Tutorial
Heavy metals in the Geul valley

Gerard Heuvelink and Titia Mulder
ISRIC - World Soil Information

28 May 2018

Contents

1 Introduction 1
2 Input data 1
3 Software 2
4 Libraries/packages 2
5 Analysing and modelling spatial variation 3
6 Kriging the topsoil lead concentration 6
7 Spatial stochastic simulation 7
8 Regression kriging of the topsoil lead concentration 7
9 Universal kriging 11
10 Answers 12

1 Introduction

The floodplain of the Geul river valley, located in the south of the Netherlands, is strongly polluted by heavy metals. Historic metal mining has caused the widespread dispersal of lead, zinc and cadmium in the alluvial soil. Each time the river flooded the area, polluted sediments were deposited on the river banks. The pollutants may constrain the land use in these areas, so detailed maps are required that delineate zones with high concentrations. In today's practical you will quantify the spatial variation of the topsoil lead (Pb) concentration with a semivariogram and use ordinary and regression kriging to create a map of Pb from point observations. Fig. 1 shows the study area and sampling locations. One of the advantages of kriging is that it also creates a map of the interpolation error, by means of the kriging standard deviation map.

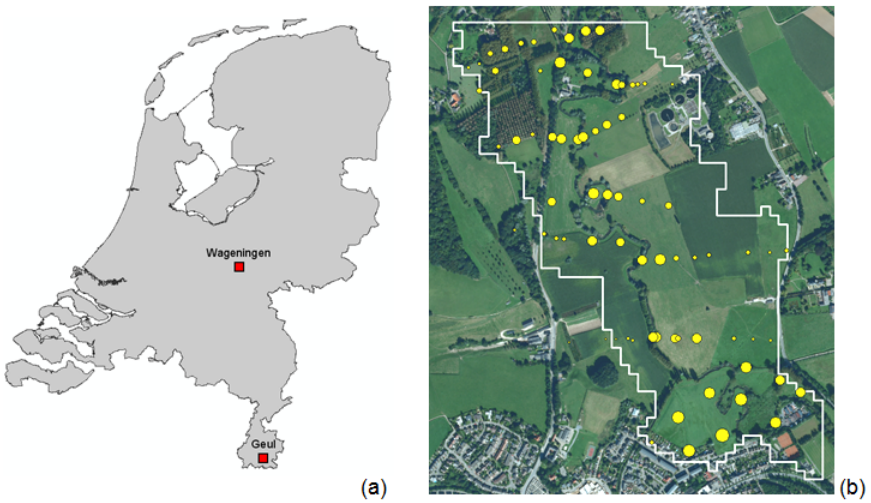

Figure 1. Location of the Geul study area in The Netherlands (a). Zoom in on the study area (b) shows part of the Geul valley and the sampling locations; the size of the symbols is proportional to the lead concentration of the topsoil.

2 Input data

The data needed in this tutorial are stored in the following files (all coordinates are in the Dutch grid projection system):

- `geuldata.txt` ascii file with the 100 lead concentration observations including their geographic coordinates;
- `mask_aoi_shp.xxx` shape files delineating the area of interest;
- `river_line.xxx` shape files representing the river Geul;
- `geul_mask.txt` raster map of the area of interest;

- `geul_dem.txt` digital elevation model of the study area;
- `geul_slope.txt` slope map of the area derived from the digital elevation model;
- `river_dist.txt` raster map with shortest distance to the Geul river for each grid cell.

You can download these files from the course website (geulgeostats.zip). You may wish to open the `.txt` files with a plain text editor to take a look at their content.

3 Software

We use R (http://www.R-project.org) to do all computations. In this practical R is assessed via RStudio™ (http://rstudio.org/), a free and open source integrated development environment for R. R is a very powerful statistical package. Many scientists and computer programmers add new functionality every day. For geostatistical analyses, the R packages `gstat` and `geoR` represent the state-of-the-art and are also used in professional applications. However, using R is not always easy. There are no menu-driven user interfaces where you select your type of analysis with mouse clicks. Instead, you need to type your instructions as high-level programming commands. The handiest way of doing this is by creating a 'script' file in which you type **and save** your commands. You can run these commands by highlighting them and typing 'Ctrl-Enter' or clicking the 'Run line or selection button' in RStudio. By creating script files you automatically archive your work and save it for future use. It also avoids having to type in the same command repeatedly.

R commands are not always logical to the beginning (and experienced!) user and error messages can be very cryptic. Help files also seem sometimes to be written with only an advanced R user in mind. Nonetheless, we have decided to work with R because it is very powerful and integrates basic statistical, geostatistical and GIS functionality in one tool.

4 Libraries/packages

The comprehensive R archive (CRAN) has many useful libraries, which are also known as packages. You can install a library from within R using the `install.packages()` function or in RStudio using Packages | Install Packages (lower right pane). If you are asked to specify a CRAN mirror, choose one of the options in the Netherlands (Amsterdam or Utrecht). For today's practical we use the following libraries: `sp`, `rgdal`, `maptools`, `gstat`, `rgeos` and `MASS`. Packages must first be installed on your computer and next loaded into memory. Packages are loaded in memory using one of the functions `library()` or `require()`. In RStudio, installed

packages can also simply be checked on the tab 'Packages' (lower right pane) to load them, but the other two options also work.

5 Analysing and modelling spatial variation

Copy the `geulgeostats.zip` file to a working directory and unzip it. Start RStudio by double-clicking the shortcut on the desktop or by selecting it with Start - All Programs, etc. Select your working directory in the 'Files' window (using the ... button), set the Working Directory to 'Files pane location' under the 'Session' tab and download and install the above packages. Ask for help if you do not manage. Create a new script file by clicking choose File | New | R script. This opens an editor in the upper left pane in which you can type the commands below. Make sure to **regularly save** the script file. When saving, it is common to use the extension '.R'.

First type and run some preliminaries:

```
> rm(list=ls()) # clean memory
> # load libraries
> library(sp)
> library(rgdal)
> library(maptools)
> library(gstat)
> library(rgeos)
> library(MASS)
```

You can check what commands and functions do and which arguments they use by typing `'?<name of function>'` on the R command line, for example try `'?rm'`.

Next load the topsoil lead concentration data:

```
> geul <- read.table("geuldata.txt", header = TRUE)
```

And try:

```
> dim(geul)
> names(geul)
> summary(geul)
> class(geul)
```

Q1 : *What do the four commands above mean?* *Jump to A1* •

Try also:

```
> # explore the data
> geul
> geul$pb
> mean(geul$pb)
> median(geul$pb)
> min(geul$x); max(geul$x)
```

Calculate and display a histogram of the lead data:

```
> hist(geul$pb, main = "Topsoil lead concentration Geul valley",
+      col="LightBlue")
```

Q2 : *Is the distribution of the topsoil lead concentration symmetric? Are there any obvious outliers?* *Jump to A2* •

We can now convert the Geul dataset to a spatial dataset by converting the x and y columns to coordinates. Type:

```
> # make spatial
> class(geul)
> coordinates(geul) <- ~x+y
> class(geul)
```

Q3 : *R is an object-oriented language. The type or class of the geul object has changed by executing the command above. What class is it now?* *Jump to A3* •

Read the boundary of the study area, the river and make a geographical display of the observations:

```
> # read boundary study area and riverline
> studarea <- readOGR("mask_aoi_shp.shp")
> riverline <- readOGR("river_line.shp")
>
> # plot observations
> spplot(geul, zcol = "pb", xlim = c(190000,192000),
+        ylim = c(314000,316000), cex = 1.5, main = "Pb data",
+        key.space = list(x = 0.02, y = 0.26, corner = c(0,1)),
+        sp.layout = list(list("sp.polygons", studarea),
+        list("sp.lines", riverline, col="red", lwd=2)),
+        scales=list(draw=T), col.regions = bpy.colors(5))
```

Note that long commands can be broken off at the end of a line and continued on the next. The `spplot` function has many arguments, most of which are only meant to upgrade the plot. You may wish to check them out by removing an option and see how the plot changes.

Q4 : *What does the plot reveal about the spatial distribution of Pb (see also Figure 1)?* *Jump to A4* •

You will have noticed that the Pb data have a skew distribution and that high concentrations are found close to the river. This may be incorporated in kriging by transforming the data and/or incorporating a trend that is governed by explanatory variables, such as the distance to the river, elevation or soil type. Common transformations that are often used in geostatistics are the log-transformation and the square root-transformation. More advanced transformations are the Box-Cox transformation and the normal-score transformation. However, for the time being we will neither apply a transformation of the Pb data nor include a trend and proceed with ordi-

nary kriging of the untransformed data (we will incorporate a trend later on, and if time permits you may try kriging the transformed Pb yourself). We begin with calculating the experimental semivariogram and fitting a semivariogram model.

To calculate the experimental (or sample) semivariogram we first define a gstat object:

```
> # define gstat object and compute experimental semivariogram
> gpb <- gstat(formula = pb~1, data = geul)
```

Q5 : *Check with '?gstat' what the arguments that are used mean, particularly the argument 'formula'.* *Jump to A5*

•

Next compute and plot the experimental semivariogram:

```
> vgpb <- variogram(gpb)
> plot(vgpb, plot.nu=FALSE)
```

Q6 : *Is the lead concentration of the topsoil spatially correlated? Explain. Run the above again with option 'plot.nu = TRUE'. What do you get?* *Jump to A6* •

Make a visual estimate of the semivariogram parameters, next define the semivariogram model and plot it together with the experimental semivariogram:

```
> # define initial semivariogram model
> vgmpb <- vgm(nugget = 5000, psill = 25000, range = 400, model = "Sph")
> plot(vgpb, vgmpb)
```

Note that we used a spherical semivariogram as a first guess, but feel free to try other shapes too, such as the exponential ('exp') or Gaussian ('Gau') semivariograms. Next use 'fit.variogram' to fit a semivariogram:

```
> # fit semivariogram model
> vgmpb <- fit.variogram(vgpb, vgmpb, fit.method=7)
> plot(vgpb, vgmpb)
> vgmpb
> attr(vgmpb, "SSErr")
```

Q7 : *Use different options for the semivariogram shape and fit a semivariogram model. Which do you consider best? Why? Do the semivariograms differ much when a different shape is used? Does the nugget of the semivariogram depend much on the chosen shape?* *Jump to A7*

•

6 Kriging the topsoil lead concentration

Ordinary kriging is applied in gstat using the function `krige`. Figure out yourself with `'?krige'` which arguments are needed. One of the arguments is `'newdata'`, in which you define the prediction locations. For this you can use the data in the file `'geul\_mask.txt'`. Load the mask with:

```
> # read prediction grid
> mask <- readGDAL("geul_mask.txt")
```

Run ordinary kriging and store the output of `krige` in the object `'geul.krig'` with the command:

```
> # ordinary point kriging
> geul.krig <- krige(formula = pb~1, locations = geul, newdata = mask,
+                    model = vgmpb)
> names(geul.krig)
```

Q8 : *What are the attributes of geul.krig? What do these represent?* *Jump to A8* •

Q9 : *Plot the kriging prediction map. What do you see? Where are the highs and lows?* *Jump to A9* •

```
> spplot(geul.krig, zcol = "var1.pred", col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="Pb predictions [ppm]")
```

The kriging standard deviation can easily be obtained by applying the function 'sqrt()' to the kriging variance.

Q10 : *Calculate the kriging standard deviation map and plot it. What do you see? Where are the highs and the lows? What are the minimum and maximum value? Do you consider the ordinary kriging prediction map accurate or not?* *Jump to A10* •

```
> geul.krig$var1.sd <- sqrt(geul.krig$var1.var)
> spplot(geul.krig, zcol = "var1.sd", col.regions = bpy.colors(),
+        main="st dev [ppm]", xlim=c(190200,191300), ylim=c(314300,315600),
+        sp.layout=list("sp.points",geul, pch=1, cex=2))
```

Q11 : *What would happen if you would not specify the semivariogram model in the krige call (it will produce a map, but what map is this)? Hypothesize (or 'speculate') first and then check. Is the interpolated map very different from the kriged map?* *Jump to A11* •

```
> # IDW when no semivariogram model specified
> geul.idw <- krige(formula = pb~1, locations = geul, newdata = mask)
> names(geul.idw)
> spplot(geul.idw, zcol = "var1.pred", col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="IDW Pb predictions [ppm]")
```

```
> summary(geul.idw$var1.var)
```

7 Spatial stochastic simulation

Instead of kriging we can also generate 'possible realities' of the topsoil lead concentration by sampling from its probability distribution using a pseudo-random number generator. The theory has not been explained in the lecture but will be addressed in the 'Uncertainty' module later this week. The idea is that at each location a possible value of the true lead concentration is generated by first calculating the kriging probability distribution at that location and next drawing a value from that probability distribution, such that the probability of drawing a certain value is proportional to its probability density. The simulated maps that are generated must take the spatial correlation as specified by the semivariogram into account and must 'condition' the simulations to the observations.

In gstat this is achieved by running the krige command with an additional argument `'nsim'`:

```
> # spatial stochastic simulation
```

```
> geul.sim <- krige(pb~1, geul, newdata = mask, vgmpb, nsim = 9, nmax = 24)
> names(geul.sim)
> spplot(geul.sim, zcol = "sim1", xlim=c(190200,191300),
+        ylim=c(314300,315600), col.regions = bpy.colors())
> spplot(geul.sim[1:6], xlim=c(190200,191300),
+        ylim=c(314300,315600), col.regions = bpy.colors())
```

Q12 : *Compare the simulated maps with the kriging map. What are the main differences? What do the differences between simulated maps represent? If we would generate a large number of simulations (say >10,000), what would a map of their average look like? And a map of the per grid-cell standard deviations? What could be the practical use of such simulated maps?* *Jump to A12* •

8 Regression kriging of the topsoil lead concentration

The spatial variation in soil properties depends on existing soil-landscape relationships within the study area. Therefore, soil properties are often correlated with environmental properties such as land use, elevation, slope, etc. In essence, these environmental properties are proxies of the soil forming factors (i.e., CLORPT = CLimate, Organisms, Relief, Parent material and Time). Including these 'covariates' in geostatistical prediction models often improves prediction accuracy. This can be done with a technique called 'regression kriging'.

We first load the rasters with environmental covariate data.

```
> dem <- readGDAL("geul_dem.txt")
> dist <- readGDAL("river_dist.txt")
> slope <- readGDAL("geul_slope.txt")
```

To analyse the relationship between Pb and covariates, the data for elevation, distance to the river and slope at the observation locations need to be added to the geul object. This can be done using the function `over` of the package `sp`. For help type `?sp::over` (you must write it like this because there are more packages having a function "over").

```
> # add explanatory data to geul object and inspect correlations
> geul$elev <- over(geul, dem)[[1]] # we need data from data frame
> geul$riverdist <- over(geul, dist)$band1 # alternative way
> geul$slope <- over(geul, slope)$band1
>
> # exploratory scatterplots
> plot(geul$elev, geul$pb)
> plot(geul$riverdist, geul$pb)
> plot(geul$slope, geul$pb)
> # correlation coeffcients
> cor(geul@data)
```

Q13 : *Which environmental covariate has the strongest correlation with Pb? Is the correlation positive or negative? What does this mean?* *Jump to A13* •

It appears that there is a relationship between Pb and the auxiliary variables, although it is not strong. To benefit from the covariates in kriging we fit a regression model that predicts Pb from the three covariates and calculate the semivariogram of the residuals of the regression. If the semivariogram of the residuals has a smaller sill and/or nugget, then this implies that the kriging variance will be smaller. In that case part of the spatial variation in Pb is explained by the covariates and taking advantage of this improves prediction accuracy. To explore this line of thinking you will now fit the regression model, calculate the regression residuals and calculate and fit a semivariogram to these.

```
> # fit a linear regression model and inspect the results
> geul.lm <- lm(pb~elev+riverdist+slope, data = geul)
> summary(geul.lm)
```

Not all covariates are significant predictors. We refit the model with the least significant covariate removed.

```
> # refit the model
> geul.lm <- lm(pb~elev+riverdist, data = geul)
> summary(geul.lm)
> str(geul.lm)
```

Since we had only three covariates to begin with we could easily remove the least significant covariate ourselves and fit the simplified model. If there were many more covariates this would become a tedious procedure and instead we might opt for an automated model selection procedure, such as a stepwise regression approach using the `stepAIC` function of the `MASS`

package.

Q14 : *How much of the variance in Pb is explained by the regression model?* *Jump to A14* •

The output of the `lm` function has an attribute named 'residuals'. This can be added to the geul object using:

```
> # append residuals to geul dataset
> geul$residuals <- geul.lm$residuals
```

Once geul has a new attribute (you may check this with: `names(geul)` or `summary(geul)`) the semivariogram estimation and fitting procedures can be applied to the residual.

```
> names(geul)
> summary(geul)
> # define gstat object and compute experimental semivariogram
> gpb2 <- gstat(formula = residuals~1, data = geul)
> vgpb2 <- variogram(gpb2)
> plot(vgpb2, plot.nu=FALSE)
> # define initial semivariogram model
> vgmpb2 <- vgm(nugget = 5000, psill = 15000, range = 200, model = "Exp")
> plot(vgpb2, vgmpb2)
> # fit semivariogram model
> vgmpb2 <- fit.variogram(vgpb2, vgmpb2, fit.method=7)
> plot(vgpb2, vgmpb2)
> vgmpb2
```

Q15 : *Does the semivariogram of the regression residuals have a smaller nugget and/or sill than that of the topsoil lead concentration? If so, which of the explanatory variables explains more of the lead spatial variation: elevation or distance to the river? Is the reduction, if any, in agreement with the results of regression analysis?* *Jump to A15* •

Check if the residuals approximately follow a normal distribution:

```
> # check if the residuals approximately follow a normal distribution
> hist(geul$residuals) # here they do so there is no need for a transformation
```

We can now run regression kriging in three steps:

1. Run linear regression using the covariate grids as input to create a trend map;
2. Krige the residuals;
3. Add regression predictions and kriged residuals to obtain the final predictions.

In preparation for the first step, we compile the covariate layers into one object and convert it to a data.frame to predict the trend.

```
> # convert to data.frame to predict the trend
> mask$elev <- dem$band1
```

```
> mask$riverdist <- dist$band1
> mask.df <- as.data.frame(mask)
```

Next we make predictions with the linear regression model.

```
> # regression prediction
> geul.trend <- predict(geul.lm, newdata = mask.df)
```

Predictions outside the mask are set to NA:

```
> # set predictions outside mask to NA
> geul.trend <- ifelse(test = is.na(mask$band1), yes = NA, no = geul.trend)
```

Check the predicted values:

```
> # check predictions
> summary(geul.trend)
> hist(geul.trend)
```

At some prediction locations we have predicted a Pb concentration smaller than 0 ppm. This is the result of using a linear model as a trend model, which is unbounded. The negative predictions are the result of the negative correlation with distance to river in combination with large distance to river values. Here we use a pragmatic solution by setting predicted values smaller than zero to zero.

```
> # set predictions smaller than 0 to 0
> geul.trend <- ifelse(geul.trend<0, yes = 0, no = geul.trend)
```

The next step is to krige the residuals.

```
> # krige the residuals
> geul.rk <- krige(formula = residuals~1, locations = geul,  beta = 0,
+                  newdata = mask, model = vgmpb2)
```

Next we change the name of the field that holds the predicted values and append regression predictions to the spatial object:

```
> names(geul.rk)[1] <- "resid"
> geul.rk$trend <- geul.trend
```

Set the kriged residuals and variance outside the mask to NA:

```
> # set kriged residuals and variance outside mask to NA
> geul.rk$resid <- ifelse(test = is.na(mask$band1), yes = NA,
+                         no = geul.rk$resid)
> geul.rk$var1.var <- ifelse(test = is.na(mask$band1), yes = NA,
+                            no = geul.rk$var1.var)
```

Finally, we obtain the RK prediction map by summing the predicted trend and kriged residuals.

```
> # obtain RK prediction
> geul.rk$predict <- geul.rk$trend + geul.rk$resid
>
> # set predictions smaller than 0 to 0
> geul.rk$predict  <- ifelse(geul.rk$predict < 0, yes = 0,
+                            no = geul.rk$predict )
> # plot the RK predictions
```

```
> spplot(geul.rk, zcol = "predict", col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="RK Pb predictions [ppm]")
```

Compute the kriging standard deviation by taking the square root of the kriging variance and generate a plot.

```
> # compute the kriging standard deviation
> geul.rk$var1.sd <- sqrt(geul.rk$var1.var)
> spplot(geul.rk, zcol = "var1.sd", col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="st dev [ppm]",
+        sp.layout=list("sp.points",geul, pch=1, cex=2))
```

For comparison, plot again the ordinary kriging standard deviation.

```
> spplot(geul.krig, zcol = "var1.sd", col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="st dev [ppm]",
+        sp.layout=list("sp.points",geul, pch=1, cex=2))
```

Q16 : *What do you observe when you compare the ordinary and regression kriging standard deviation maps. Can you explain the difference?*
Jump to A16 •

9 Universal kriging

A more elegant and efficient way of doing regression kriging is to predict the trend and residuals simultaneously. This is what in `gstat` is referred to as 'universal kriging'.

```
> # universal kriging
> geul.uk <- krige(formula = pb ~ elev + riverdist, locations = geul,
+                  newdata = mask, model = vgmpb2)
> names(geul.uk)[1] <- "predict"
```

Set predictions smaller than zero to zero.

```
> geul.uk$predict  <- ifelse(geul.uk$predict <0, yes = 0,
+                            no = geul.uk$predict )
```

Set predictions and variances outside the mask to zero.

```
> geul.uk$predict <- ifelse(test = is.na(mask$band1), yes = NA,
+                           no = geul.uk$predict)
> geul.uk$var1.var <- ifelse(test = is.na(mask$band1), yes = NA,
+                            no = geul.uk$var1.var)
```

Plot the UK predictions.

```
> spplot(geul.uk, zcol = "predict", col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="UK Pb predictions [ppm]")
```

Compute and plot the kriging standard deviation.

```
> geul.uk$var1.sd <- sqrt(geul.uk$var1.var)
> spplot(geul.uk, zcol = "var1.sd", col.regions = bpy.colors(),
+        xlim=c(190200,191300), ylim=c(314300,315600),
+        main="UK st dev [ppm]",
+        sp.layout=list("sp.points",geul, pch=1, cex=2))
```

Summarize the RK and UK predictions and variances.

```
> # compare RK and UK (predictions and variances)
> summary(geul.uk$predict)
> summary(geul.rk$predict)
>
> summary(geul.uk$var1.var)
> summary(geul.rk$var1.var)
> # end of script;
```

Q17 : *Compare the RK and UK predictions and variances. What do you conclude? Why is the UK variance somewhat larger than the RK variance?* *Jump to A17* •

10 Answers

A1 :

`dim` *yields the dimensions of the data frame, that is the number of rows and columns;*

`names` *lists the names of the attributes stored in the dataframe;*

`summary` *provides summary statistics;*

`class` *reports the object type of the function argument.* *Return to Q1* •

A2 : *The distribution is not symmetric because it has the highest frequencies for low lead concentrations, at the left side of the distribution. It has a tail to the right side, although there are no obvious outliers. It may be useful to try a transformation of the data prior to the geostatistical analysis, in order to make the distribution more symmetric and more normal. Although we will not pursue this in this practical, you may wish to plot a histogram of the square root of the lead concentration using function* `sqrt()`*. Try it out!* *Return to Q2* •

A3 : *The object has changed its class type. First it was a* `data.frame`*, now it is a* `SpatialPointsDataFrame`*. This class is defined in the* `sp` *package, and signifies that all lead concentration values have a geographic coordinate. Observations* `x` *and* `y` *are no longer ordinary attributes, which you may check with* `names(geul)`. *Return to Q3* •

A4 : *There is a hotspot in the southwest corner of the study area. It is also immediately apparent that high lead concentrations occur mainly close to the river and that low concentrations are found further away from the river, where floods will be much less frequent. Notice also that quite a few observations are outside the*

study area. These observations are still useful because they may help characterize the model of spatial variation and can even aid the kriging interpolation, if they are not too far away from the study area boundary. *Return to Q4* •

A5 *: The* `formula` *argument specifies the model that relates the dependent variable to explanatory variables. For those familiar with linear regression in R (i.e. function* `lm`*) will recognize the notation. The dependent variable is listed first, followed by a tilde symbol (*~*) and next all explanatory variables, with a +, : or * symbol in between (try* `?formula` *if you want to know more). In this case we are using ordinary kriging, which assumes a constant trend. In other words, there are no explanatory variables. Therefore, the formula only includes an intercept, indicated by the number 1, i.e.* `formula = pb ~ 1`*. The* `data` *argument specifies the name of the dataset, which should be a* `SpatialPointsDataFrame`*. In fact, the data may also be just a* `dataframe`*, but then the gstat function requires the additional argument* `locations`*.* *Return to Q5* •

A6 *: It is definitely spatially correlated because the semivariances at small separation distances are smaller than those at large distances. The nugget is only about 20% of the sill and the range is about 500 m, which is about the width of the study area. The combination of a small nugget and large range indicate that there is substantial spatial correlation. When option* `plot.nu=TRUE` *is set, the number of point pairs used to calculate each point of the experimental semivariogram is shown.* *Return to Q6* •

A7 *: Visual inspection of the joint plot of the sample semivariogram and semivariogram model indicates that the exponential model fits the sample semivariogram better than the spherical model, although this is a subjective judgement that not everybody may agree with. Potentially the Matern model (use* `model="Mat"`*) can do even better because it has an additional shape parameter. For those of you who do not like subjective evaluations an objective criterion may be used to choose between models: issue* `attr(vgmpb, "SSErr")` *on the command line. This gives the Sum of Squared Errors between sample and fitted semivariogram. It can be used to compare different semivariogram shapes, but only when the same lags and fit.method are used.*

All fitted semivariogram models will be quite similar because they must pass through the sample semivariogram values, but indeed the effect on the nugget is quite strong: the fitted spherical semivariogram has a much larger nugget than the fitted exponential semivariogram. Perhaps the nugget of the latter is unrealistically small, given that there will also be measurement errors and lead concentration may have short-distance variation. More insight into the nugget variance may be obtained using smaller lag sizes at short distances, e.g. by issuing `vgpb <- variogram(gpb, boundaries = c(20,50,100,200,500,1000))`*. This indicates that the nugget is about 3000. However, note that the evidence for this is weak because there are only a few point pairs for the shortest lag.* *Return to Q7* •

A8 *: The object has two attributes, named* `var1.pred` *and* `var1.var`*. The first is the map of the kriging predictions, the second that of the kriging variances.* *Return*

to Q8 •

A9 : *High lead concentration predictions are near the river, and in particular in the south-west hotspot. Low concentrations are predicted further away from the river, uphill, at the border of the study area. This is in agreement with what we deduced from the point observation concentrations. Note, however, that the kriging prediction map produces a smoothed version of reality, much smoother than the true spatial distribution of the lead concentration.* *Return to Q9* •

A10 : *The spatial pattern is very different now. High values are obtained far away from observation locations, low values close to observation locations. Indeed you can see from the kriging standard deviation map that observations were taken along transects laid out across the study area. The kriging standard deviation is particularly large at the boundary of the study area. This signifies that spatial extrapolation is more difficult than spatial interpolation.The minimum sd is 78.5, the maximum sd 160.6. Given that the Pb predictions range from 6.0 to 630.0 we must conclude that the ordinary kriging prediction map is not that accurate.* *Jump to A17* •

Return to Q10 •

A11 : *In such case, when the semivariogram is not specified, kriging cannot be used. However, gstat still produces an interpolated map, by using an interpolation method that does not require a semivariogram. This method is inverse-distance weighted interpolation (IDW). The resulting map is overall similar to the kriging prediction map, although there are also meaningful differences. Most noteworthy is that IDW creates 'islands' around observation locations, because near observations it relies almost entirely on the nearest observation. Kriging does not do so, at least not when the nugget is non-zero. This is because kriging 'knows' that there are measurement errors and short-distance spatial variation, indicating that one should not put all trust in the nearby observation.* *Return to Q11* •

A12 : *The simulated maps are much more noisy than the kriged map, because these maps do not smooth the reality. Each of them could be the true reality, it is just that we do not know which one it is. Thus, the differences between the simulated maps represent the uncertainty about the lead concentration at any one prediction location (grid cell). The average of a large number of simulated maps yields the kriging prediction map, while the standard deviations of the simulated maps would be equal to the kriging standard deviation map. The practical use of simulated maps will be addressed in the 'uncertanty' module on Wednesday.* *Return to Q12* •

A13 : *The slope and distance to river show the strongest correlation with Pb content. For both variables there is a negative correlation with Pb. This means the further from the river the lower the Pb content, which makes sense since the river is the source of the pollution.* *Return to Q13* •

A14 : *The trend model explains 22% of the variance in Pb data.* *Return to Q14* •

A15 : *The sill of the semivariogram of the regression residuals is lower than the sill of the semivariogram of the topsoil lead content. The regression explains some of the variation in the Pb data, hence the variance of the residuals (var(geul$residuals)) is smaller than the variance of the Pb concentration (var(geul$pb). This results in a smaller sill. In fact, the percentual reduction of the sill should be more or less equal to the percentage of variance explained by the regression model (22%).* *Return to Q15* •

A16 : *The RK standard deviation is smaller than the OK standard deviation. The RK predictions are closer to reality than the OK predictions because some of the variation in the Pb data was explained by the RK trend model. We use additional information and hence can reduce uncertainty and get a more accurate map.* *Return to Q16* •

A17 : *The maps are pretty similar, although the universal kriging variance is greater than the regression kriging vartiance. This is because regression krigingignores the uncertainty associated with the trend model prediction (i.e., the regression coeffcients), while it is accounted for in universal kriging. Note: some geostatisticians use the term 'kriging with external drift' instead of 'universal kriging'. In their view, the term 'universal kriging' may only be used when the covariates are (polynomials) of the geographic coordinates.* *Return to Q17* •