur.github.io/IntroToRaster/



Thanks for listening...and now let's practice!