

# Spatial Uncertainty Propagation Analysis with the **spup** R Package

by Kasia Sawicka, Gerard B.M. Heuvelink and Dennis J.J. Walvoort

**Abstract** Many environmental and geographical models, such as those used in land degradation, agro-ecological and climate studies, make use of spatially distributed inputs that are known imperfectly. The R package **spup** provides functions for examining the uncertainty propagation from input data and model parameters onto model outputs, via the environmental model. The functions include uncertainty model specification, stochastic simulation and propagation of uncertainty using Monte Carlo (MC) techniques. Uncertain variables are described by probability distributions. Both numerical and categorical data types are handled. Spatial auto-correlation within a variable and cross-correlation between variables is accommodated for. The MC realizations may be used as input to the environmental models written in or called from R. This article provides theoretical background and three worked examples that guide users through the application of **spup**.

## Introduction and motivation

Environmental models that are used to understand and manage natural systems often rely on spatial data as input. Uncertainty in the input data propagates into model predictions. There is a need for systematic analysis of spatial uncertainty propagation in all major inputs and accounting for spatial auto- and cross-correlation between input uncertainties. Spatial uncertainty propagation analysis has been widely recommended and practised across environmental disciplines, for example, hydrology and water quality (e.g. Hengl et al., 2010; Hamel and Guswa, 2015; Muthusamy et al., 2016; Nijhof et al., 2016; Yu et al., 2016), biogeochemistry (e.g. Boyer et al., 2006; Nol et al., 2010; Pennock et al., 2010; Hugelius, 2012; Vanguelova et al., 2016), or ecology and conservation planning (e.g. Jager et al., 2005; Fisher et al., 2010; Lechner et al., 2014). However, as far as we know, there is no widely applicable software that facilitates such analysis.

Currently, a growing number of software tools are available for uncertainty propagation and uncertainty assessment, some of which have functionality for dealing with spatial uncertainties. These include both free software, like OpenTURNS (Andrianov et al., 2007), Dacota (Adams et al., 2009) and DUE (Brown and Heuvelink, 2007), commercial, like COSSAN (Schueller and Pradlwarter, 2006), or free, but written for licenced software, e.g. SAFE (Pianosi et al., 2015) or UQLab (Marelli and Sudret, 2014) toolboxes for MATLAB. A broad review of existing software packages are available in Bastin et al. (2013). The use of powerful but complex languages like C++ (e.g. Dakota), Python (e.g. OpenTURNS) or Java (e.g. DUE) often discourages relevant portions of the scientific community without formal programming skills from the adoption of otherwise powerful tools. There is a number of R packages relating to uncertainty analysis through sensitivity analysis or use of Bayesian frameworks for model calibration. We have found only three packages: **propagate** (Spiess, 2014), **errors** (Ucar, 2017) and **metRology** (Ellison, 2017) that deal with uncertainty propagation explicitly. However, none of these packages provides functionality for spatial models and variables. Therefore, we undertook a project to develop an R package that facilitates uncertainty propagation analysis in spatial environmental and geographical modelling.

In this article we present the **spup** package for R (Sawicka et al., 2017), which implements a methodology for spatial uncertainty propagation analysis using Monte Carlo (MC) methods. **spup** includes functions for uncertainty model specification, sampling, and propagation of uncertainty using MC techniques, and examples of results visualization. In this way, it provides support for the entire chain from uncertainty model definition, simulation, propagation to visualization. It is also a generic tool that can handle a great variety of cases. It is suitable for models that consider spatial variables saved in a spatial data frame or raster, as well as non-spatial variables, both continuous and categorical. It works with models written in R or other programming languages that can be called from R. **spup** is intended for researchers and practitioners who recognise the problems of uncertainty in data and models, and are looking for a simple, accessible implementation of a methodology for uncertainty propagation assessment and visualization. At the same time, it is designed to enable more experienced users to easily understand, customise, and possibly further develop the code.

Here, we provide a description of the implemented methodology, describe the package design and illustrate the usage with three simple examples. We demonstrate that the **spup** package is an effective and easy tool that can be used in multi-disciplinary research, model-based decision support and for educational purposes.

## Methodology implemented in spup

Uncertainty propagation aims to analyse how uncertainties in spatial data (e.g. from measurement error, sampling, interpolation), combined with model uncertainties (e.g. in model parameters and structure) propagate through the model (Heuvelink et al., 2007). Many environmental and geographical phenomena of interest are spatial in nature and often have strong spatial correlations imposed by the physics and dynamics of the natural systems, making uncertainty evaluation difficult.

A common approach to uncertainty propagation analysis makes use of MC stochastic simulation (Hammersley and Handscomb, 1979; Lewis and Orav, 1989). The MC approach is very flexible and can reach an arbitrary level of accuracy, and therefore it is generally preferred over analytical methods such as the Taylor series method (Heuvelink et al., 1989). The MC method for uncertainty propagation analysis consists of the following main steps: (i) uncertainty about the variables is expressed by probability distribution functions (pdfs); (ii) many sets of possible uncertain inputs are generated from their marginal or joint probability distribution using a pseudo-random number generator; (iii) the model is run for each of the simulated input sets; (iv) the set of model outputs forms a random sample from the output pdf, so that the parameters of the output distribution, such as the mean, standard deviation and quantiles, can be estimated from the sample. In other words, the spread in the model outputs characterises how uncertainty about the model inputs has propagated to the model output. Note that the above ignores uncertainty in model parameters and model structure, but these can easily be included if available as pdfs. For example, these might have been obtained through Bayesian calibration (Van Oijen et al., 2005). The above four steps are described in detail further below.

### Step 1 – Characterise uncertain model inputs with pdfs

The most frequently used uncertainty quantification approach represents uncertainty with probability distribution functions (pdfs). In order to represent input uncertainty with a pdf we have to define, for each possible input value, the corresponding probability (or probability density). Probability distribution functions can be very complex and can have many parameters. In practice, when information and data from which to derive the pdf are limited, assumptions and simplifications have to be made to obtain reliable estimates of the pdf. In order to reduce the complexity of a pdf of a continuous variable, the distribution function may be described with a simple parametric shape, such as the normal, uniform or lognormal distribution (Bertsekas and Tsitsiklis, 2008). Categorical uncertain variables can be represented by non-parametric distribution, comprising a set of possible outcomes and their probabilities. The probabilities should be non-negative and the sum of the outcome probabilities should equal 1 (Heuvelink, 2007).

Individual variables, and the uncertainties associated with them, may also be statistically dependent on other variables and their uncertainties. For example, visibility and water vapour in the atmosphere are correlated, and so is the uncertainty between these two properties. Note that correlation between variables as derived from pair observations need not be the same as the correlation between measurement or interpolation errors associated with these variables. For example, elevation and temperature in mountainous area are negatively correlated, while the associated uncertainties may not be correlated at all, because of independent measurements methods. Thus, when multiple inputs are uncertain, cross-correlations in their uncertainties must be addressed. In that case, the uncertainty model is described using a joint probability distribution function (jpdf). If the uncertainties are independent, the jpdf is the product of the univariate pdfs and can be estimated by estimating each pdf separately. If the uncertainties are statistically dependent, these dependencies must be estimated alongside the pdfs. In practice, few parametric shapes are available to describe the jpdf of variables that are statistically dependent. For continuous numerical variables, a common assumption is that the multivariate uncertainty model follows a joint-normal distribution with a vector of means  $\mu$  and variance-covariance matrix  $\Sigma$ , where the latter must be square, symmetric and positive-semidefinite.

In different environmental or geographical studies, many variables are spatially distributed and it is likely that there is a relationship between nearby values (i.e. they are auto-correlated). Uncertainty in such cases may be described by the following geostatistical model:

$$Z(x) = \mu(x) + \sigma(x)\varepsilon(x) \quad (1)$$

where  $\mu(x)$  is the (deterministic) mean of the variable  $Z(x)$  at any location  $x \in D$ ,  $D$  is the geographic domain of interest,  $\sigma(x)$  is a spatially variable standard deviation associated with the prediction  $\mu(x)$  (spatial variation of  $\sigma$  reflects that in some parts of the study area the uncertainty may be greater than in other parts), and  $\varepsilon$  is a zero-mean, unit-variance, second-order stationary (i.e. spatial auto-correlation depends only on a distance vector) residual, modelled with a semivariogram or a correlogram (Diggle and Ribeiro, 2007; Webster and Oliver, 2007; Plant, 2012). For defining spatial correlation, **spup** supports a wide range of positive-definite functions implemented in the **gstat** package (Pebesma, 2004;

Gräler et al., 2016). The mathematical background for these calculations is presented in Brown and Heuvelink (2007) and Heuvelink (2007). In case of spatial variables that are cross-correlated with other spatial variables, the linear model of coregionalization (Goovaerts, 1997) is assumed.

Table 1 summarises common assumptions and simplifications in uncertainty propagation analysis for three categories of data: continuous, discrete and categorical. These assumptions are included in the implementation of **spup**.

## Step 2 – Repeatedly sample from (spatial) pdfs of uncertain inputs

When an uncertain variable is characterised by a pdf, simulation from it relies on a pseudo-random number generator. The most common sampling method is IID sampling (independent identically distributed sampling), in which case each realization is generated independently, each time simulating from the same pdf. In case of a single parameter or univariate distribution, a realization can be easily drawn from various pdfs using algorithms implemented in the **stats** package in R (see <https://cran.r-project.org/web/views/Distributions.html> for details). For a spatially distributed variable (e.g. a DEM) that has a multivariate normal distribution **spup** relies on the unconditional Gaussian simulation algorithm implemented in the **gstat** package (Pebesma, 2004). For simulation from multivariate normal pdfs of non-spatial variables **spup** uses the **mvtnorm** package (Genz et al., 2017; Genz and Bretz, 2009).

In case the input is high-dimensional, straightforward random drawing from the jpdf may not yield a sample that is representative across the whole range of allowable values for each variable, unless the sample size is extremely large. Therefore stratified sampling methods, where values are selected for each variable from each of a pre-specified number of intervals with equal probability mass (Iman and Conover, 1982), are also implemented in **spup**. The latter can also be done in a spatial context (Pebesma and Heuvelink, 1999).

## Step 3 – Run model with sampled inputs and store model outputs

Step 3 consists of running the model (e.g. an environmental model) for all sample elements ('simulated realities') generated in step 2. It is crucial that all models are automated and can be run in batch mode, as in many practical cases, the number of model runs for a Monte Carlo analysis should be at least 100, and often many more (Heuvelink, 1998). In general, the MC sample size must be large enough to guarantee that the outcome of the uncertainty propagation analysis is sufficiently accurate (Heuvelink, 2006). The limitation here is that the required sample size cannot be calculated beforehand because it is derived from the MC output.

It is important to plan what information one needs to retrieve from the set of model outputs, i.e. is the interest in the combined effect of multiple uncertain parameters, or the separate effect of uncertain inputs. The simplest approach to estimating the uncertainty contribution of a single uncertain input or parameter is to perform the uncertainty propagation analysis with only the input or, one or more parameters made uncertain. The remaining uncertain parameters are then assumed certain and fixed on their central value. Comparison of the output uncertainties of different inputs (e.g. ratio of variances) provides a measure of the relative contribution. More details on stochastic sensitivity analysis can be found in Saltelli et al. (2000).

Running the model for multiple simulated realities resulting from step 2 may be computationally demanding, particularly when the model is complex and requires much computing time. The computation time can be reduced, as the MC method is very suited for parallel computing with multi-core machines and for grid computing technology (Leyva-Suarez et al., 2015). In this way, computation times can be dramatically reduced provided the equipment is available in large enough numbers. Numerous high-performance and parallel computing packages have been developed in R suiting various approaches. A continuously maintained descriptive list of these is available online at <https://cran.r-project.org/web/views/HighPerformanceComputing.html> (Eddelbuettel, 2017).

## Step 4 – Compute and visualise summary statistics of model outputs

The resulting set of model outputs forms a random sample from the model output probability distribution. This means that output pdf parameters, such as the mean, median, variance, standard deviation, interquartile range, percentiles, coefficient of variation and threshold exceedance probabilities can be estimated from the sample. For example, the percentiles of the sample can be computed by the `quantile()` function in R. The accuracy of the sample estimates can also be computed (de Gruijter et al., 2006), although this is currently not implemented in **spup**. The sample of output values should be stored in case the model output is used as uncertain input for a next model.

It is important that the output uncertainty statistics are communicated to users in an efficient

Table 1: Common assumptions for defining uncertainty models with a probability distribution function for continuous and discrete variables, and categorical variables.

	Quantifying the univariate pdf	Uncertainty in spatially distributed variables	Cross-correlation between variables
Continuous variable (e.g. rainfall, pollutant concentration, elevation)	<p>A pdf is often assumed to follow the normal distribution with mean <math>\mu</math> and standard deviation <math>\sigma</math>. Non-normal distributions (such as the exponential or lognormal distribution) may also be assumed. For distributions supported in <b>spup</b> see the <b>spup</b> manual. Alternatively, variables may be transformed to ensure that the transformed variable has a normal distribution. Common transformations are square root, logarithm and the Box-Cox transform. Hard minima and maxima can be imposed by using truncated distributions that leave the shape of distributions unchanged except that below the minimum and above the maximum the probability (density) is set to zero.</p>	<p>For continuous uncertain inputs that are spatially correlated, the normal distribution is often imposed because without it the uncertainty quantification would become too complex. Therefore, it is recommended to assume that the input or some transformation of it is multivariate normally distributed and define the uncertainty in terms of the normally distributed (transformed) variable. In addition, 'second-order stationarity' is also often assumed (Goovaerts, 1997; Diggle and Ribeiro, 2007), indicating that the spatial correlation between errors at two locations depends only on the distance vector between the locations. This enables us to represent spatial correlation with a simple function (correlogram).</p>	<p>The statistical dependencies between jointly normally-distributed uncertain variables are fully characterised with a correlation matrix. The correlation matrix is a square, symmetric and positive-semidefinite matrix with unit values on the diagonal and values between -1 and 1 on the off-diagonals. Correlations can be estimated using paired observations of the variables of interest. However, it is important to note that the correlation between variables as derived from paired observations need not be the same as the correlation between measurement or interpolation errors associated with these variables. E.g., DOC and NH4 concentrations of river water are typically positively correlated, while their respective measurement errors are not, if measured independently.</p>
Discrete variable (e.g. fish count, number of wildfires in a given area and time period)	<p>The pdf can be represented by a parametric shape, e.g. Poisson distribution, with mean or rate <math>\lambda</math>, where <math>\mu_Z = \sigma_Z = \lambda</math>. In case no appropriate shape is available, all possible outcomes and associated probabilities must be tabulated or listed in a 'non-parametric' pdf.</p>	<p>When discrete or categorical variables are spatially distributed, the simplest approach is to assume spatial statistical independence or perfect spatial dependence. More advanced approaches that allow intermediate levels of statistical dependence include the Markov random field approach (Blake et al., 2011) and the Generalised Linear Geostatistical Model (Diggle and Ribeiro, 2007). It is also possible to divide or cluster the area into sub-regions, with a complete dependence within and independence between sub-regions.</p>	<p>Cross-dependencies between uncertain categorical variables are addressed by treating all combinations of categories as separate classes. Each combination is assigned a probability. This increases the number of classes dramatically, so that generalisation is often needed in practice and the total number of combined classes is reduced to ten or fewer.</p>
Categorical variable (e.g. land use, building type)	<p>For most categorical variables, a parametric shape may not be available. In that case, each possible outcome and associated probability must be listed in a 'non-parametric' pdf.</p>		



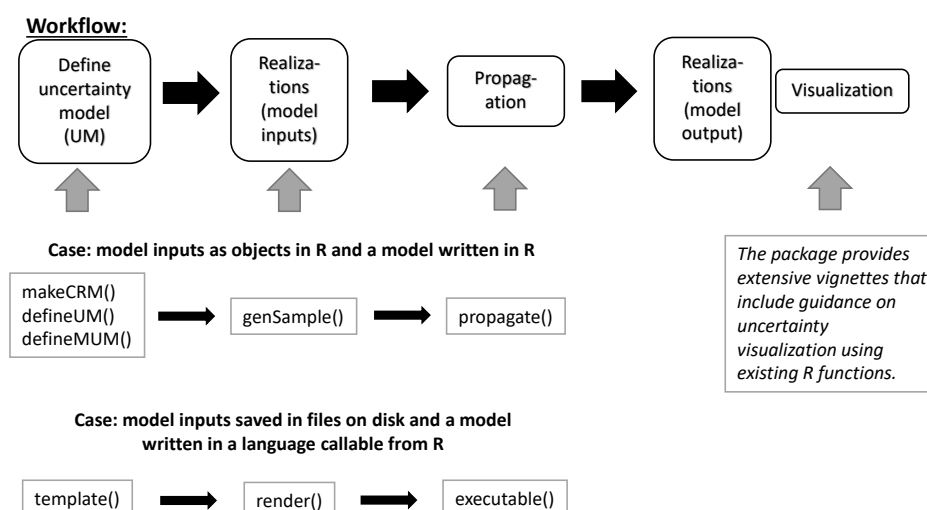
way, and also that these are understandable to users that have little background in statistics. There should be ample explanation of the results and a visual representation of the uncertainty (MacEachren, 1992; Hengl et al., 2004). For non-spatial and non-temporal data, a number of simple statistical plots can be used to visualise uncertainty represented by probability distributions. If the full probability distribution function is known, this can be visualised in pdf or cumulative distribution function (cdf) plots. Error bars, interquartile range and box plots can be used to show the distribution of values. Uncertainties about categorical data can be shown using stacked bars for the probability for each of the categories, or entropy that provides a summary measure of the spread of probabilities over the categories.

For uncertainty visualisation of spatial data it is important to distinguish between presentation techniques and presentation modes. General presentation techniques are (i) adjacent maps, (ii) sequential presentation and (iii) bi-variate maps (MacEachren et al., 2005). These techniques can be presented in three modes: static, dynamic and interactive (Kinkeldey et al., 2014). A common example of adjacent maps is the display of maps of the mean and standard deviation (as uncertainty measure of a continuous variable) next to each other. For visualising spatial categorical attribute uncertainty, the simplest method is also using adjacent maps, for example by showing a map of the the most probable category next to a map of the probability of that category. Sequential presentation may be shown in dynamic mode, e.g. as an animation of realizations.

## The spup package

The **spup** package is developed in R to provide a simple solution for those who wish to include uncertainty analysis in their studies, with emphasises on studies including spatial variables. **spup** implements the methodology described in Section 2.2. Here we describe version 1.3-1 of the package available at <https://CRAN.R-project.org/package=spup/>. The development version is available at <https://github.com/ksawicka/spup>. The package is freely available under the GPL ( $\geq 3$ ) licence.

Three example datasets are provided with the spup package: (i) Digital Elevation Model of Zlatibor region in Serbia (3x4.5km, 30m resolution) (Hengl et al., 2008), (ii) Soil organic carbon and total nitrogen content of the south region of Lake Alaotra in Madagascar (33x33km, 250m resolution) (Hengl et al., 2017), (iii) Rotterdam neighbourhood buildings distribution (Kadaster, The Netherlands). These datasets are used in Section 2.4 and in the vignettes included in the package. Additionally, the package includes simple R and C functions that represent environmental models used in examples.



**Figure 1:** Flowchart presenting workflow with functions implemented in **spup**.

Figure 1 presents a flowchart with the workflow and key functions implemented in **spup**. The package is designed in a way that the key functions correspond with the steps presented in Section 2.2. The output of functions for defining the uncertainty model becomes an input to functions for generating a MC sample. The generated MC sample in turn is used as input to functions facilitating propagation of uncertainty. Qualitative descriptions of the key functions are given in Table 2. More details about functions arguments and output can be found in the package manual (Sawicka et al., 2017) and in the examples below.

**Table 2:** Key functions in package **spup**.

Function	Description
<code>makeCRM()</code>	Define a spatial correlogram model.
<code>defineUM()</code>	Define marginal uncertainty distributions for model inputs/parameters for subsequent Monte Carlo analysis. Output class depends on the arguments provided.
<code>defineMUM()</code>	Allows the user to define joint uncertainty distributions for continuous model inputs/parameters for subsequent Monte Carlo analysis.
<code>genSample()</code>	Methods for generating a Monte Carlo sample of uncertain variables.
<code>template()</code>	Defines a 'container' class to store all templates with model inputs.
<code>executable()</code>	Produces an R function wrapper around a model that can be called using <code>system2()</code> from R.
<code>render()</code>	Replaces the tags in moustaches in template file or character object by text.
<code>propagate()</code>	Runs the model repeatedly with Monte Carlo realizations of the uncertain input/parameters.

## Connections to other packages

**spup** includes imports of the following packages: **graphics** (R Core Team, 2017) facilities for `plot()` method for the "SpatialCorrelogramModel" class; **gstat** (Pebesma, 2004; Gräler et al., 2016) is required by the `genSample()` method when the unconditional Gaussian simulation sampling method is selected for spatially correlated variables; **magrittr** (Bache and Wickham, 2014) provides pipes operators; **methods** (R Core Team, 2017) facilitates for implementation of formal classes and methods; **mvtnorm** (Genz et al., 2017; Genz and Bretz, 2009) is used by the `genSample()` function and supports sampling from a joint distribution of spatially independent variables. **purrr** (Henry and Wickham, 2017) provides functional programming tools; **raster** (Hijmans, 2016) is used in `defineUM()` and `genSample()` functions if variables of interest are saved in raster format; **whisker** (de Jonge, 2013) facilitates templating and rendering.

**spup** suggests packages **dplyr** (Wickham et al., 2017b), **GGally** (Schloerke et al., 2017), **gridExtra** (Auguie, 2016), **knitr** (Xie, 2017, 2015, 2014), **readr** (Wickham et al., 2017c) and **sp** (Pebesma and Bivand, 2005; Bivand et al., 2013) which are used in the vignettes. The **knitr** and **roxygen2** (Wickham et al., 2017a) are required for package development.

## Application examples

### Example 1 - uncertainty propagation analysis with auto- and cross- correlated variables

In many geographical studies, variables are cross-correlated. As a result, uncertainty in one variable may be statistically dependent on uncertainty in the other. For example, soil properties, such as organic carbon (OC) and total nitrogen (TN) content, are cross-correlated. These variables are used to derive the C/N ratio, which is important information to evaluate soil management and increase crop productivity. Errors in OC and TN maps will propagate into the C/N ratio map. The cross-correlation between the uncertainties in OC and TN will affect the degree of uncertainty in the C/N ratio.

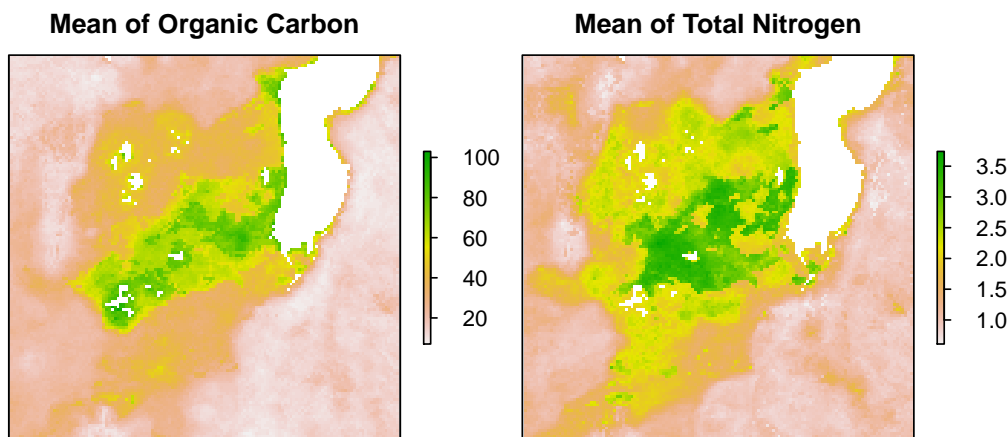
The example data for C/N calculations are a 250m resolution mean OC and TN of a 33km x 33km area adjacent to lake Alaotra in Madagascar (Figure 1) (Hengl et al., 2017). The datasets include four spatial objects: a mean OC and TN of the area and their standard deviations assumed to be equal to 10% of the mean.

```
# load packages
library(spup)
library(raster)
library(purrr)

# set seed
set.seed(12345)
```

```
# load and view the data
data(OC, OC_sd, TN, TN_sd)
class(OC); class(TN)

par(mfrow = c(1, 2), mar = c(1, 1, 2, 2), mgp = c(1.7, 0.5, 0), oma = c(0,0,0,1),
    las = 1, cex.main = 1, tcl = -0.2, cex.axis = 0.8, cex.lab = 0.8)
plot(OC, main = "Mean of Organic Carbon", xaxt = 'n', yaxt = 'n')
plot(TN, main = "Mean of Total Nitrogen", xaxt = 'n', yaxt = 'n')
```



**Figure 2:** Maps of means of soil organic carbon and soil total nitrogen near lake Alaotra in Madagascar.

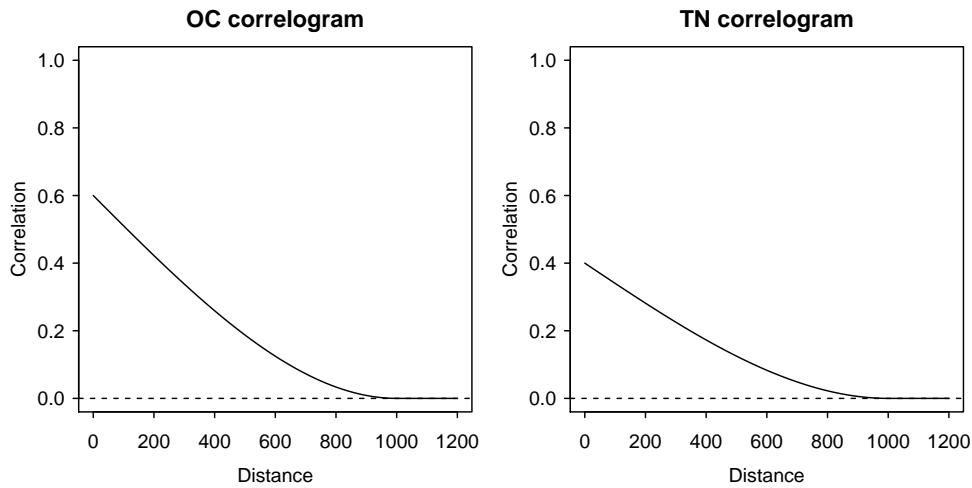
The first step in uncertainty propagation analysis is to define an uncertainty model for the uncertain input variables, here OC and TN, which will be used in the Monte Carlo uncertainty propagation analysis. First, the marginal uncertainty model is defined for each variable separately, and next the joint uncertainty model is defined for the variables together. In case of OC and TN, the  $\varepsilon$  (Eq. 1) are spatially correlated and in order to include this in the analysis, these have to be described by spatial correlograms. For each of the variables, the `makeCRM()` function collates all necessary information into a list. Here it is assumed that the spatial autocorrelation of the OC and TN errors are both described by a spherical correlation function with a short-distance correlation of 0.6 for OC and 0.4 for TN, and a range parameter of 1000m (Figure 3). It is important at this step to ensure that the correlation function types as well as the ranges are the same for each variable. This is a requirement for further analysis, because `spup` employs the linear model of co-regionalization (Wackernagel, 2003).

```
# define spatial correlogram models
OC_crm <- makeCRM(acf0 = 0.6, range = 1000, model = "Sph")
TN_crm <- makeCRM(acf0 = 0.4, range = 1000, model = "Sph")

par(mfrow = c(1, 2), mar = c(3, 2.5, 2, 1), mgp = c(1.7, 0.5, 0), oma = c(0,0,0,0),
    las = 1, cex.main = 1, tcl = -0.2, cex.axis = 0.8, cex.lab = 0.8)
plot(OC_crm, main = "OC correlogram")
plot(TN_crm, main = "TN correlogram")
```

Spatial correlograms summarize patterns of spatial autocorrelation in data and model residuals. They show the degree of correlation between values at two locations as a function of the separation distance between the locations. In the case above the correlation declines with distance (which is usually the case). Here, the correlation is zero for distances greater than 1000m. More information about correlograms is included in the `spup` DEM vignette. In order to complete the description of each individual uncertain variable the `defineUM()` function collates all information about the OC and TN uncertainty.

```
# define uncertainty model for the OC and TN
OC_UM <- defineUM(distribution = "norm", distr_param = c(OC, OC_sd),
    crm = OC_crm, id = "OC")
```



**Figure 3:** Spatial correlograms of soil OC and TN in the study area.

```
TN_UM <- defineUM(distribution = "norm", distr_param = c(TN, TN_sd),
  crm = TN_crm, id = "TN")
```

```
class(OC_UM)
[1] "MarginalNumericSpatial"
class(TN_UM)
[1] "MarginalNumericSpatial"
```

Both variables are of the same class "MarginalNumericSpatial". This is one of the requirements for defining a multivariate uncertainty model next. The `defineMUM()` function collates information about uncertainty in each variable as above, and information about their cross-correlation.

```
# define multivariate uncertainty model
mySpatialMUM <- defineMUM(UMlist = list(OC_UM, TN_UM),
  cormatrix = matrix(c(1,0.7,0.7,1), nrow=2, ncol=2))

class(mySpatialMUM)
[1] "JointNumericSpatial"
```

Thus, in this example a geostatistical model for OC and TN is defined that assumes multivariate normality and has spatial correlation functions:

$$\begin{aligned} \rho_{OC}(h) &= 1 \quad \text{for } h = 0 \\ \rho_{OC}(h) &= 0.6 \cdot \left\{ 1 - \frac{3}{2} \frac{h}{1000} + \frac{1}{2} \left( \frac{h}{1000} \right)^3 \right\} \quad \text{for } 0 < h < 1000 \\ \rho_{OC}(h) &= 0 \quad \text{for } h \geq 1000 \end{aligned} \quad (2)$$

$$\begin{aligned} \rho_{TN}(h) &= 1 \quad \text{for } h = 0 \\ \rho_{TN}(h) &= 0.4 \cdot \left\{ 1 - \frac{3}{2} \frac{h}{1000} + \frac{1}{2} \left( \frac{h}{1000} \right)^3 \right\} \quad \text{for } 0 < h < 1000 \\ \rho_{TN}(h) &= 0 \quad \text{for } h \geq 1000 \end{aligned} \quad (3)$$

$$\begin{aligned} \rho_{OC,TN}(h) &= 0.7 \quad \text{for } h = 0 \\ \rho_{OC,TN}(h) &= 0.7 \cdot \sqrt{0.6 \cdot 0.4} \cdot \left\{ 1 - \frac{3}{2} \frac{h}{1000} + \frac{1}{2} \left( \frac{h}{1000} \right)^3 \right\} \quad \text{for } 0 < h < 1000 \\ \rho_{OC,TN}(h) &= 0 \quad \text{for } h \geq 1000 \end{aligned} \quad (4)$$



where  $h$  is Euclidean distance and where the mathematical definition of the spherical correlation function (i.e., semivariogram) has been used (Webster and Oliver, 2007).

Generating possible realities of the selected variables is completed by using the `genSample()` function:

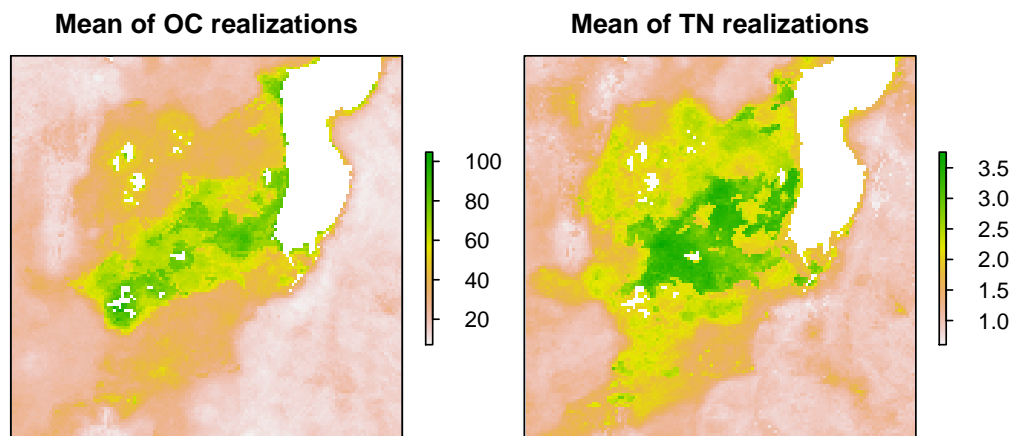
```
# create possible realizations from the joint distribution of OC and TN
MC <- 100
OCTN_sample <- genSample(UMobject = mySpatialMUM, n = MC, samplemethod = "ugs",
                        nmax = 20, asList = FALSE)
```

Note the argument `asList` has been set to `FALSE`. This indicates that the sampling function will return an object of a class of the distribution parameters class. This is useful if there is a need to visualize the sample or compute summary statistics quickly.

A first assessment of uncertainty is done by plotting the means and standard deviations of the sampled OC and TN. Note that if the sample size were very large then the sample mean and standard deviation would approximate the mean and standard deviation as defined in `defineUM()`.

```
# compute and plot OC and TN sample statistics
# e.g. mean and standard deviation
OC_sample <- OCTN_sample[[1:MC]]
TN_sample <- OCTN_sample[(MC+1):(2*MC)]
OC_sample_mean <- mean(OC_sample)
TN_sample_mean <- mean(TN_sample)
OC_sample_sd <- calc(OC_sample, fun = sd)
TN_sample_sd <- calc(TN_sample, fun = sd)

par(mfrow = c(1, 2), mar = c(1, 1, 2, 2), mgp = c(1.7, 0.5, 0), oma = c(0,0,0,1), las = 1,
    cex.main = 1, tcl = -0.2, cex.axis = 0.8, cex.lab = 0.8)
plot(OC_sample_mean, main = "Mean of OC realizations", xaxt = 'n', yaxt = 'n')
plot(TN_sample_mean, main = "Mean of TN realizations", xaxt = 'n', yaxt = 'n')
```



**Figure 4:** Maps of means of sampled realizations of OC and TN.

In order to perform uncertainty propagation analysis using **spup**, the model through which uncertainty is propagated needs to be defined as an R function.

```
# C/N model
C_N_model_raster <- function (OC, TN) OC/TN
```

The propagation of uncertainty occurs when the model is run with the uncertain inputs. Running the model with a sample of realizations of uncertain input variable(s) yields an equally large sample

of model outputs that can be further analysed. To run the C/N ratio model with the OC and TN realizations we use the `propagate()` function. To run the propagation function the samples of uncertain input variables must be saved in lists and then collated into a list of these lists. The existing OCTN\_sample object can be coerced or generated automatically by setting `asList = TRUE` in `genSample()`.

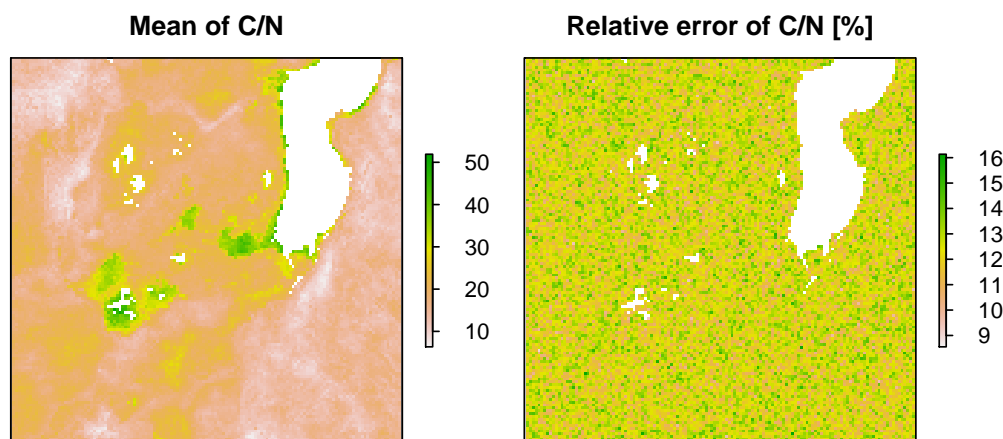
```
# coerce a raster stack to a list
l <- list()
l[[1]] <- map(1:100, function(x){OCTN_sample[[x]]})
l[[2]] <- map(101:200, function(x){OCTN_sample[[x]]})
OCTN_sample <- l

# run uncertainty propagation
CN_sample <- propagate(realizations = OCTN_sample, model = C_N_model_raster, n = MC)

# coerce C/Ns list to a raster stack
CN_sample <- stack(CN_sample)
names(CN_sample) <- paste("CN.", c(1:nlayers(CN_sample)), sep = "")

# compute and plot the slope sample statistics
CN_mean <- mean(CN_sample)
CN_sd <- calc(CN_sample, fun = sd)

par(mfrow = c(1, 2), mar = c(1, 1, 2, 2), mgp = c(1.7, 0.5, 0), oma = c(0,0,0,1),
    las = 1, cex.main = 1, tcl = -0.2, cex.axis = 0.8, cex.lab = 0.8)
plot(CN_mean, main = "Mean of C/N", xaxt = 'n', yaxt = 'n')
plot((CN_sd/CN_mean)*100, main = "Relative error of C/N [%]", xaxt = 'n', yaxt = 'n')
```



**Figure 5:** Maps of standard deviation and relative error of C/N ratio sample in the study area.

Further analysis may include identification of all locations where the C/N ratio is in a specific range, for example where it is higher than 20, with decreasing probability. This information might be used by farmers to identify which plots require action on improving soil quality.

```
# calculate quantiles
CN_sample_df <- as(CN_sample, "SpatialGridDataFrame")
CN_q <- quantile_MC_sgdf(CN_sample_df, probs = c(0.01, 0.1, 0.5, 0.9), na.rm = TRUE)

CN_q$good4crops99perc <- ifelse(CN_q$prob1perc > 20, 1, 0)
CN_q$good4crops90perc <- ifelse(CN_q$prob10perc > 20, 1, 0)
CN_q$good4crops50perc <- ifelse(CN_q$prob50perc > 20, 1, 0)
```

```

CN_q$good4crops10perc <- ifelse(CN_q$prob90perc > 20, 1, 0)

CN_q$good4crops <- CN_q$good4crops99perc + CN_q$good4crops90perc +
  CN_q$good4crops50perc + CN_q$good4crops10perc

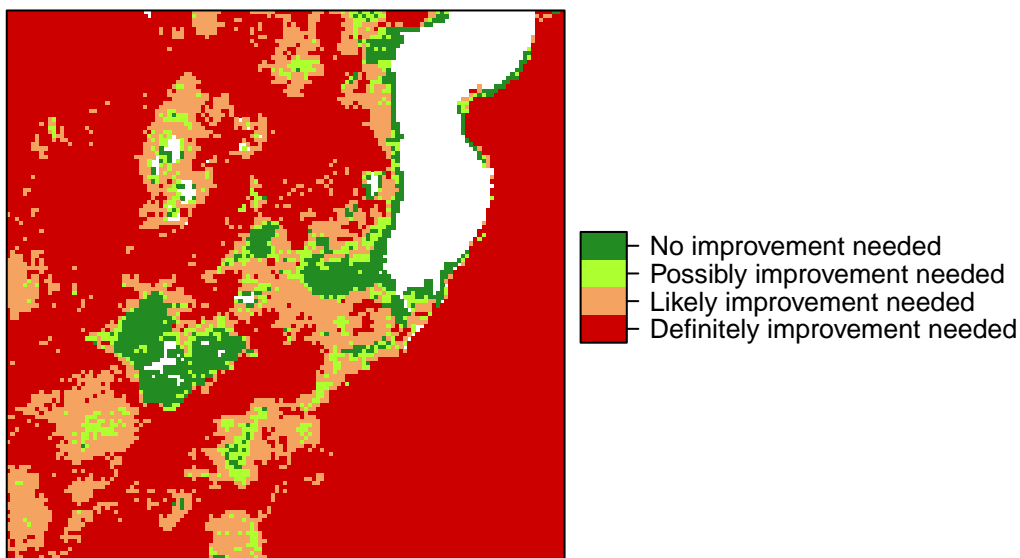
CN_q$good4crops[CN_q$good4crops == 4] <- "No improvement needed"
CN_q$good4crops[CN_q$good4crops == 3] <- "Possibly improvement needed"
CN_q$good4crops[CN_q$good4crops == 2] <- "Likely improvement needed"
CN_q$good4crops[CN_q$good4crops == 1] <- "Definitely improvement needed"
CN_q$good4crops[CN_q$good4crops == 0] <- "Definitely improvement needed"

CN_q$good4crops <- factor(CN_q$good4crops,
  levels = c("Definitely improvement needed",
    "Likely improvement needed",
    "Possibly improvement needed",
    "No improvement needed"))

spplot(CN_q, "good4crops", col.regions = c("red3", "sandybrown", "greenyellow",
  "forestgreen"), main = "Areas with sufficient C/N for cropping")

```

### Areas with sufficient C/N for cropping



**Figure 6:** Map of soil quality classification (depending on C/N ratio) for crop production.

### Example 2 - uncertainty propagation analysis with categorical data

In many aspects of life, information is available and required in a form of categorical data. For example, in a city neighbourhood or a district, each building has assigned a function (housing, office, recreation, etc.). Depending on the building function the city council may impose different tax levels. In this simple example, the function of a building is assigned depending on the building percentage of area inside the building that is used for living (i.e., residential area) and number of addresses inside each building. If the residential area is greater than 80% and has at least one address registered then the building is classified as "residential". If the residential area is smaller than 80% and at least one address is present the building is classified as "office". If no address is present the building function is classified as "other". Depending on these categories, the city council imposes tax levels of €1000, €10000, €10 per year, respectively. The 80% threshold is approximate though and some buildings classified as "others"

could have got an address assigned that has not yet been entered into the system yet. Therefore, the council wishes to calculate the uncertainty in tax revenue depending on errors introduced by erroneous building classification. The example data for tax calculation contain the spatial distribution of buildings in a neighbourhood of Rotterdam, NL.

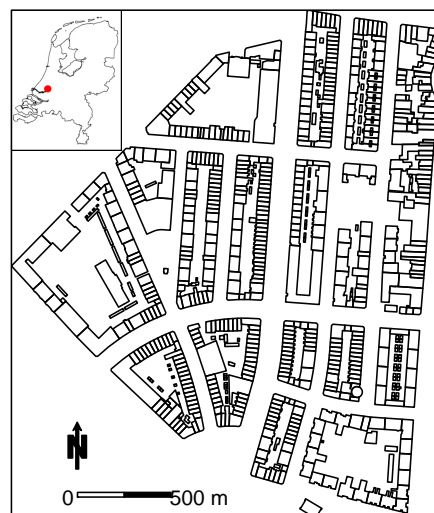
```
# load packages
library(sp)
library(spup)
library(purrr)
library(png)

set.seed(12345)

# load data
data(woon)

# Netherlands contour and Rotterdam location
NL <- readPNG("RotterdamNL.png")
# collate info about figure size and type
NL_map <- list("grid.raster", NL, x = 89987, y = 436047, width = 120,
              height = 152, default.units = 'native')
# collate info about a scale bar location in the figure, type and colour
scale <- list("SpatialPolygonsRescale", layout.scale.bar(),
              offset = c(90000,435600), scale = 100,
              fill=c("transparent","black"))
# collate info about minimum value on a scale bar
text1 <- list("sp.text", c(89990,435600), "0", cex = 0.8)
# collate info about maximum value on a scale bar
text2 <- list("sp.text", c(90130,435600), "500 m", cex = 0.8)
# collate info about North arrow
arrow <- list("SpatialPolygonsRescale", layout.north.arrow(),
              offset = c(89990,435630), scale = 50)
# plot 'woon' object with a location miniature,
# North arrow and scale bar defined above
spplot(woon, "check", do.log = TRUE, col.regions = "transparent",
        colorkey = FALSE, key.space = list(x = 0.1, y = 0.93, corner = c(0,1)),
        sp.layout = list(scale, text1, text2, arrow, NL_map),
        main = "Neighbourhood in Rotterdam, NL")
```

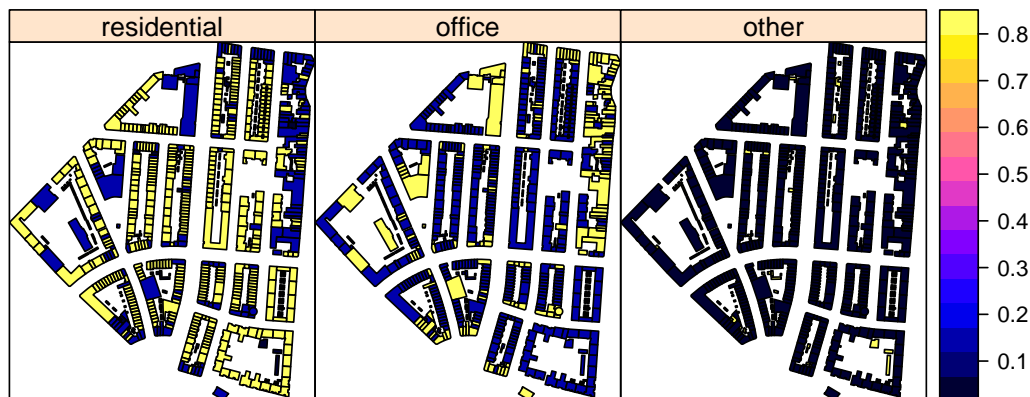
**Neighbourhood in Rotterdam, NL**



**Figure 7:** Map of a neighbourhood of Rotterdam used as study area. Contours represent buildings.

The 'woon' object is a `SpatialPolygonDataFrame` where each building is represented by one polygon (Figure 7). The attributes contain: `vbos` - number of addresses present in the building, `woonareash` - residential area [%]; `Function` - assigned category depending on `vbos` and `woonareash` - for residential it is 1, for office it is 2, for other it is 3; `residential` - probability that the building is residential; `office` - probability that the building is an office; `other` - probability that the building has another function (Figure 8).

```
# plot probabilities for each polygon
splot(woon[c(4,5,6)])
```



**Figure 8:** Probabilities of correct classification of each building.

In case of categorical variables, the uncertainty is described by a non-parametric pdf. In this case it is a vector of probabilities that a building belongs to a certain category. In case of spatially distributed variables, this must be done for each polygon, hence the dataset has maps of these probabilities saved in the same object. We further assume that uncertainties are spatially independent, implying that the joint pdf of all buildings in the neighbourhood is the product of the marginal pdfs for all buildings. Thus we can generate possible realizations for all buildings independently. To unite all information of the uncertainty model for the building function we use the `defineUM()` function that collates all information into one object.

```
# define uncertainty model for the building function
woonUM <- defineUM(TRUE, categories = c("residential", "office", "other"), cat_prob =
  woon[, c(4:6)])

# create possible realizations of the building function
woon_sample <- genSample(woonUM, 100, asList = FALSE)

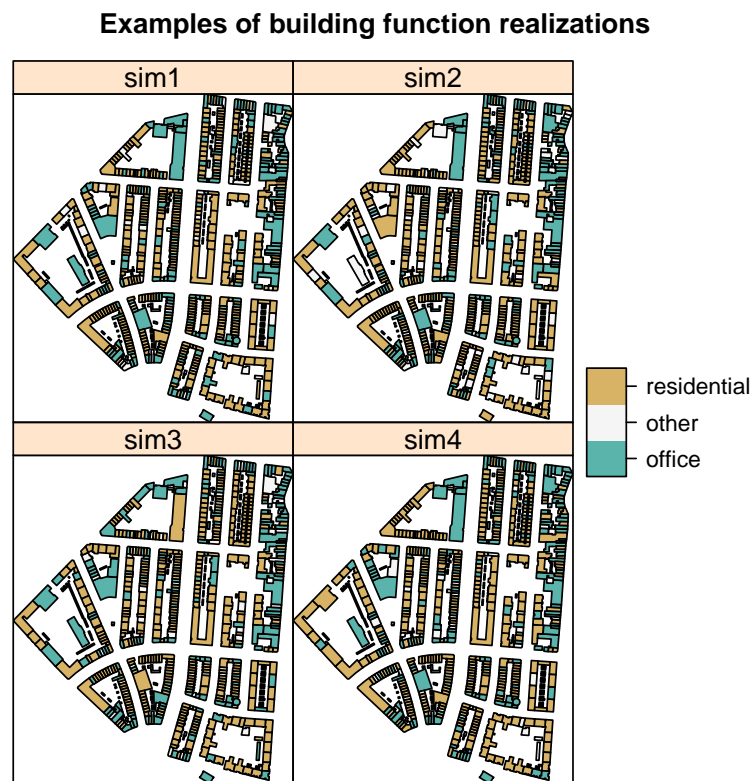
# view several realizations
woon_s <- woon_sample@data
woon_s <- lapply(woon_sample@data, as.factor)
woon_sample@data <- as.data.frame(woon_s)
rm(woon_s)

splot(woon_sample[c(3,4,1,2)], col.regions = c("#5ab4ac", "#f5f5f5", "#d8b365"),
  main = list(label = "Examples of building function realizations", cex = 1))
```

Examples of building functions Monte Carlo realizations are shown in (Figure 9).

To propagate uncertainty, repeatedly run the model with the sample created above. The tax model is defined as:





**Figure 9:** Examples of Monte Carlo realizations of possible classifications for each building.

```
# define tax model
tax <- function (building_Function)
{
  building_Function$tax2pay <- NA
  building_Function$tax2pay[building_Function$Function == "residential"] <- 1000
  building_Function$tax2pay[building_Function$Function == "office"] <- 10000
  building_Function$tax2pay[building_Function$Function == "other"] <- 10
  total_tax <- sum(building_Function$tax2pay)
  total_tax
}

# As in the previous C/N ratio example, in order to run the propagation function
# the sample of an uncertain input variable must be saved in a list.
# Either coerce the existing woon_sample object or get it automatically by
# setting up \code{aslist = TRUE} argument in \code{genSample()}.

# coerce SpatialPolygonDataFrame to a list of individual SpatialPolygonDataFrames
woon_sample <- map(1:ncol(woon_sample), function(x){woon_sample[x]})
for (i in 1:100) names(woon_sample[[i]]) <- "Function"

# run uncertainty propagation
tax_sample <- propagate(woon_sample, model = tax, n = 100)

tax_sample <- unlist(tax_sample)
ci95perc <- c(mean(tax_sample) - 1.96*(sd(tax_sample)/10),
              mean(tax_sample) + 1.96*(sd(tax_sample)/10))
ci95perc
[1] 2281978 2312449
```

The result of the uncertainty propagation shows that on average, the city council should obtain a tax revenue of approximately €2,300,000 and that the associated 95% confidence interval equals <€2,281,978, €2,312,449>.

### Example 3 - uncertainty propagation analysis with a model written in C

Often, environmental models are developed in other languages than R, for example C or FORTRAN. It can significantly speed up processing. In this example, it is shown how to perform uncertainty analysis with a model developed in a different language than R. In this example it is a basic model written in C.

```
library(spup)
library(dplyr) # a grammar of data manipulation
library(readr) # fast I/O
library(whisker) # rendering methods
library(purrr) # functional programming

set.seed(12345)
```

Spatial (or other) inputs to the models are often stored in ASCII files. In that case, when using external models in R, additional code is needed to (i) modify ASCII input files, and (ii) run the external model. For rendering ASCII input files, the mustache templating framework is implemented (<https://mustache.github.io>). In R this is implemented in the package ‘whisker’.

For example, suppose we have a simple linear regression model with two parameters  $b_0$  and  $b_1$  that computes output  $Y$  as a linear transformation of the input  $X$ :  $Y = b_0 + b_1 * X$ . This model needs the input file: ‘input.txt’ which contains values for the two parameters  $b_0$  and  $b_1$ . The contents of the original file may look like:

```
read_lines("input.txt")
[1] "-0.789 0.016"
```

Function ‘template()’ allow user to define a ‘container’ class to store all templates with model inputs. The aim of this class is to organise model input files and perform necessary checks. A ‘print()’ method is also provided for the class “template”.

A template is simply a model input file with:

1. The additional extension ‘.template’.
2. Input that needs to be modified is replaced by mustache-style tags.

The corresponding template file should be named as ‘input.txt.template’. It contains replacement of the original numbers by symbols  $b_0$  and  $b_1$  placed in moustaches: {{ ... }}:

```
read_lines("input.txt.template")
[1] "{{b0}} {{b1}}"
```

Function template() reads the file into R as an object of class “template”:

```
my_template <- template("input.txt.template")
my_template %>%
  read_lines
[1] "{{b0}} {{b1}}"
```

Rendering is the process of replacing the tags in moustaches by text. Rendering will create a new file, called ‘input.txt’.

```
my_template %>%
  render(b0 = 3, b1 = 4)
[1] "input.txt"
```

Below is an example external model written in the C language. This code can be saved in a file with the .c extension, for example, dummy\_model.c. The model below calculates a value of a simple linear regression:

```

#include <stdio.h>
int main() {
    FILE *fp;
    double x[9] = {1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0};
    double y;
    double b0;
    double b1;
    int i;

    fp = fopen("input.txt", "r");
    if (fp == NULL) {
        printf("Can't read input.txt\n");
        return 1;
    }
    fscanf(fp, "%lf %lf\n", &b0, &b1);
    fclose(fp);

    fp = fopen("output.txt", "w");
    if (fp == NULL) {
        printf("Can't create output.txt\n");
        return 1;
    }
    else {
        for (i=0; i<9; i++) {
            y = b0 + b1 * x[i];
            fprintf(fp, "%10.2f\n", y);
        }
        fclose(fp);
    }
    return 0;
}

```

To compile this code to a MS-Windows executable one can use the free C compiler GCC and running the command `gcc dummy_model.c -o dummy_model`. This will create a file 'dummy\_model.exe'. Next this model can be wrapped in R as follows:

```
dummy_model <- executable("dummy_model.exe")
```

Running the rendering procedure allows to pass any values for `b0` and `b1` and the model gives:

```

# render the template
render(my_template, b0 = 3.1, b1 = 4.2)

# run external model
dummy_model()

# read output (output file of dummy_model is "output.txt")
scan(file = "output.txt", quiet = TRUE)
[1]  7.3 11.5 15.7 19.9 24.1 28.3 32.5 36.7 40.9

```

To perform the uncertainty propagation analysis we derive multiple realizations of the model output in steps as follows: (i) render the template, (ii) run the model, (iii) read the results, and (iv) process the results. For example:

```

# number of Monte Carlo runs
n_realizations <- 100

n_realizations %>%
purrr::rerun({

```

```

# render template
render(my_template, b0 = rnorm(n = 1), b1 = runif(n = 1))

# run model
dummy_model()

# read output
scan("examples/output.txt", quiet = TRUE)
}) %>%
set_names(paste0("r", 1:n_realizations)) %>%
as_data_frame %>%
apply(MARGIN = 1, FUN = quantile)

```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
0%	-1.660	-0.9300	-0.7400	-0.730	-0.710	-0.700	-0.680	-0.660	-0.650
25%	-0.060	0.4775	0.8625	1.203	1.720	1.970	2.345	2.607	2.848
50%	0.745	1.2400	1.7200	2.310	2.965	3.375	3.910	4.405	4.850
75%	1.518	2.1425	2.7725	3.453	4.147	4.832	5.453	6.135	6.947
100%	2.990	3.3800	4.2700	5.170	6.080	6.990	7.890	8.800	9.710

## Summary and future directions

In this paper we presented the R package **spup** which implements methodologies for spatial uncertainty propagation analysis. The package architecture and its core components were described and three examples were presented. We explained the package functions for examining the uncertainty propagation, starting from input data and model parameters, via the environmental model onto model predictions. This includes uncertainty model specification, stochastic simulation and propagation of uncertainty using MC techniques, as well as recommendations on visualization of results. We also explained that both numerical and categorical data types are handled and spatial auto-correlation within an attribute and cross-correlation between attributes is accommodated for. The MC realizations may be used as an input to the environmental models written in R or other programming languages, provided that these can be called from R. The presented examples, and more extensively the package vignettes, provide insight into selection of appropriate static uncertainty visualizations that are understandable by non-experts with limited background in statistics. As an effective and easy tool to apply, **spup** has potential to be used in multi-disciplinary research, model-based decision support and for educational purposes.

The package could benefit from further development including facilitating uncertainty analysis with time series, which is one of the most common types of data in environmental studies. For uncertainty propagation, the package has implemented the MC approach with efficient sampling algorithms, i.e. stratified random sampling, and implementing Latin Hypercube sampling would complement this approach. Implementation of interactive visualization methods could be developed as a separate R package that utilises functionality of the **shiny** package (Chang et al., 2017). As it is based on MC methodology, **spup** is suitable to be used with parallel computing.

Currently, two ‘real-world’ applications are in progress. The first study is focused on how uncertainty in spatially distributed rainfall and nitrogen deposition influences greenhouse gas emission predictions using the LandscapeDNDC model (Haas et al., 2013). The second study assesses spatial uncertainty propagation in a metaldehyde pollution prediction model developed as bespoke project for the Severn Trend Water company in the UK.

## Acknowledgements

We thank Sytze de Bruin, Damiano Luzzi and Stefan van Dam for valuable contributions to the development of **spup**. We thank Filip Biljecki and Branislav Bajat for providing the data used in the Rotterdam and Zlatibor examples. This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 607000.

## Authors contributions

Kasia Sawicka is the main developer of **spup** and author of the text in this article. Gerard Heuvelink is a co-author of **spup**, he contributed to this article with his statistical and programming knowledge and edited the text. Dennis Walvoort is a contributor to the **spup** development and the article format.

## Bibliography

- B. M. Adams, L. E. Bauman, W. J. Bohnhoff, K. R. Dalbey, M. S. Ebeida, J. P. Eddy, M. S. Eldred, P. D. Hough, K. T. Hu, J. D. Jakeman, L. P. Swiler, and D. M. Vigil. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.4 user's manual. Report Sandia Technical Report SAND2010-2183, Sandia National Laboratories, 2009. [p1]
- G. Andrianov, S. Burriel, S. Cambier, A. Dufloy, I. Dutka-Malen, E. de Rocquigny, B. Sudret, P. Benjamin, R. Lebrun, F. Mangeant, and M. Pendola. Openturns, an open source initiative to treat uncertainties, risks'n statistics in 520 a structured industrial approach. In *ESREL'2007 Safety and Reliability Conference*, 2007. [p1]
- B. Auguie. *gridExtra: Miscellaneous Functions for "Grid" Graphics*, 2016. URL <https://CRAN.R-project.org/package=gridExtra>. R package version 2.2.1. [p6]
- S. M. Bache and H. Wickham. *Magrittr: A Forward-Pipe Operator for R*, 2014. URL <https://CRAN.R-project.org/package=magrittr>. R package version 1.5. [p6]
- L. Bastin, D. Cornford, R. Jones, G. B. M. Heuvelink, E. Pebesma, C. Stasch, S. Nativi, P. Mazzetti, and M. Williams. Managing uncertainty in integrated environmental modelling: The uncertweb framework. *Environmental Modelling & Software*, 39:116–134, 2013. ISSN 1364-8152. URL <https://doi.org/10.1016/j.envsoft.2012.02.008>. [p1]
- D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability*. Athena Scientific, 2nd edition, 2008. ISBN 9781886529236. [p2]
- R. S. Bivand, E. Pebesma, and V. Gomez-Rubio. *Applied Spatial Data Analysis with R, Second Edition*. Springer-Verlag, 2013. ISBN 9781461476177. [p6]
- A. Blake, P. Kohli, and C. Rother. *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011. [p4]
- E. W. Boyer, R. B. Alexander, W. J. Parton, C. Li, K. Butterbach-Bahl, S. D. Donner, R. W. Skaggs, and S. J. D. Grosso. Modeling denitrification in terrestrial and aquatic ecosystems at regional scales. *Ecological Applications*, 16(6):2123–2142, 2006. ISSN 1939-5582. URL [https://doi.org/10.1890/1051-0761\(2006\)016\[2123:mditaa\]2.0.co;2](https://doi.org/10.1890/1051-0761(2006)016[2123:mditaa]2.0.co;2). [p1]
- J. D. Brown and G. B. M. Heuvelink. The data uncertainty engine (due): A software tool for assessing and simulating uncertain environmental variables. *Computers & Geosciences*, 33(2):172–190, 2007. ISSN 0098-3004. URL <https://doi.org/10.1016/j.cageo.2006.06.015>. [p1, 3]
- W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson. *Shiny: Web Application Framework for R*, 2017. URL <https://CRAN.R-project.org/package=shiny>. R package version 1.0.5. [p17]
- J. J. de Gruijter, D. J. Brus, M. F. P. Bierkens, and M. Kotters. *Sampling for Natural Resource Monitoring*. Springer-Verlag, 2006. URL <https://doi.org/10.1007/3-540-33161-1>. [p3]
- E. de Jonge. *Whisker: mustache for R, Logicless Templating*, 2013. URL <https://CRAN.R-project.org/package=whisker>. R package version 0.3-2. [p6]
- P. J. Diggle and P. J. Ribeiro. *Model-Based Geostatistics*. Springer-Verlag, 2007. ISBN 0387485368. URL <https://doi.org/10.1007/978-0-387-48536-2>. [p2, 4]
- D. Eddelbuettel. Cran task view: High-performance and parallel computing with r, 2017. [p3]
- S. L. R. Ellison. *Support for Metrological Applications*, 2017. URL <https://CRAN.R-project.org/package=metRology>. R package version 0.9-26-2. [p1]
- R. Fisher, N. McDowell, D. Purves, P. Moorcroft, S. Sitch, P. Cox, C. Huntingford, P. Meir, and F. Ian Woodward. Assessing uncertainties in a second-generation dynamic vegetation model caused by ecological scale limitations. *New Phytologist*, 187(3):666–681, 2010. ISSN 1469-8137. URL <https://doi.org/10.1111/j.1469-8137.2010.03340.x>. [p1]



- A. Genz and F. Bretz. *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics. Springer-Verlag, Heidelberg, 2009. ISBN 9783642016882. [p3, 6]
- A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2017. URL <https://CRAN.R-project.org/package=mvtnorm>. R package version 1.0-6. [p3, 6]
- P. Goovaerts. *Geostatistics for Natural Resources Evaluation*. Oxford University Press, New York, 1997. ISBN 9780195115383. [p3, 4]
- B. Gräler, E. Pebesma, and G. Heuvelink. Spatio-temporal interpolation using gstat. *The R Journal*, 8: 204–218, 2016. ISSN 2073-4857. [p3, 6]
- E. Haas, S. Klatt, A. Frohlich, P. Kraft, C. Werner, R. Kiese, R. Grote, L. Breuer, and K. Butterbach-Bahl. Landscapedndc: A process model for simulation of biosphere-atmosphere-hydrosphere exchange processes at site and regional scale. *Landscape Ecology*, 28(4):615–636, 2013. URL <https://doi.org/10.1007/s10980-012-9772-x>. [p17]
- P. Hamel and A. J. Guswa. Uncertainty analysis of a spatially explicit annual water-balance model: Case study of the cape fear basin, north carolina. *Hydrol. Earth Syst. Sci.*, 19(2):839–853, 2015. ISSN 1607-7938. URL <https://doi.org/10.5194/hess-19-839-2015>. [p1]
- J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Chapman and Hall, London, 1979. [p2]
- T. Hengl, D. J. J. Walvoort, A. Brown, and D. G. Rossiter. A double continuous approach to visualization and analysis of categorical maps. *International Journal of Geographical Information Science*, 18(2):183–202, 2004. ISSN 1365-8816. URL <https://doi.org/10.1080/13658810310001620924>. [p5]
- T. Hengl, B. Bajat, D. Blagojevic, and H. I. Reuter. Geostatistical modeling of topography using auxiliary maps. *Computers & Geosciences*, 34(12):1886–1899, 2008. ISSN 0098-3004. URL <https://doi.org/10.1016/j.cageo.2008.01.005>. [p5]
- T. Hengl, G. B. M. Heuvelink, and E. E. van Loon. On the uncertainty of stream networks derived from elevation data: The error propagation approach. *Hydrol. Earth Syst. Sci.*, 14(7):1153–1165, 2010. ISSN 1607-7938. URL <https://doi.org/10.5194/hess-14-1153-2010>. [p1]
- T. Hengl, J. Mendes de Jesus, G. B. M. Heuvelink, M. Ruiperez Gonzalez, M. Kilibarda, A. Blagotić, W. Shangguan, M. N. Wright, X. Geng, B. Bauer-Marschallinger, M. A. Guevara, R. Vargas, R. A. MacMillan, N. H. Batjes, J. G. B. Leenaars, E. Ribeiro, I. Wheeler, S. Mantel, and B. Kempen. Soilgrids250m: Global gridded soil information based on machine learning. *PLOS ONE*, 12(2): e0169748, 2017. URL <https://doi.org/10.1371/journal.pone.0169748>. [p5, 6]
- L. Henry and H. Wickham. *Purrr: Functional Programming Tools*, 2017. URL <https://CRAN.R-project.org/package=purrr>. R package version 0.2.2.2. [p6]
- G. B. M. Heuvelink. *Error Propagation in Environmental Modelling with GIS*. Taylor and Francis, London, 1998. [p3]
- G. B. M. Heuvelink. *Analysing Uncertainty Propagation in GIS: Why Is It Not That Simple?*, book section 10, pages 155–165. John Wiley & Sons, 2006. ISBN 9780470035269. URL <https://doi.org/10.1002/0470035269.ch10>. [p3]
- G. B. M. Heuvelink. Error-aware gis at work: Real-world applications of the data uncertainty engine. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2007. [p2, 3]
- G. B. M. Heuvelink, P. A. Burrough, and A. Stein. Propagation of errors in spatial modelling with gis. *International Journal of Geographical Information Systems*, 3(4):303–322, 1989. ISSN 0269-3798. URL <https://doi.org/10.1080/02693798908941518>. [p2]
- G. B. M. Heuvelink, J. D. Brown, and E. E. van Loon. A probabilistic framework for representing and simulating uncertain environmental variables. *International Journal of Geographical Information Science*, 21(5):497–513, 2007. ISSN 1365-8816. URL <https://doi.org/10.1080/13658810601063951>. [p2]
- R. J. Hijmans. *Raster: Geographic Data Analysis and Modeling*, 2016. URL <https://CRAN.R-project.org/package=raster>. R package version 2.5-8. [p6]

- G. C. G. B. Hugelius. Spatial upscaling using thematic maps: An analysis of uncertainties in permafrost soil carbon estimates. *Global Biogeochemical Cycles*, 26(2):n/a–n/a, 2012. ISSN 1944-9224. URL <https://doi.org/10.1029/2011gb004154>. [p1]
- R. L. Iman and W. J. Conover. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics - Simulation and Computation*, 11(3):311–334, 1982. ISSN 0361-0918. URL <https://doi.org/10.1080/03610918208812265>. [p3]
- H. I. Jager, A. W. King, N. H. Schumaker, T. L. Ashwood, and B. L. Jackson. Spatial uncertainty analysis of population models. *Ecological Modelling*, 185(1):13–27, 2005. ISSN 0304-3800. URL <https://doi.org/10.1016/j.ecolmodel.2004.10.016>. [p1]
- C. Kinkeldey, A. M. MacEachren, and J. Schiewe. How to assess visual communication of uncertainty? a systematic review of geospatial uncertainty visualisation user studies. *The Cartographic Journal*, 51(4):372–386, 2014. ISSN 0008-7041. URL <https://doi.org/10.1179/1743277414y.0000000099>. [p5]
- A. M. Lechner, C. M. Raymond, V. M. Adams, M. Polyakov, A. Gordon, J. R. Rhodes, M. Mills, A. Stein, C. D. Ives, and E. C. Lefroy. Characterizing spatial uncertainty when integrating social data in conservation planning. *Conserv Biol*, 28(6):1497–511, 2014. ISSN 0888-8892. URL <https://doi.org/10.1111/cobi.12409>. [p1]
- P. A. W. Lewis and E. J. Orav. *Simulation Methodology for Statisticians, Operations Analysts, and Engineers*, volume 1. Wadsworth & Brooks/Cole, 1989. ISBN 9780534094508. [p2]
- E. Leyva-Suarez, G. S. Herrera, and L. M. de la Cruz. A parallel computing strategy for monte carlo simulation using groundwater models. *Geofísica Internacional*, 54(3):245–254, 2015. ISSN 0016-7169. URL <https://doi.org/10.1016/j.gi.2015.04.020>. [p3]
- A. M. MacEachren. Visualizing uncertain information. *Cartographic Perspective*, 13:10–19, 1992. [p5]
- A. M. MacEachren, A. Robinson, S. Hopper, S. Gardner, R. Murray, M. Gahegan, and E. Hetzler. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science*, 32(3):139–160, 2005. ISSN 1523-0406. URL <https://doi.org/10.1559/1523040054738936>. [p5]
- S. Marelli and B. Sudret. *UQLab: A Framework for Uncertainty Quantification in Matlab*, pages 2554–2563. American Society of Civil Engineers, 2014. ISBN 978-0-7844-1360-9. URL <https://doi.org/10.1061/9780784413609.257>. [p1]
- M. Muthusamy, A. Schellart, S. Tait, and G. B. M. Heuvelink. Geostatistical upscaling of rain gauge data to support uncertainty analysis of lumped urban hydrological models. *Hydrol. Earth Syst. Sci. Discuss.*, 2016:1–27, 2016. ISSN 1812-2116. URL <https://doi.org/10.5194/hess-2016-279>. [p1]
- C. O. Nijhof, M. A. Huijbregts, L. Golsteijn, and R. van Zelm. Spatial variability versus parameter uncertainty in freshwater fate and exposure factors of chemicals. *Chemosphere*, 149:101–7, 2016. ISSN 0045-6535. URL <https://doi.org/10.1016/j.chemosphere.2016.01.079>. [p1]
- L. Nol, G. B. M. Heuvelink, A. Veldkamp, W. de Vries, and J. Kros. Uncertainty propagation analysis of an n2o emission model at the plot and landscape scale. *Geoderma*, 159(1–2):9–23, 2010. ISSN 0016-7061. URL <https://doi.org/10.1016/j.geoderma.2010.06.009>. [p1]
- E. J. Pebesma. Multivariable geostatistics in s: The gstat package. *Computers & Geosciences*, 30(7):683–691, 2004. ISSN 0098-3004. URL <https://doi.org/10.1016/j.cageo.2004.03.012>. [p2, 3, 6]
- E. J. Pebesma and R. S. Bivand. Classes and methods for spatial data in R. *R News*, 5(2):9–13, 2005. URL <https://CRAN.R-project.org/doc/Rnews/>. [p6]
- E. J. Pebesma and G. B. M. Heuvelink. Latin hypercube sampling of gaussian random fields. *Technometrics*, 41(4):303–312, 1999. ISSN 00401706. URL <https://doi.org/10.2307/1271347>. [p3]
- D. Pennock, T. Yates, A. Bedard-Haughn, K. Phipps, R. Farrell, and R. McDougal. Landscape controls on n2o and ch4 emissions from freshwater mineral soil wetlands of the canadian prairie pothole region. *Geoderma*, 155(3–4):308–319, 2010. ISSN 0016-7061. URL <https://doi.org/10.1016/j.geoderma.2009.12.015>. [p1]
- F. Pianosi, F. Sarrazin, and T. Wagener. A matlab toolbox for global sensitivity analysis. *Environmental Modelling & Software*, 70:80–85, 2015. ISSN 1364-8152. URL <https://doi.org/10.1016/j.envsoft.2015.04.009>. [p1]

- R. E. Plant. *Spatial Data Analysis in Ecology and Agriculture Using R*. CRC Press, 2012. ISBN 1439819130. [p2]
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL <https://www.R-project.org/>. [p6]
- A. Saltelli, S. Tarantola, and F. Campolongo. Sensitivity analysis as an ingredient of modeling. *Statistical Science*, pages 377–395, 2000. ISSN 0883-4237. [p3]
- K. Sawicka, G. B. M. Heuvelink, and D. J. J. Walvoort. *Spup: Uncertainty Propagation Analysis*, 2017. R package version 0.1-1. [p1, 5]
- B. Schloerke, J. Crowley, D. Cook, F. Briatte, M. Marbach, E. Thoen, A. Elberg, and J. Larmarange. *GGally: Extension to 'ggplot2'*, 2017. URL <https://CRAN.R-project.org/package=GGally>. R package version 1.3.1. [p6]
- G. I. Schueller and H. J. Pradlwarter. Computational stochastic structural analysis (cossan) — a software tool. *Structural Safety*, 28(1–2):68–82, 2006. ISSN 0167-4730. URL <https://doi.org/10.1016/j.strusafe.2005.03.005>. [p1]
- A.-N. Spiess. *Propagate: Propagation of Uncertainty*, 2014. URL <https://CRAN.R-project.org/package=propagate>. R package version 1.0-4. [p1]
- I. Ucar. *Errors: Error Propagation for R Vectors*, 2017. URL <https://CRAN.R-project.org/package=errors>. R package version 0.0.2. [p1]
- M. Van Oijen, J. Rougier, and R. Smith. Bayesian calibration of process-based forest models: Bridging the gap between models and data. *Tree Physiology*, 25(7):915–927, 2005. ISSN 0829-318X. [p2]
- E. I. Vangelova, E. Bonifacio, B. De Vos, M. R. Hoosbeek, T. W. Berger, L. Vesterdal, K. Armolaitis, L. Celi, L. Dinca, O. J. Kjønaas, P. Pavlenda, J. Pumpanen, Ü. Püttsepp, B. Reidy, P. Simončič, B. Tobin, and M. Zhiyanski. Sources of errors and uncertainties in the assessment of forest soil carbon stocks at different scales—review and recommendations. *Environmental Monitoring and Assessment*, 188(11):630, Oct 2016. ISSN 1573-2959. doi: 10.1007/s10661-016-5608-5. URL <https://doi.org/10.1007/s10661-016-5608-5>. [p1]
- H. Wackernagel. *Multivariate Geostatistics: An Introduction with Applications*. Springer-Verlag, 2003. [p7]
- R. Webster and M. A. Oliver. *Geostatistics for Environmental Scientists*. John Wiley & Sons, 2nd edition, 2007. [p2, 9]
- H. Wickham, P. Danenberg, and M. Eugster. *Roxygen2: In-Line Documentation for R*, 2017a. URL <https://CRAN.R-project.org/package=roxygen2>. R package version 6.0.1. [p6]
- H. Wickham, R. Francois, L. Henry, and K. Müller. *Dplyr: A Grammar of Data Manipulation*, 2017b. URL <https://CRAN.R-project.org/package=dplyr>. R package version 0.7.1. [p6]
- H. Wickham, J. Hester, and R. Francois. *Readr: Read Rectangular Text Data*, 2017c. URL <https://CRAN.R-project.org/package=readr>. R package version 1.1.1. [p6]
- Y. Xie. Knitr: A comprehensive tool for reproducible research in R. In V. Stodden, F. Leisch, and R. D. Peng, editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC, 2014. ISBN 9781466561595. [p6]
- Y. Xie. *Dynamic Documents with R and Knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. ISBN 9781498716963. [p6]
- Y. Xie. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2017. URL <https://CRAN.R-project.org/package=knitr>. R package version 1.16. [p6]
- X. Yu, A. Lamacova, C. Duffy, P. Kram, and J. Hruska. Hydrological model uncertainty due to spatial evapotranspiration estimation methods. *Computers & Geosciences*, 90, Part B:90–101, 2016. ISSN 0098-3004. URL <https://doi.org/10.1016/j.cageo.2015.05.006>. [p1]

Kasia Sawicka\*  
Soil Geography and Landscape Group  
Wageningen University  
PO Box 47  
6700AA Wageningen

*The Netherlands*  
*(Current address: Centre for Ecology & Hydrology*  
*Deiniol Road*  
*Bangor*  
*LL57 2UW*  
*United Kingdom)*  
[katwic55@ceh.ac.uk](mailto:katwic55@ceh.ac.uk)

*Gerard B.M. Heuvelink*  
*Soil Geography and Landscape group*  
*Wageningen University*  
*PO Box 47*  
*6700AA Wageningen*  
*The Netherlands*  
[gerard.heuvelink@wur.nl](mailto:gerard.heuvelink@wur.nl)

*Dennis J.J. Walvoort*  
*Wageningen Environmental Research*  
*PO Box 47*  
*6700AA Wageningen*  
*The Netherlands*  
[dennis.walvoort@wur.nl](mailto:dennis.walvoort@wur.nl)