Tutorial
Lead ingestion in the Geul valley

Gerard Heuvelink and Bas Kempen
ISRIC - World Soil Information

31 May 2018

Contents

1 Introduction

This practical on uncertainty propagation uses the same data set that was used in the geostatistical practical on Monday. It also makes use of the kriging maps that you produced on Monday, so the first step today is to run Monday's computer practical again. The full script is provided so it should only take a few minutes.

For the general assessment of health risks to children playing in the Geul valley, it could be sensible to make maps of potential daily lead ingestion. This can be done if it is known how much soil a child is likely to ingest per day. Experimental data on the daily ingestion of soil by young children playing on a camping ground have shown that these fit a lognormal distribution [1] with *mean* 0.120, *median* 0.052 and *standard deviation* of 0.250 g/day. A map of the potential daily ingestion of lead can now be obtained by multiplying the map of the lead concentration of the soil by the amount of soil consumed. Uncertainty in both the lead concentration of the soil and the daily soil consumption will propagate to the daily lead ingestion. This implies that even when a site is on average safe, there can be incidences in which the children's health is at risk. In this practical you will assess these risks and calculate which part of the study area is at least with 95% probability safe.

In summary, you will:

1. compute how much lead on *average* a child is expected to ingest per day;
2. use the Monte Carlo method to analyse how uncertainty in the topsoil lead concentration and in the soil consumption propagate to the daily lead ingestion;
3. compute the probability that the daily lead ingestion is above a critical threshold, known as the *Acceptable Daily Intake* (ADI). The ADI is a measure of the amount of a specific substance that can be ingested (orally) over a lifetime without an appreciable health risk. We will assume an ADI of $50\,\mu g$ Pb/day;
4. delineate the area that is below the ADI with 95 % certainty;
5. determine which is the main source of uncertainty: topsoil lead concentration or soil consumption; and
6. (EXTRA) assess how many Monte Carlo runs are needed to reach stable results.

2 Ordinary kriging the lead concentration of the soil

To calculate how much lead a child is expected to ingest per day we multiply the kriged lead concentration map with the mean soil consumption. For this we need the kriged map. Download the `geulgeostats.zip` file from the course webpage, unzip it and run the entire R script in RStudio. This should go without errors, provided all required libraries are installed on your computer and that you have all input files stored in your working directory.

Alternatively, you may also run the script below, which repeats the essentials of Monday's course.

```
> rm(list = ls())  # clean memory
>
> # load libraries
> library(foreign)
> library(sp)
> library(rgdal)
> library(maptools)
> library(gstat)
>
> # read Pb data
> geul <- read.table("geuldata.txt", header = TRUE)
> coordinates(geul) <- ~x+y
>
> # define gstat object and compute variogram
> gpb <- gstat(id = c("pb"), formula = pb~1, data = geul)
> vgpb <- variogram(gpb, boundaries=c(50,100,150,200,300,400,600,800,1000))
> vgmpb <- vgm(psill = 15000, range = 400, model = "Sph",
+              add.to = vgm(psill = 1000, range = 100, model = "Sph"))
> vgmpb <- fit.variogram(vgpb,vgmpb)
> plot(vgpb,vgmpb, plot.numbers = TRUE)
> # read mask
> mask <- readGDAL("geul_mask.txt")
>
> # point kriging
> geul.krig <- krige(pb~1, geul, newdata = mask, vgmpb)
> geul.krig$var1.sd <- sqrt(geul.krig$var1.var)
> spplot(geul.krig[c("var1.pred","var1.sd")], col.regions=bpy.colors())
```

3 Calculating the safe area while ignoring uncertainty

Next we use the `ifelse` function to classify the area where the expected daily lead ingestion is above the ADI:

```
> # safe area based on expected values:
> geul.krig$safe <- factor(ifelse(geul.krig$var1.pred*0.12 > 50,
+                                 1, 0), labels=c("safe","hazard"))
> spplot(geul.krig, zcol = "safe", xlim=c(190200,191300),
+        ylim=c(314300,315600), col.regions = c("green","red"),
+        main = "Safe areas based on point kriging prediction")
```

Q1 : *Which part of the Geul study area is safe, based on the* **expected daily lead ingestion?** **Jump to A1** •

4 Calculating the area that is 95% safe

To analyse the uncertainty propagation and eventually calculate which part of the area is safe with 95% probability we require simulated topsoil lead concentration maps and simulations from the soil consumption probability distribution. We begin with the first. In fact you had already simulated nine realisations during the practical on Monday. Run the same simulation command again but extend the total number of simulations to 100.

```
> # number of Monte Carlo runs
> MC <- 100
>
> # simulate lead concentration maps
> geul.sim <- krige(pb~1, geul, newdata = mask, vgmpb,
+                   nsim = MC, nmax = 24)
>
> # plot a few arbitrarily chosen maps for visual checking
> spplot (geul.sim, zcol = c("sim1", "sim3", "sim8", "sim15"),
+         xlim=c(190200,191300), ylim=c(314300,315600),
+         col.regions = bpy.colors())
```

The soil consumption is a lognormally distributed variable (see above) that is not spatially distributed (it is constant in space). Simulation from the lognormal distribution can be done in R with the function `rlnorm`:

```
> sc <- rlnorm(...,...,...)
```

The only difficulty is how to choose the right values for the arguments used by this function. Check this out with `'?rlnorm'`. One problem is that the function does not use the mean and standard deviation of the original log-normally distributed variable, but instead the parameters of the log-transformed variable (in other words, the mean and standard deviation of the distribution on the log scale). Conversion of these parameters is not trivial, but you can get help from wikipedia: Log-normal_distribution.

Q2 : *Figure out which parameter values should be used when calling* `rlnorm` *to simulate from the lognormal distribution with mean 0.12 and standard deviation 0.25. Once you have the solution, check it by simulating e.g. 100,000 values, calculating the mean and standard deviation, and plotting a histogram.* *Jump to A2* •

Recall that we use 100 Monte Carlo runs, so after testing that the simulation of the soil consumption works fine we simulate only 100 values of the soil consumpion. You had also already generated 100 simulations of the topsoil lead concentration. All that we need to do next is to multiply the simulated soil consumption with the simulated lead concentration map for each Monte Carlo run. This can be achieved with the following code:

```
> ingest <- matrix(nrow=6400,ncol=MC)
> for (i in 1:MC)
+   ingest[,i] <- sc[i] * geul.sim[[i]]
```

Explanation: First, we define a new object `'ingest'` which is a matrix having one row for each pixel of the simulated lead concentration maps and MC columns (recall that the study area is 80 by 80, hence 6400 pixels). The elements of the matrix are filled in the for loop with the product of the

simulated soil consumption and lead concentration. The double brackets refer to the contents of the *i*-th element in the dataframe "data" of "geul.sim" (you may wish to check this with functions `names` or `str`). `sc` does not need double brackets because it is a numeric vector.

To calculate the probability that the Acceptable Daily Intake (ADI) of 50 µg/day is exceeded, you can run the following line of code, which applies the function `mean` on the rows of a matrix produced by the function `ifelse`.

```
> geul.sim$prob <- apply(ifelse(ingest > 50, 1,0), 1, mean)
```

Q3 : *Try to understand how the above code computes probabilities. Display the probability map. Check what are the maximum and minimum probabilities of exceeding the ADI in the study area.* *Jump to A3* •

Finally, calculating the part of the study area that is safe with 95% certainty can be derived from `geul.sim$prob`, again using an `ifelse` statement.

Q4 : *Calculate and display the area that is safe with 95% certainty. Is it much smaller than the area computed in Q1? Is this as expected?* *Jump to A4* •

5 Analysing the contribution of uncertainty sources

Q5 : *Repeat the analysis with only the lead concentration of the soil being uncertain. Do the results change a lot? Repeat the analysis also with only the soil consumption uncertain. Do the results change a lot? What conclusion can you draw from this?* *Jump to A5*
•

6 (EXTRA) Deciding on the number of Monte Carlo runs required

The results you obtained are based on only 100 Monte Carlo runs. To check whether 100 runs is enough, you can repeat the entire analysis (this should be easy provided you have stored all your commands in a script file) and see if the results are similar. You will probably conclude that 100 is not enough.

Q6 : *Increase the number of Monte Carlo runs to 500 or 2000. Repeat the analysis above in which you compare the results from two independent uncertainty propagation analyses (i.e., using a different seed of the pseudo-random number generator). Are results more stable now and can you conclude that 500 (or 2000) Monte Carlo runs is enough?* *Jump to A6* •

7 Answers

A1 : *The majority of the area is safe, except for the hotspot area in the south and a few locations close to the river.*

```
> spplot(geul.krig, zcol = "safe", xlim=c(190200,191300), ylim=c(314300,315600),
+        col.regions = c("green","red"),
+        main = "Safe areas based on point kriging prediction")
```

Return to Q1 •

A2 : *The formulas at wikipedia provide us with the means to calculate the 'location' and 'scale' parameters from the mean and variance of the original variable. The script below implements these in R and checks the result by sampling 100,000 times from the lognormal distribution and calculating the mean and standard deviation. If the mean and standard deviation are close to 0.120 and 0.250, respectively, then we can be assured that we implemented the formula correctly.*

```
> # simulate soil consumption
> set.seed(169)
> MC <- 100000
> m <- 0.12
> s <- 0.250
> locat <- log(m) - 0.5 * log(1 + s**2/m**2)
> scale <- sqrt(log(s**2/m**2 + 1))
>
> sc <- rlnorm(MC, locat, scale)
> mean(sc); sd(sc)
> hist(sc, breaks = "Freedman-Diaconis", xlim=c(0, 2), col="Lightblue")
>
> MC <- 100  # reset MC to its original value
```

Return to Q2 •

A3 : *The apply function applies the* `mean` *function to the outcome of all* `ifelse` *statements (check with* `?apply`*). The latter are ones and zeroes, where a one refers to the case that the lead ingestion is greater than the ADI. Taking the mean of the ones and zeroes boils down to counting how often the lead ingestion is greater than the ADI, and dividing this by the total number of simulations. In other words, it is the proportion of cases that the threshold is exceeded, and hence an estimate of the exceedance probability. The estimate will be close to the true probability if the number of Monte Carlo runs is large.*

```
> spplot(geul.sim, zcol = "prob", col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="P(ingestion > 50 ţg Pb/day) from MC")
> max(geul.sim$prob, na.rm = T)
> min(geul.sim$prob, na.rm = T)
```

Return to Q3 •

A4 : *We again apply the* `ifelse` *function, now checking whether the exceedance probability is greater than 0.05. If it is, the area is unsafe. It turns out that this is a much larger area than when uncertainty is not taken into account. This is not surprising, because we wish to be on the safe side, which means that a smaller part of the study area is considered safe. However, we could not anticipate that such a large part of the area is no longer safe. As it happens, there remains only about*

11% of the area that is safe.

```
> geul.sim$safe = factor(ifelse(geul.sim$prob > 0.05, 0, 1),
+                        labels=c("hazard","safe"))
>
> # plot and save result
> spplot(geul.sim, zcol = "safe", col.regions = c("red", "green"),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="safe areas from MC; ingestion & map uncertain")
> # proportion of area classified as 'safe'
> mean(as.numeric(geul.sim$safe)-1, na.rm=T)
```

Return to Q4 •

A5 : *The answers are obtained by setting each of the uncertain inputs to their default (expected) value, and running the Monte Carlo analysis with sampling only from the remaining uncertain input. Comparsion of results show that soil consumption is the main source of uncertainty in this case.*

```
> # ONLY Pb concentration UNCERTAIN
> MC <- 100
> ingest2 <- matrix(nrow=6400, ncol=MC)
> for (i in 1:MC)
+   ingest2[,i] <- m * geul.sim[[i]]
> geul.sim$prob2 <- apply(ifelse(ingest2 > 50, 1,0), 1, mean)
> geul.sim$safe2 = factor(ifelse(geul.sim$prob2 > 0.05, 1, 0),
+                         labels=c("safe","hazard"))
> geul.sim$var2 <- apply(ingest2, 1, var)
>
> spplot(geul.sim, zcol = "safe2", col.regions = c("green", "red"),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main=paste("Pb concentration uncertain, ",MC, "realizations"))
> # ONLY soil consumption UNCERTAIN
> ingest3 <- matrix(nrow=6400, ncol=MC)
> for (i in 1:MC)
+   ingest3[,i] <- sc[i] * geul.krig$var1.pred
> geul.sim$prob3 <- apply(ifelse(ingest3 > 50, 1,0), 1, mean)
> geul.sim$safe3 = factor(ifelse(geul.sim$prob3 > 0.05, 1, 0),
+                         labels=c("safe","hazard"))
> geul.sim$var3 <- apply(ingest3, 1, var)
> spplot(geul.sim, zcol = "safe3", col.regions = c("green", "red"),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main=paste("Soil consumption uncertain, ",MC, "realizations"))
> # contributions to total variance
> geul.sim$PBcontr <- 100*geul.sim$var2/(geul.sim$var2+geul.sim$var3)
> geul.sim$Scontr <- 100*geul.sim$var3/(geul.sim$var2+geul.sim$var3)
> spplot(geul.sim, zcol = c("PBcontr","Scontr"), col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="Uncertainty contributions (per cent)")
```

Return to Q5 •

A6 : *Below are maps of the 95% safe area for two independent uncertainty propagation analyses using each time 2000 Monte Carlo runs.*

```
> MC <- 2000
>
> # first analysis
> set.seed(784213)
> sc <- rlnorm(MC, locat, scale)
> geul.sim <- krige(pb~1, geul, newdata = mask, vgmpb,
+                   nsim = MC, nmax = 24)
> ingest1 <- matrix(nrow=6400,ncol=MC)
> for (i in 1:MC)
```

```
+   ingest1[,i] <- sc[i] * geul.sim[[i]]
>
> # second analysis
> set.seed(647892)
> sc <- rlnorm(MC, locat, scale)
> geul.sim <- krige(pb~1, geul, newdata = mask, vgmpb,
+                   nsim = MC, nmax = 24)
> ingest2 <- matrix(nrow=6400,ncol=MC)
> for (i in 1:MC)
+   ingest2[,i] <- sc[i] * geul.sim[[i]]
>
> # merge results
> geul.sim$prob1 <- apply(ifelse(ingest1 > 50, 1,0), 1, mean)
> geul.sim$safe1 = factor(ifelse(geul.sim$prob1 > 0.05, 0, 1),
+                         labels=c("hazard","safe"))
> geul.sim$prob2 <- apply(ifelse(ingest2 > 50, 1,0), 1, mean)
> geul.sim$safe2 = factor(ifelse(geul.sim$prob2 > 0.05, 0, 1),
+                         labels=c("hazard","safe"))
>
> spplot(geul.sim, zcol=c("safe1","safe2"), col.regions=c("red", "green"),
+   xlim=c(190200,191300), ylim=c(314300,315600),
+   main="Safe areas, comparing two Monte Carlo analyses with 2000 runs")
```

References

[1] *J. H. Van Wijnen, P. Clausing, and B. Brunekreef. Estimated soil ingestion by children.* Environmental Research*, 51(2):147–162, 1990.* 1