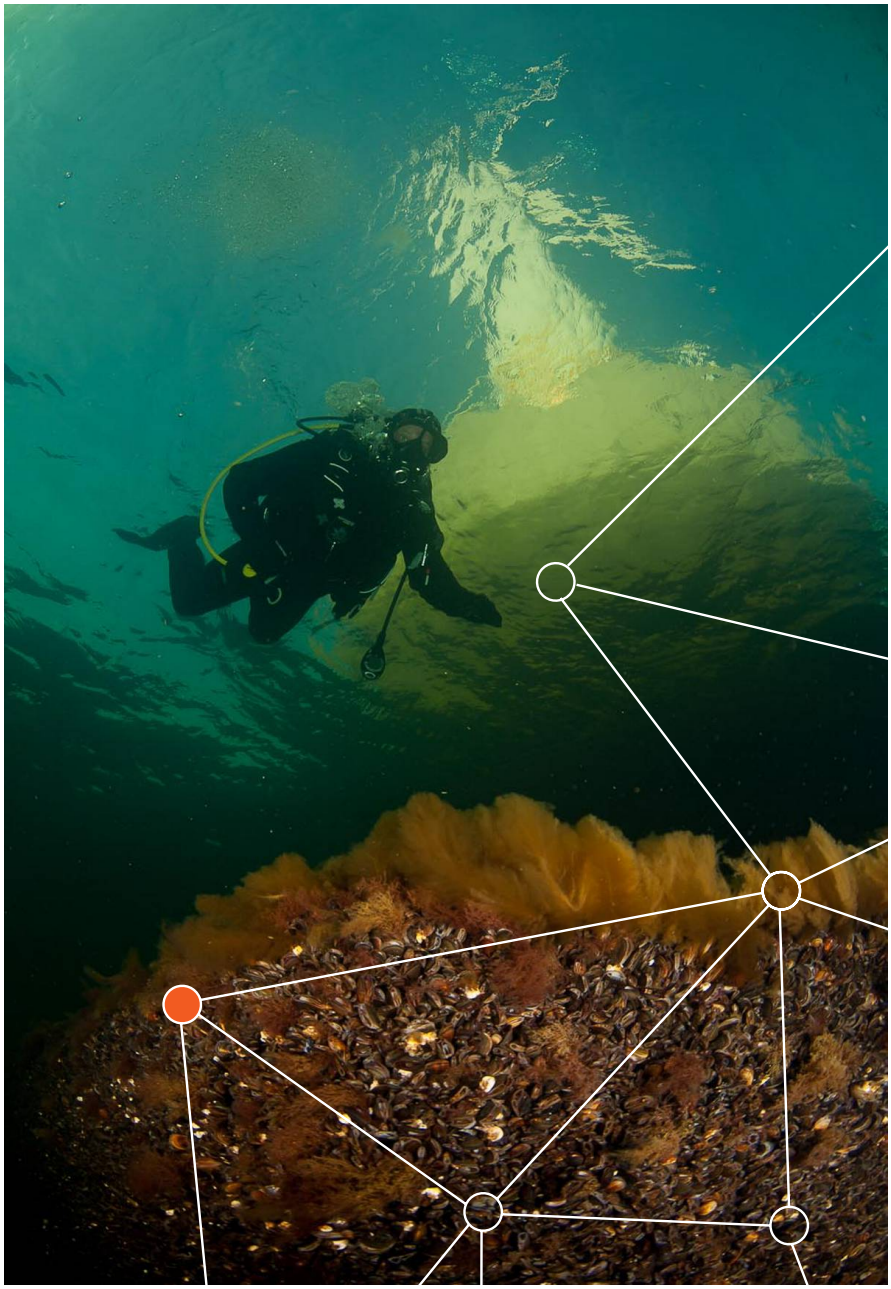
#338 DECEMBER 2017

Handbook of Geostatistics in R for Fisheries and Marine Ecology

ICES COOPERATIVE RESEARCH REPORT

RAPPORT
DES RECHERCHES
COLLECTIVES



ICES INTERNATIONAL COUNCIL FOR THE EXPLORATION OF THE SEA
CIEM CONSEIL INTERNATIONAL POUR L'EXPLORATION DE LA MER

ICES COOPERATIVE RESEARCH REPORT
RAPPORT DES RECHERCHES COLLECTIVES

NO. 338

DECEMBER 2017

Handbook of Geostatistics in R for Fisheries and Marine Ecology

Pierre Petitgas • Mathieu Woillez
Jacques Rivoirard • Didier Renard • Nicolas Bez

ICES International Council for the Exploration of the Sea
CIEM Conseil International pour l'Exploration de la Mer

International Council for the Exploration of the Sea
Conseil International pour l'Exploration de la Mer

H. C. Andersens Boulevard 44–46
DK-1553 Copenhagen V
Denmark
Telephone (+45) 33 38 67 00
Telefax (+45) 33 93 42 15
www.ices.dk
info@ices.dk

Recommended format for purposes of citation:

Petitgas, P., Woillez, M., Rivoirard, J., Renard, D., and Bez, N. 2017. Handbook of geostatistics in R for fisheries and marine ecology. ICES Cooperative Research Report No. 338. 177 pp.

Series Editor: Emory D. Anderson

DOI: http://doi.org/10.17895/ices.pub.3717

ISBN 978-87-7482-209-7
ISSN 1017-6195

Contents

Foreword

The course in Fontainebleau was a great experience which provided helpful technical support for my work on the variation of spatial distributions of fish and increased my repertoire of methods. In particular, I am now able to deal with a huge amount of trawl survey station data, analyse the structure in their spatial and temporal variability, and derive maps from sample datapoints. In addition, I also see how useful it can be for other hot topics such as plastics in the ocean or impacts of oil rigs, especially during a time when environmental issues receive increasing public attention. During the course, I was also able to socialize and meet a variety of interesting, smart, and friendly scientists from all over the world. Courses such as the one in Fontainebleau are like the cherry on a scientist's cake. It is about working together, which is more important in science than in every other business, and it is about long nights in a Scottish pub in the heart of France with German beer, English gin, and Chilean jokes.

Karl-Michael Werner

University of Bergen student and participant in the 2014 ICES training course on geostatistics

1 Introduction

Fisheries surveys to estimate the abundance of populations have become a pillar in providing fishery-independent data to determine the status of fish stocks and monitor ecosystems. Since the early 1990s, geostatistics has been used for designing sampling at sea and estimating the precision of estimates of global population biomass or abundance (ICES, 1993; Rivoirard *et al*., 2000). Now, the ecosystem approach to fisheries management calls for methods that deal explicitly with spatial issues. In effect, the spatial management of human activities and/or the conservation of particular habitats require precise distribution maps of resources at various stages in their life cycle. Geostatistics offers a range of solutions for mapping and characterizing different aspects of spatial distributions. On more ecological grounds, geostatistics is also useful for modeling habitats and understanding the ecology of spatial distributions.

The varied range of geostatistical methods is largely based on the theory of random functions and random fields. The cornerstone of the geostatistical approach to applying this statistical framework for mapping lies in the so-called structural analysis, where the spatial (or spatio-temporal) correlation structure in the data is analyzed and modeled by a so-called variogram. Model types (e.g. power, exponential, spherical) are chosen based on their underlying physical and mathematical properties relative to the spatial process to be modelled (Matheron, 1989). Once the model type is chosen, it is best fitted to the data using standard statistical fitting procedures. The model is then used for interpolating the data on a grid, which results in a map of the variable studied (local and global estimation) and a map of the estimation error (precision of the estimation). It is worth noting that being model-based, the estimation variance calculated by geostatistics applies to any sampling design and particularly to regular designs, in which sample point locations are spatially correlated. This frees the practicioner from using random designs only to compute design-based statistics, as random designs may provide lower precision than regular designs. Further, geostatistics and classical statistics correspond to different approaches when using the same statistical framework of random functions (Matheron, 1989). In particular, geostatistics estimates regional quantities (mean value of the process over a domain) while classical statistics focusses on estimating the process mean. In addition, classical statistics computes the variance of the estimate, while geostatistics also develops the variance of the estimation error (ICES, 1993; Petitgas, 2001). Depending on the spatial model, sampling intensity, and size of the domain, the estimates may or may not differ, which justifies differentiating between the two approaches (Matheron, 1989). The objective of this handbook is to summarize and explain the basic notions on the wide range of geostatistical methods (linear, multivariate, non-linear, simulations) that are useful for mapping in the context of the ecosystem approach and offer to the reader illustrative case studies with code in R language.

Global estimation of population abundance (or biomass) with its precision for different survey designs (even systematic design) is a key issue in fisheries science for which geostatistics provides solutions given a variogram model (Petitgas, 2001; Bez, 2002). This is explained in chapters 4 and 5 on variography and variances. This latter chapter discusses the relationship between structure and scale. Further, when the variable to estimate is a non-linear combination of primary parameters that are those sampled, simulations may be required, as is explained in Chapter 9 on simulations.

Variation in spatial distributions with population abundance and/or environmental factors is another key issue. The many aspects of spatial distributions can be characterized by spatial indicators and monitored over time (Bez and Rivoirard, 2001; Woillez *et al.*, 2007, 2009a). Chapter 3 is dedicated to spatial indicators.

Mapping resources and habitats is clearly paramount. The geostatistical solution to mapping is kriging, which constructs local unbiased estimates of minimum variance. For that, one assumes an underlying random function and its variogram model. The various types of kriging and interpolation settings (Chilès and Delfiner, 2012) are presented in Chapter 6.

Mapping habitats may be more complex than kriging fish concentrations. One may be interested in thresholding the data to consider the prevalence in species occurrence or hotspots. Or one may be interested in incorporating in the mapping particular relationships with environmental parameters, some of which may be qualitative variables. Thus, multivariate kriging and non-linear approaches using thresholds (Rivoirard, 1994; Chilès and Delfiner, 2012) are developed in chapters 7 and 8.

The applications of a wide range of geostatistical tools are expected to increase with the development of the package RGeostats (Renard *et al.*, 2016), which is now freely available for the R language environment. This handbook is intended to summarize the principles of geostatistics and provide to the reader the capability to apply the methods using demonstration scripts in the R language. It compiles the materials of the 2013 and 2014 ICES training courses held by the authors in Fontainebleau. The handbook is constructed from lecture notes presenting the theoretical background with illustrative fisheries survey data studies. The annexes detail the practice in applying the methods. The R package RGeostats is presented in Annex 1. Example data sets used throughout the document are presented in Annex 2. Demonstration Rscripts are provided in Annex 3. Each script allows the user to perform a particular geostatistical study on an example dataset. Each script can be copy/pasted in the R environment for demonstration. The examples illustrating the theory are taken from the Rscripts provided in Annex 3.

2 Basic notions

Geostatistics is a set of models and methods that are designed to study variables which are distributed in space (or possibly space-time). Such variables possess both a structured and a random aspect and cannot be simply described by a regular function of the coordinates. Such a variable was coined a regionalized variable denoted as $z(x)$; that is, the variable z at location x (considered as punctual or quasi-punctual, x being a short notation for the 2D (x,y) or 3D (x,y,z) coordinates).

Examples of such regionalized variables are: (i) bottom depth at 2D point, (ii) fish density in 2D, or (iii) concentration in 3D (number or weight of fish per unit 2D area or 3D volume). Occasionally, x can represent a point in 1D, e.g. the transect biomass obtained by summing fish densities along transects with a given direction. In most cases throughout this handbook, the target variable will be the 2D fish density of a spatial population, reflecting the majority of sampling tools that are used in fisheries surveys (e.g. echosounders, trawls, images, video). Then, the abundance Q over a region V is the sum of the fish density over this region:

$$Q = \int_V z(x)\mathrm{d}x$$

and the mean fish density over this region is:

$$z(V) = \frac{Q}{V} = \frac{1}{V}\int_V z(x)\mathrm{d}x$$

Such variables are usually not known everywhere in space. Data may be available at isolated datapoints (e.g. sampling stations for trawl surveys), along transects (e.g. acoustic or video surveys), or, for example, over a gridded map (satellite data).

New variables obtained by transforming original ones can also be considered. For example, the indicator of presence of fish: regionalized variable equal to 0 where the fish density is 0 and equal to 1 otherwise. Or the logarithm of a non-zero concentration to better describe a histogram and reduce the influence of the largest values (care should be taken, however, when using such non-linear transformations: back-transforming statistics are not straightforward, for example, the antilog of the mean of the logarithm is not the mean of the variable).

Basic statistics and visualization are very helpful at the beginning of a data analysis, particularly to make the distinction between different statistical populations or to detect outliers or extreme values. Appropriate tools include the histogram of a variable, the scatter plot between two variables, and basemaps overlaid with bubbles of area/size proportional to the value of a third variable. Relevant statistics are:

- the arithmetic mean of values: $m = \frac{1}{n}\sum_1^n z_i$;
- their variance: $\sigma^2 = \frac{1}{n}\sum_1^n (z_i - m)^2 = \frac{1}{n}\sum_1^n z_i^2 - m^2$, which measures the dispersion around the mean, in squared units (when the mean is 0, the variance is the mean of the squared values);
- the standard deviation σ (square root of variance $\sigma = \sqrt{\sigma^2}$), measuring the dispersion around the mean (in the unit of the variable);
- for a positive (and possibly null) variable, the coefficient of variation CV = σ/m, measuring the relative dispersion around the mean;

- the correlation coefficient between two variables: $\rho = \frac{C_{12}}{\sigma_1 \sigma_2}$

  where $C_{12} = \frac{1}{n}\sum_{1}^{n}(z_{1i} - m_1)(z_{2i} - m_2)$ is the covariance between the two variables, which lies between –1 and +1 and measures the linear dependence between the two variables.

In many software packages (including R), variances (and covariances) are computed by dividing by $n - 1$, not n. Division by n corresponds to the variance with respect to the observed mean (empirical variance). Division by $n - 1$ corresponds to the estimation of the variance of a theoretical probability distribution with unknown mean, when supposing that the data are independent outcomes of this distribution. Because of spatial correlations among the samples, division by $n - 1$ is not appropriate in geostatistics.

2.1 Support and additivity

The variables may be measured or considered at punctual or quasi-punctual locations (e.g. trawl stations), but also on user-defined segments along a line (e.g. acoustic transects), on blocks (e.g. ICES rectangles), or on any kind of geographical domains. This corresponds to the support on which the variable is considered. The same variable, considered at different supports, will have different statistics, notably in terms of variability (see the example of acoustic "pings" in Rivoirard *et al.*, 2000).

A variable is "additive" if its mean over a set of points (e.g. a block) equals its arithmetic mean (what are considered as points being in reality small and equal units). For example, a bottom depth or a fish density are additive. On the contrary, the mean length of fish (e.g. measured at trawl stations) is not additive; the mean length over several stations is not the arithmetic mean of the mean length, as it has to be weighted by the number of individuals (supposed to be counted using the same trawled area). The same can be said for proportions at age.

2.2 Referencing

To compute distances, longitude and latitude have to be converted into 2D coordinates of absolute distance (km or nautical miles). For short distances at medium–low latitudes, a simple projection x = 60 × longitude × cos(average latitude): y = 60 × latitude can be used to convert to nautical miles, which is often used in marine applications because of the fact that 1 nautical mile is equivalent to 1° of latitude anywhere in the world (see Application 2.1). This simple projection presents the advantage of transforming a regularly gridded set of points into another gridded set of points.

Application 2.1. Change of reference system

A bottom-trawl survey is represented with red circles proportional to trawled hake (*Merluccius merluccius*) densitites in Figure 2.1. The axes are labeled in longitude and latitude.

```
# pre-requisite
projec.toggle(0)
rg.load(filename="Demo.hake.bob.db.data",objname="db.data")

# Display in Longitute/Latitude system (left)
plot(db.data,title="",xlab="Longitude",ylab="Latitude",asp=1/cos(45*p
i/180))
map("worldHires",add=T,fill=T,col=8)
```

```
# Display in transformed reference (Right)
projec.define(projection="mean",db=db.data)
plot(db.data,title="",xlab="Nautical mile",ylab="Nautical mile",asp=1
)
```

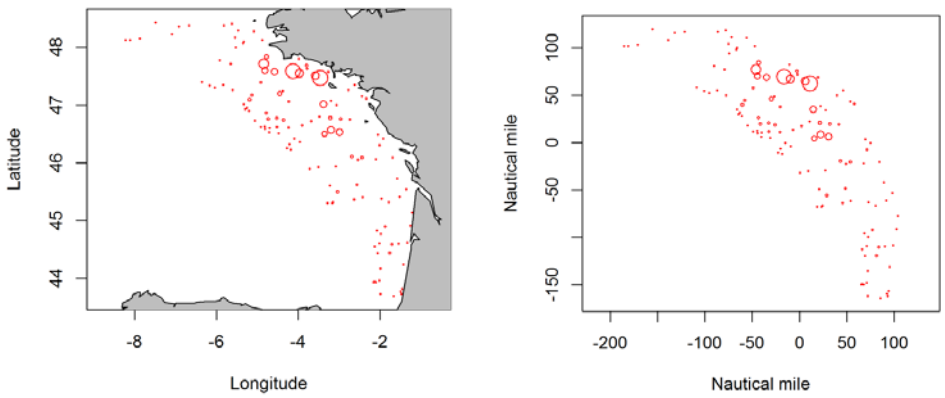


Figure 2.1. Example of cosine coordinate transformation. Left: geographical space; right: after transformation (Hake bottom-trawl survey, Ifremer, France).

If the latitude is too high or varies too much, a more complex projection may be preferable and can be found in dedicated R libraries. Points that are regularly spaced in the geographical space will not be so after projection, and vice versa; this may be important when computing and representing map values.

Sometimes a natural coordinate system may be used to better follow natural spatial continuities (e.g. distance along and off shelf edge). One possible way of doing this consists of an *ad hoc* transformation, projecting points on a reference line either predefined (e.g. coastline) or user defined (see Application 2.2).

Application 2.2. Change of reference system

An acoustic survey track is represented in black in Figure 2.2. The reference line is defined by a set of ordered points and the segments with directions U_i (i = 1, 2, ...) that join them (in red). Each datapoint is projected on one segment of the reference line.

```
# pre-requisite
projec.toggle(0)
rg.load(filename="Demo.Nansen.db.data", objname="db.data")
rg.load(filename="Demo.Nansen.polyline", objname="polyline")

# Display in Longitute/Latitude system (left)
plot(db.data,pch=20,col="black",name.post=1,cex=0.1,asp=1,
     title="",xlab="Longitude",ylab="Latitude")
lines(polyline,col="red",pch=2)
map("worldHires",add=T,fill=T,col=8)

# Display in transformed reference (Right)
db.data = db.unfold.polyline(db.data,polyline$x,polyline$y)
plot(db.data,pch=20,col="black",cex=0.1,title="")
abline(v=0,col="red")
```

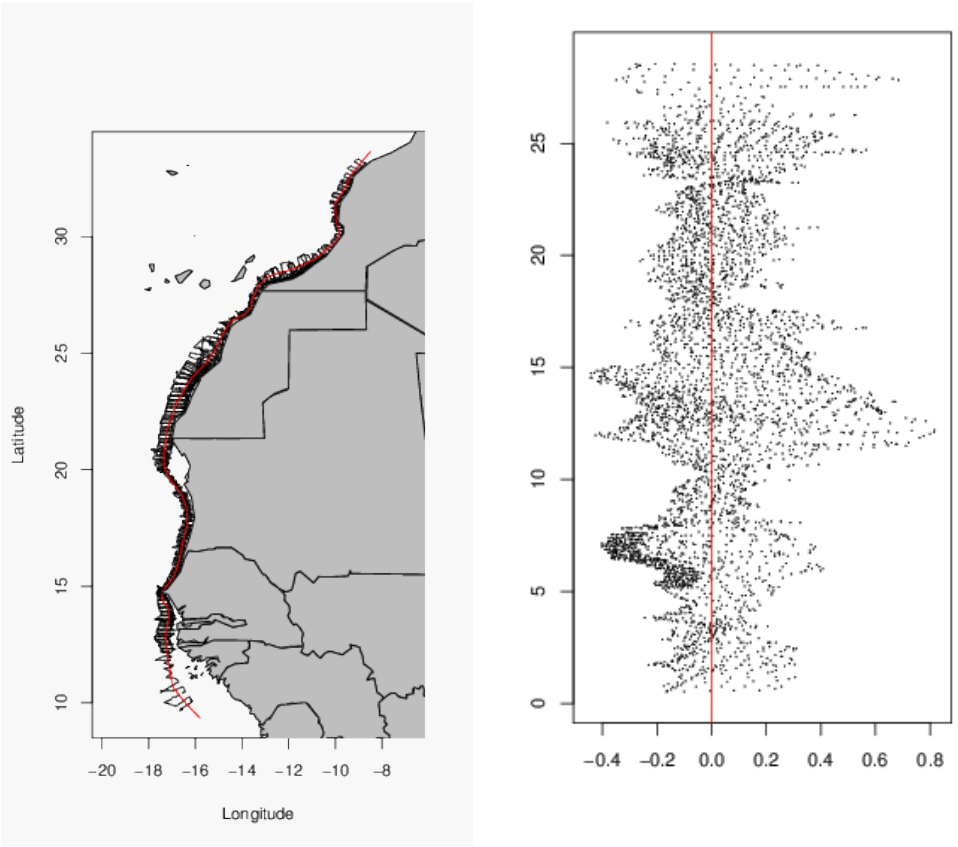


Figure 2.2. Example of *ad hoc* coordinate transformation. Left: geographical space; right: after transformation (Nansen acoustic survey datapoints).

2.3 Domain

The question of the area or "domain" to be considered is important because many statistical results will depend on this. In many cases, there is no problem in defining the domain to be considered. For a fish spatial population, the domain may be the domain of presence of fish, with fixed boundaries [e.g. Norwegian herring (*Clupea harengus*) in fjords in winter]. Often, a species may occupy part of the sampled domain and present diffuse limits, such that it may not be easy to delineate the limits. In such a case, it may be difficult to decide which datapoints corresponding to the numerous zero values should be included in the analysis.

There is another point related to this issue. While many statistical tools depend on the domain considered, some other methods do not depend on the delineation of such domains and on the inclusion or not of zero fish density values when studying spatial populations. The basic statistics seen above (mean, variance, and so on) clearly depend on the domain and on the possible zeros in it. In geostatistics, the current intrinsic approach with the variogram (see Chapter 4 on variography) also depends on the domain. The domain is considered as a window, within which we study the behavior of the regionalized variable which could be thought to extend outside. Another geostatistical approach, the transitive one, does not require the delineation of a domain for a spatial population providing that sampling extends beyond its limits. The contribution of zero fish density values is zero, just like for global abundance. In the next chapter, several spatial indices are presented that have the same advantage and allow following spatial populations in time.

3 Indices of spatial distributions

3.1 Context

Survey data obtained from monitoring exploited populations provides opportunities for ecological investigations of relationships between spatial pattern and population dynamics (MacCall, 1990). Spatial indicators (not to be confused with indicator variables) are statistics that aim to describe and summarize the spatial distribution of populations (in terms of fish density, location, or possibly environmental variables such as depth). They are useful for investigating such relationships and making fishery-independent diagnostics by an indicator-based approach. They can be helpful in identifying how spatial distributions of fish populations vary with density-dependence or climate, as highlighted by Petitgas (1998).

3.2 Theoretical framework

The spatial indicators selected here are statistics typically made on fish density values, and care was taken to select statistics that would depend not on the inclusion or exclusion of zero values (see "Domain" in Chapter 2 on "Basic Notions").

A list of several spatial indicators (Table 3.1) characterizes the location (centre of gravity and spatial patches), occupation of space (inertia, isotropy, positive area, spreading area, and equivalent area), fine-scale structure (microstructure), and overlap between populations (global index of collocation). The list does not, of course, intend to be either fixed or exhaustive.

Table 3.1. List of spatial indicators documented and population characteristics to which they are related.

Indicator	Abbrev.	Units or range	Population characteristics
Centre of gravity	CG	Geographical coordinates	Mean geographic location of population
Inertia	I	Nautical miles^2	Dispersion of population around its centre of gravity
Anisotropy	An	≥1	Elongation of spatial distribution of population
Isotropy	Is	[0, 1]	Elongation of spatial distribution of population
Global index of collocation	GIC	[0, 1]	Overlap of two spatial populations
Local index of collocation	LIC	[0, 1]	Collocated occurence of two spatial populations
Number of spatial patches	NP	>0	Patchiness
Positive area	PA	Nautical miles^2	Area of presence occupied by stock, even with a low density
Spreading area	SA	Nautical miles^2	Measure of area occupied by stock that takes into account variations in fish density.
Equivalent area	EA	Nautical miles^2	An individual-based measure of area occupied by stock

Microstructure index	M	[0, 1]	Fine-scale variability of fish density surface

3.3 Preliminary: total abundance

Let x be a point in two-dimensional space [short for the usual two-dimension notation (x, y)] and $z(x)$ be the population density at location x within a region V. As mentioned earlier, the total abundance of the population in this region is:

$$Q = \int_V z(x)\mathrm{d}x$$

Note that the number of individuals at location x is proportional to $z(x)$, so that the probability density function of the location $\underline{x}$ of a random individual is $z(x)/Q$.

In practice, Q can be estimated from the data through a discrete summation over sample locations x_i $(i = 1, \ldots, N)$. In the case of irregular sampling, areas of influence around samples (determined in the projected space) can be used as weighting factors. Thus, from sample values $z_i = z(x_i)$ with areas of influence s_i, we have the following estimate:

$$\sum_{i=1}^{N} s_i z_i$$

This corresponds to the mean fish density $z(V) = Q/V$ over this region being estimated as $\frac{\sum_{i=1}^{N} s_i z_i}{\sum_{i=1}^{N} s_i}$.

Application 3.1. Areas of influence, total abundance, and mean fish density

Area of influence of a sample location is defined as the area made up of the points in space that are closer to this sample than to others. It can be evaluated by overlying a very fine regular grid and counting grid points closer to the sample. Known or supposed boundaries (e.g. land, a limit distance of influence from a sample location) of the sampled population may be used.

The following R code computes and plots these areas of influence using the function infl() (Figure 3.1). They are computed from the sample locations of a bottom-trawl survey carried out in the Bay of Biscay for a given year. The function is run after a projection has been defined.

```
# pre-requisite
rg.load(filename="Demo.hake.bob.db.data",objname="db.data")
rg.load(filename="Demo.hake.bob.poly.data",objname="poly.data")
projec.define(projection="mean",db=db.data)

# Compute areas of influence of survey samples
db.data <- db.delete(db=db.data,names=6)
db.data <- infl(db.data,nodes=c(400,400),origin=c(-11,43),extend=c(11
,7),
                dmax=100,polygon=poly.data,plot=T,asp=1)
```

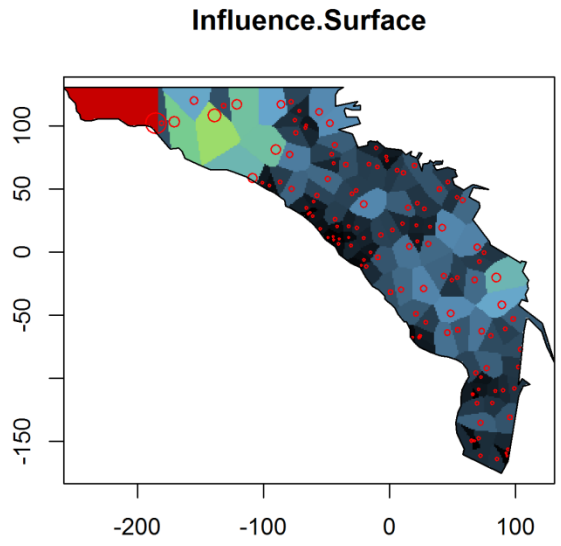


Figure 3.1. Map of areas of influence from the sample locations of a bottom-trawl survey in the Bay of Biscay. Areas of influence are coloured according to their values, using a colour palette ranging from black (small) to red (large). Red circles proportional to areas of influence are also represented at the trawl stations.

The areas of influence can then be used as weighting factors when computing total abundance, spatial indices, and mean fish density over the survey domain. This is illustrated using the age 0 hake density data that were monitored during this bottom-trawl survey. The following R code details the computation of the total abundance extracting the areas of influence and age 0 hake densities from the database. Total abundance can also be obtained directly using the function SI.stats(), as well as some areas indices (see next chapter). The computation of the mean age 0 hake density over the survey domain is then acheivable, as well as the distribution and the mean of a spatially distributed population parameter (not illustrated). Indeed, one can be interested in computing, for example, the depth distribution occupied by the population or any other parameters (e.g. temperature, sediment type.

```
# Extract areas of influence and age 0 hake densities
si <- db.extract(db.data,"Influence.Surface") # in nautical mile²
zi <- db.extract(db.data,"A0")                # in nb/nautical mile²

# Compute total abundance (i.e. number of individuals) in two ways
sum(zi*si)
## [1] 69868442

SI.stats(db.data)
## $totab
## [1] 69868442
##
## $parea
## [1] 22915.34
##
## $eqarea
## [1] 4775.287
##
## $sparea
## [1] 5671.599
```

```
# Compute the mean hake density over the survey domain
sum(zi*si)/sum(si)
## [1] 2026.287
```

3.4 Positive area, equivalent area, and spreading area

The positive area (PA) is the measure, in nautical miles^2, of the space occupied by fish densities strictly above zero (Woillez *et al.*, 2007, 2009a). It is estimated from data as the sum of the areas of influence around samples where there are positive fish densities (Figure 3.2):

$$PA = \sum_{i=1}^{N} s_i 1_{z_i>0}$$

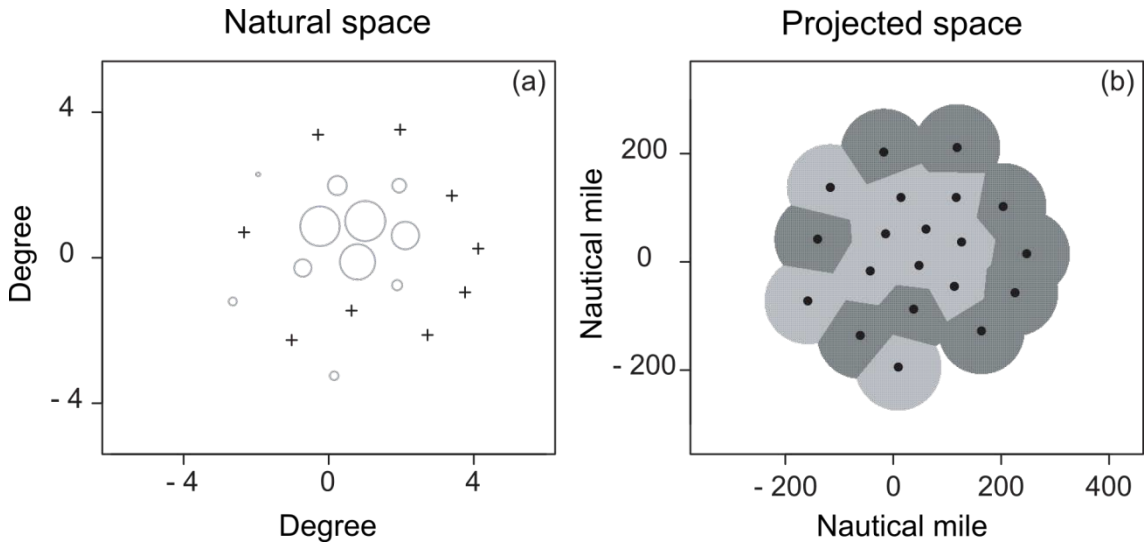


Figure 3.2. Bubbleplot of the sample values (a) and corresponding positive area (b) shaded in light grey (with a limit to the area of influence of each sample). The projection used multiple longitudes by 60 × cosine of the mean latitude (here 0°) and latitudes by 60.

The equivalent area represents the area, in nautical miles^2, that would be covered by the population if all individuals had the same density, equal to the mean density per individual (Bez and Rivoirard, 2001; Woillez *et al.*, 2009a):

$$EA = \frac{Q}{\int z(x)\frac{z(x)}{Q}dx} = \frac{Q^2}{\int z(x)^2 dx}$$

Practically, in the discrete case with sample values z_i and areas of influence s_i, this can be written as:

$$EA = \frac{(\sum_{i=1}^{N} s_i z_i)^2}{\sum_{i=1}^{N} s_i {z_i}^2}$$

The EA ranges from 0 to the positive area. It would be equal to the positive area if all strictly positive values of density were the same. The equivalent area can be related to the area occupied by the positive fish density values PA and their coefficient of variation CV_0 through $\frac{PA}{EA} = 1 + {CV_0}^2$ (Woillez *et al.*, 2007, 2009a).

The spreading area (SA) is a measure, in nautical miles^2, of how the population is distributed in space, taking into account the variations in fish density (Woillez *et al.*, 2007, 2009a). Let T be the cumulative area occupied by the density values, ranked in decreasing order, $Q(T)$ be the corresponding cumulative abundance, and Q be the overall abundance. The SA (expressed in nautical miles^2) is then simply defined as twice the area below the curve expressing $(Q - Q(T))/Q$ as a function of T (Figure 3.3):

$$SA = 2\int \frac{Q - Q(T)}{Q} dT$$

As $[Q - Q(T)]/Q$ decreases from 1 to 0 and is convex, SA is smaller than the positive area. It equals the positive area when the population is evenly spread with a constant density. The curve in Figure 3.3 is a derivation of the Lorenz curve (Gini, 1921) representing the histogram of fish density values, but it has the advantage of receiving no contribution from zero density values. The spreading area can be related to the area occupied by the positive fish density values PA and their Gini index of dispersion G_0 through $\frac{SA}{PA} + G_0 = 1$ (Woillez *et al.*, 2007, 2009a).

Application 3.2. Area indices of hake

The following R code (full script in Annex 3) computes total abundance, positive area, equivalent area, and spreading area of age 0 hake densities spatial distributions. The option "flag.plot" in the function SI.stats() permits illustrating graphically the value of the spreading area, which is simply defined as twice the area below the curve.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.hake.bob.db.data","db.data")
projec.define(projection="mean",db=db.data)

# The next lines calculate total abundance, positive area, equivalent
area, and spreading area
par(xaxs="i",yaxs="i")
SI.stats(db.data,flag.plot=TRUE)

## $totab
## [1] 69915222
##
## $parea
## [1] 22895.45
##
## $eqarea
## [1] 4771.684
##
## $sparea
## [1] 5664.107
```

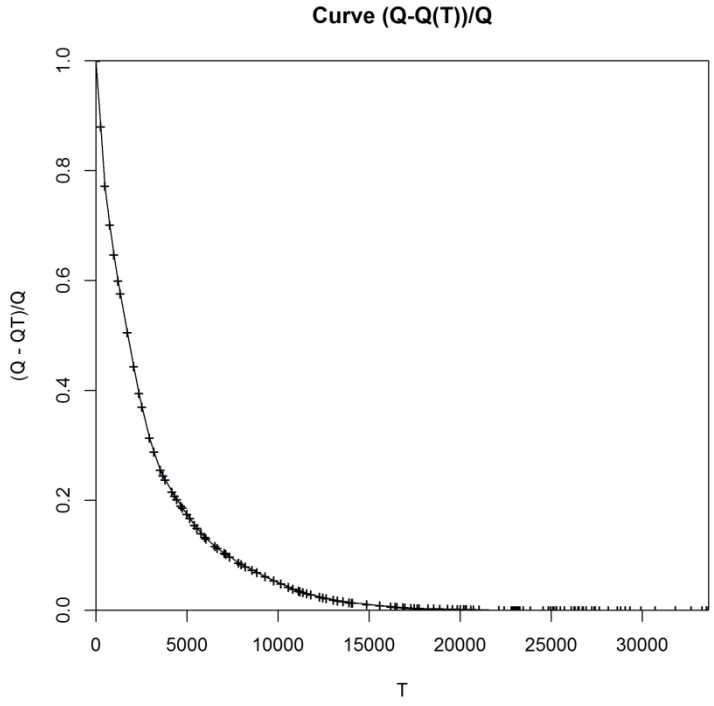


Figure 3.3. Age 0 hake densities in the Bay of Biscay. The curve represents the part of the abundance which is left when excluding the area T occupied by the highest fish densities. The spreading area (*SA*) is defined as twice the area below this curve.

3.5 Centre of gravity, inertia, and isotropy

The centre of gravity (CG) is the mean location of the population, that is, the mean of the location of the individuals that compose it (Bez and Rivoirard, 2001; Woillez *et al.*, 2009a). As the probability density function of the location $\underline{x}$ of a random individual is $z(x)/Q$, the centre of gravity (*CG*) is:

$$CG = E(\underline{x}) = \int x \frac{z(x)}{Q} \mathrm{d}x$$

Similar to the abundance, this statistic is estimated from the data through a discrete summation over sample locations. From sample values z_i at locations x_i, with areas of influence s_i, we have, for example:

$$CG = \frac{\sum_{i=1}^{N} x_i s_i z_i}{\sum_{i=1}^{N} s_i z_i}$$

The inertia is the variance of the location of individuals in the population, that is, the mean square distance between an individual fish and the centre of gravity of the population.

Inertia describes the dispersion of the population around its centre of gravity. With the notations used for *CG*, the inertia (*I*) is:

$$I = Var(\underline{x}) = \frac{\int (x - CG)^2 z(x) \mathrm{d}x}{\int z(x) \mathrm{d}x}$$

and is estimated as:

$$I = \frac{\sum_{i=1}^{N}(x_i - CG)^2 s_i z_i}{\sum_{i=1}^{N} s_i z_i}$$

When the dispersion of the population around its centre of gravity is the same along every direction, the spatial distribution is said to be isotropic. In general, if the dispersion of a population around its centre of gravity is not identical in every spatial direction, there is an anisotropy.

In two dimensions, the total inertia of a population can be decomposed into its two principal axes, orthogonal to each other, explaining the maximum and the minimum of the inertia, respectively. These two principal axes and their inertia can be obtained as the eigen vectors and values of a principal component analysis of the coordinates of individuals in the population (i.e. the coordinates of the samples weighted by fish densities). The square root of the inertia along a given axis (or root mean square distance to *CG*) gives the standard deviation of the projection of the location of the population along that axis. These can be represented conveniently on a map with a cross depicting the two principal directions, or with an ellipse (with area proportional to the total inertia). The anisotropy index (≥1) is the square root ratio between the maximum and the minimum of the inertia. Similarly, an index of isotropy can be defined as the inverse of anisotropy, ranging more conveniently from 0 to 1:

$$\text{Isotropy} = \sqrt{\frac{I_{min}}{I_{max}}} \text{ and Anisotropy} = \sqrt{\frac{I_{max}}{I_{min}}}$$

Variations through years of an index such as the *CG* may be due to variations in the sampling pattern (bad weather…) and may not be significant. In such cases, the variations in the index computed with fish density values can be compared to the variations in the same index computed on the sampling itself (unweighted, i.e. using a variable equal to 1 at each datapoint) for significance.

Application 3.3. Centre of gravity, inertia, and isotropy of hake

Here, the mean location, the dispersion around the mean location, and geometry of a fish spatial distribution is captured using the centre of gravity, the inertia, and the isotropy. This example is taken from trawl samples of the age 0 hake population in the Bay of Biscay for a given year (Woillez *et al.*, 2007). The following R code (full script in Annex 3, data details in Annex 2) computes and plots these spatial indices using the function SI.cgi() (Figure 3.4). The function is run after a projection has been defined and areas of influence have been computed to be used as weighting factors.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.hake.bob.db.data","db.data")
rg.load("Demo.hake.bob.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

# Compute and plot the inertia, the total abundance, the isotropy,
# the center of gravity, and the coordinates of the axes of inertia.
# Note that intermediate results of the PCA decomposition are provide
d
# (the eigen values and the eigen vectors).
plot(db.data,title="Centre of gravity and inertia of densities and sa
mples",asp=1,
     xlim=c(-300,150),ylim=c(-200,150),inches=5,
     xlab="Nautical mile",ylab="Nautical mile")
```

```
plot(poly.data,col=8,add=T)
SI.cgi(db.data,flag.plot=TRUE,flag.inertia=TRUE,col=2)
```

```
## $inertia
## [1] 2773.656
##
## $weight
## [1] 69915222
##
## $iso
## [1] 0.38866
##
## $center
## [1] -1.897112 40.254660
##
## $mvalue
## [1] 2409.6609  363.9952
##
## $mvector
##            [,1]      [,2]
## [1,] -0.6220663 0.7829646
## [2,]  0.7829646 0.6220663
##
## $axes
##           [,1]      [,2]
## [1,] -32.43329 78.689058
## [2,]  28.63906  1.820262
## [3,]  13.04080 52.122850
## [4,] -16.83502 28.386471
```

The previous computation is performed in the projected space. However, the projected space may not be informative for the centre of gravity. Thus, the following R code converts the projected coordinates of the centre of gravity back to the geographical space (i.e. in degrees).

```
# Get the coordinates of the centre of gravity in degrees
projec.invert(SI.cgi(db.data,flag.plot=F)$center[1],
              SI.cgi(db.data,flag.plot=F)$center[2])
```

```
## $x
## [1] -3.780402
##
## $y
## [1] 47.09953
```

The following R code computes the centre of gravity and the axes of inertia of the samples, not weighted by the fish densities. This shows that these are distinct from the mean and variance location of the age 0 hake population (Figure 3.4). This can be used, for example, to check if changes of the centre of gravity of a population are not due to a change in the distribution of the samples.

```
# Compute and plot the inertia, the total abundance, the isotropy,
# the centre of gravity, and the coordinates of the axes of inertia
# and the isotropy of the samples
plot(db.add(db.data,S=1),add=TRUE,col=1,inches=5,pch="+")
SI.cgi(db.add(db.data,S=A0>=0),flag.plot=T,flag.inertia=T,col=1)
```

```
## $inertia
## [1] 12474.91
##
## $weight
## [1] 33663.73
##
## $iso
## [1] 0.3059678
##
## $center
## [1] -14.74390  18.96255
##
## $mvalue
## [1] 11407.027  1067.883
##
## $mvector
##            [,1]      [,2]
## [1,]  0.7415709 0.6708746
## [2,] -0.6708746 0.7415709
##
## $axes
##            [,1]       [,2]
## [1,]  64.458596 -52.689323
## [2,] -93.946405  90.614427
## [3,]   7.179259  43.195963
## [4,] -36.667068  -5.270859
```

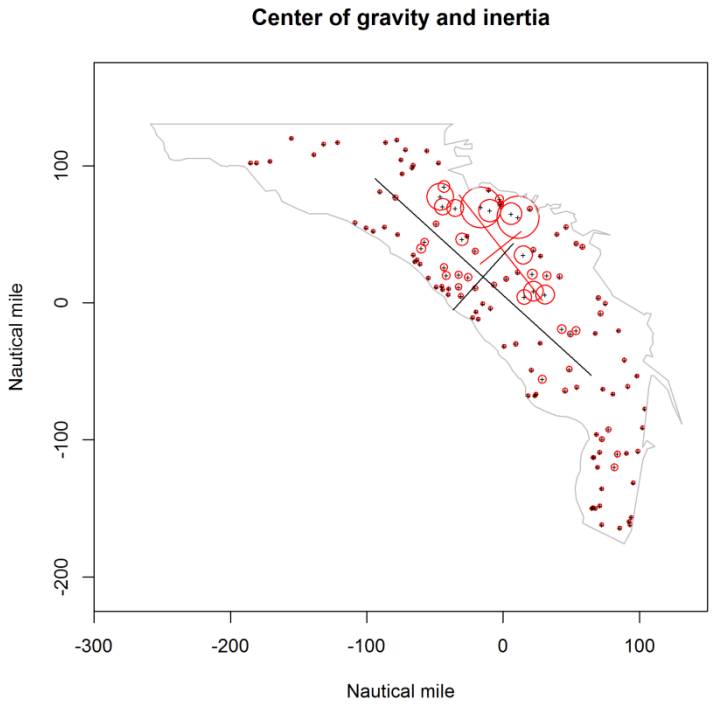


Figure 3.4. Map of the Bay of Biscay with a bubbleplot of hake densities with their centre of gravity and inertia (in red). Note the difference with the centre of gravity and inertia (black axes) of the samples (represented with a black cross).

3.6 Global index of collocation

The global index of collocation looks at the extent to which two populations are geographically distinct by comparing the distance between their *CG*s and the mean distance between individual fish taken at random and independently from each population (Bez and Rivoirard, 2001; Woillez *et al.*, 2009a).

Let us consider two populations with densities $z_1(x)$ and $z_2(x)$ at point x, with ΔCG being the distance between their centres of gravity and I_1 and I_2 their respective inertias. The mean square distance between individuals taken at random and independently from each population is $\Delta CG^2 + I_1 + I_2$, and the global index of collocation (GIC) is:

$$GIC = 1 - \frac{\Delta CG^2}{\Delta CG^2 + I_1 + I_2}$$

or 1 if $\Delta CG^2 = I_1 = I_2 = 0$. The GIC indicator ranges between 0, in the extreme case where each population is concentrated on its own single point at different locations (inertia = 0, $\Delta CG^2 > 0$), and 1, where the two CGs coincide. That the mean locations of the two populations coincide does not mean that their individuals are present at the same locations; populations may occupy the same region while not being observed at the same places within this region (then the local index of collocation to be seen next will be zero).

Application 3.4. Global index of collocation of hake

The following R script lines (see full script in Annex 3) compute the global index of collocation between age 0 (in red) and age 1 (in blue) hake densities and display the spatial distributions of both ages (Figure 3.5).

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.hake.bob.db.data","db.data")
rg.load("Demo.hake.bob.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

# Compute the global index of collocation between age 0 and age 1
SI.gic(db1=db.data,db2=db.data,name1="A0",name2="A1",
       flag.plot=T,flag.inertia=T,asp=1,inches=5,
       xlab="Nautical mile",ylab="Nautical mile",
       col1="red",col2="blue",)

## [1] 0.898

plot(poly.data,col=8,add=T)
```

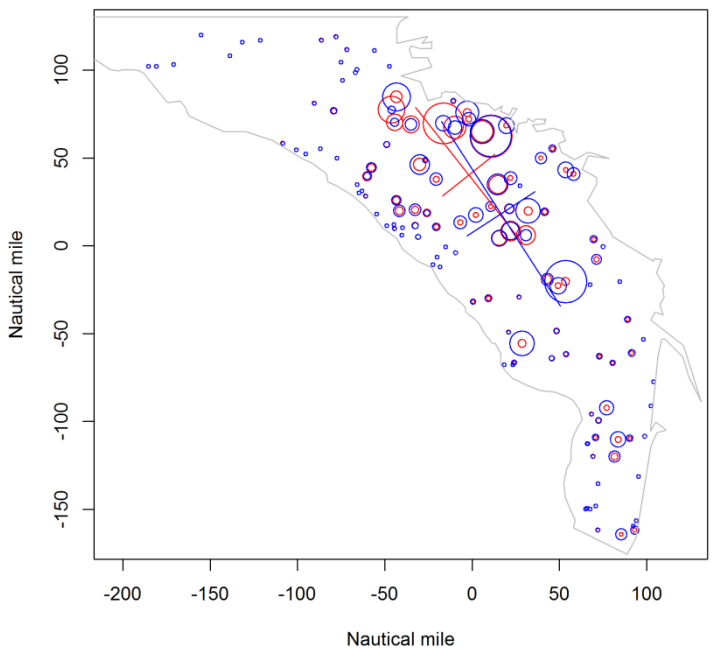


Figure 3.5. Map of the Bay of Biscay with a bubbleplot of age 0 and age 1 hake densities and their centres of gravity and inertia (in red and blue, respectively). Note that the centres of gravity are close with respect to the inertia, as illustrated by a high value of global index of collocation (= 0.898).

3.7 Local index of collocation

The local index of collocation measures the occurrence of two populations at the same locations, e.g. at the same stations (Bez and Rivoirard, 2000). Noting $z_1(x)$ and $z_2(x)$ as the densities of populations 1 and 2 at location x. The local index of collocation is:

$$LIC = \frac{\int z_1(x) z_2(x) dx}{\sqrt{\int z_1(x)^2 dx \times \int z_2(x)^2 dx}}$$

and is estimated as:

$$\frac{\sum_{i=1}^{N} s_i z_{1i} z_{2i}}{\sqrt{\sum_{i=1}^{N} s_i {z_{1i}}^2} \sqrt{\sum_{i=1}^{N} s_i {z_{2i}}^2}}$$

It varies from 0, when the two populations are never observed in the same place, to 1 when the spatial distributions of the two populations coincide [$z_2(x)$ proportional to $z_1(x)$]. The *LIC* is more demanding and so is expected to have a lower value than the *GIC*. Note that the *LIC* can be close to zero even if the two populations are in the same region and have a high *GIC*.

Application 3.5. Local index of collocation of hake ages 0 and 1

The following R code (full script in Annex 3) computes the local index of collocation between the age 0 and age 1 hake densities spatial distributions (Figure 3.6).

```
# Pre-requisite
projec.toggle(0)
```

```
rg.load("Demo.hake.bob.db.data","db.data")
rg.load("Demo.hake.bob.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

# Compute the local index of collocation between two collocated spati
al distributions
SI.lic(db.data,name1="A0",name2="A1")

## [1] 0.6774079

# Display
plot(db.locate(db.data,"A0",loctype="z"),title="",col=2,asp=1,
     xlim=c(-300,150),ylim=c(-200,150),inches=5,
     xlab="Nautical mile",ylab="Nautical mile")
plot(db.locate(db.data,"A1",loctype="z"),title="",col=4,add=TRUE,inch
es=5)
plot(poly.data,col=8,add=TRUE)
```

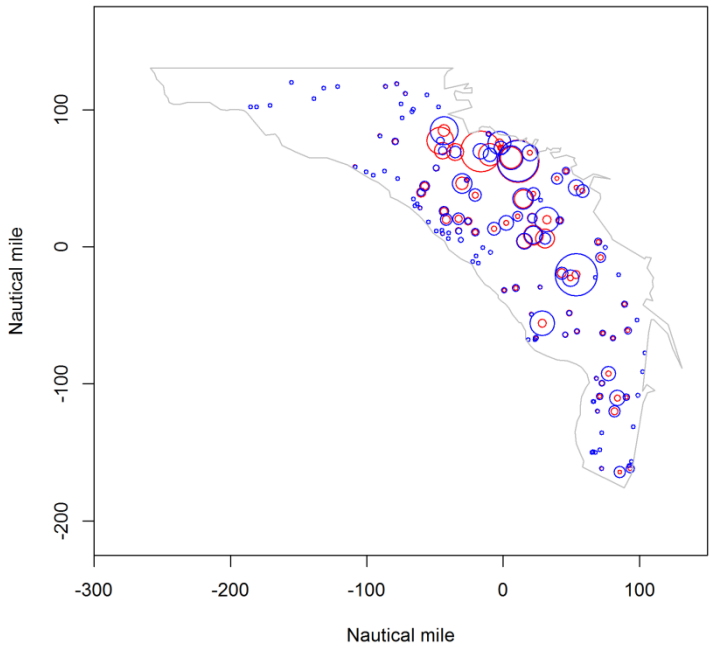


Figure 3.6. Map of the Bay of Biscay showing the proportional representation of the age 0 (in red) and age 1 (in blue) hake densities. Note that the collocated occurrence of both ages spatial distribution is high as quantified by the value of the local index of collocation (= 0.69).

3.8 Microstructure index

The microstructure index of a spatial population (Woillez *et al.*, 2007, 2009a) measures the relative importance of structural components that have a scale smaller than the sample lag (including random noise).

The microstructure index (MI) is taken as the relative decrease in the transitive covariogram (see chapter on variography) between distance zero and a distance h_0 chosen to represent the mean lag between samples (Figure 3.7):

$$MI = \frac{g(0) - g(h_0)}{g(0)}$$

It lies between 0 and 1. Values close to 0 correspond to a very regular, well-structured density surface, and values close to 1 correspond to a highly irregular, poorly structured density surface.

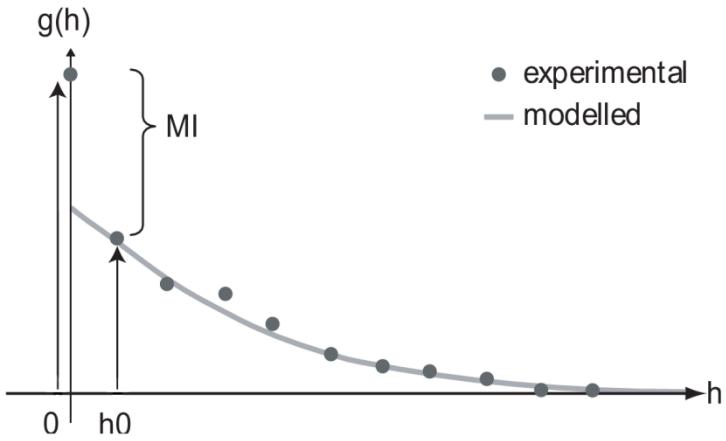


Figure 3.7. Experimental and modelled covariogram with the representation of the microstructure index (*MI*). The covariogram values for distances 0 and h_0 are represented with arrows.

Application 3.6. Microstructure index of hake

The following R code (full script in Annex 3) computes the microstructure index of age 0 hake densities spatial distributions.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.hake.bob.db.data","db.data")
rg.load("Demo.hake.bob.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

# Compute the microstructure index
SI.micro(db.data,h0=10,pol=poly.data,dlim=50,ndisc=400)
## [1] 0.4044576
```

3.9 Number of spatial patches

A population of fish may be distributed into several spatial patches much larger in size than a fish school. An algorithm has been written to identify patches (Woillez *et al.*, 2007, 2009a) by attributing each sample to the nearest patch, with respect to a maximal threshold distance to its CG.

The algorithm starts from the sample value displaying the maximum density $z(x)$ and considers every other sample in decreasing order of density. The maximum value initiates the first patch (Figure 3.8). Then, the current sample value is attributed to the nearest patch if the distance to its CG is smaller than a threshold distance D_{min}. Otherwise, the current sample value defines a new patch. So, the threshold distance D_{min} represents the limit of attraction of a patch and has to be chosen approximately as the maximal expected radius of a patch. Spatial patches whose abundance is $>A_{min}$, say 10% of overall abundance are retained. The summary index is then the number of spatial patches (Figure 3.8).

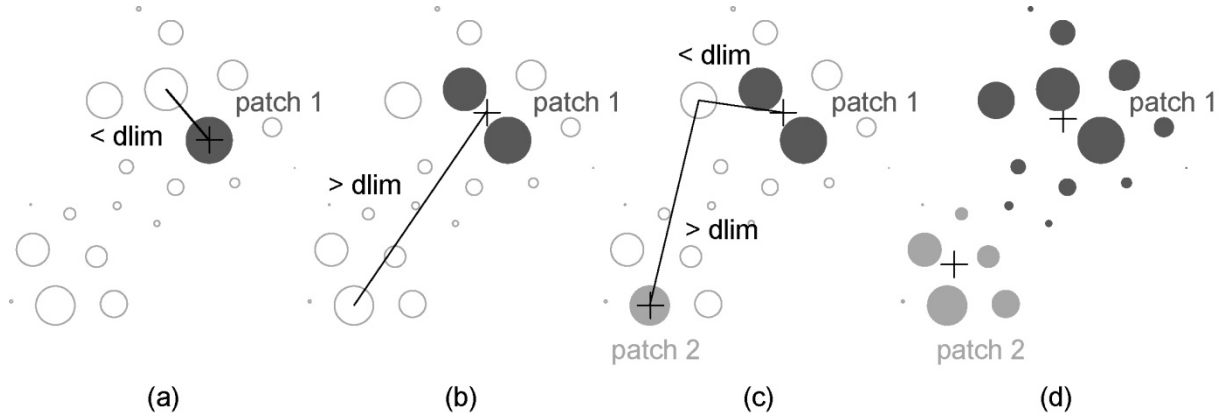


Figure 3.8. Main steps of the algorithm used to determine the number of spatial patches of a population, where the current sample value is attributed to the nearest patch, if the distance to its CG is smaller than the threshold distance D_{lim} (which corresponds to D_{min}).

Application 3.7. Number of spatial patches of hake

The following R code (full script in Annex 3) computes the number of spatial patches index from the age 0 hake densities spatial distributions. The parameters D_{min} and A_{min} need to be defined to run the algorithm.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.hake.bob.db.data","db.data")
rg.load("Demo.hake.bob.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

# The next line calculates the number of spatial patches
SI.patches(db.data,D.min=100,A.min=10)

## Total nb of patches:  4
## Nb of patches with abundance >  10 % :  1
## Percent abundance in these patches:  0.9105
## Percent area in these patches:  0.352

## $n
##   [1] 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1
1 1 1 1 1
##  [36] 1 1 1 1 1 3 1 1 1 1 1 3 1 3 1 1 1 1 1 1 1 4 4 4 4 4 4 4 4
4 4 4 4 4
##  [71] 2 2 4 4 4 2 2 4 4 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1
1 2 2 2 1
## [106] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1
##
## $mat
##   n        xg       yg pabun      parea
## 1 1 -3.887606 47.23635 91.05 0.35199389
## 2 2 -2.388963 45.54781  8.09 0.25069519
## 3 3 -6.259419 47.71692  0.72 0.29848739
## 4 4 -1.640410 44.54609  0.14 0.09882353
##
## $nsp
## [1] 1
```

```
projec.toggle(0)
plot(poly.data,col=8,add=T)
title("Spatial patches")
```

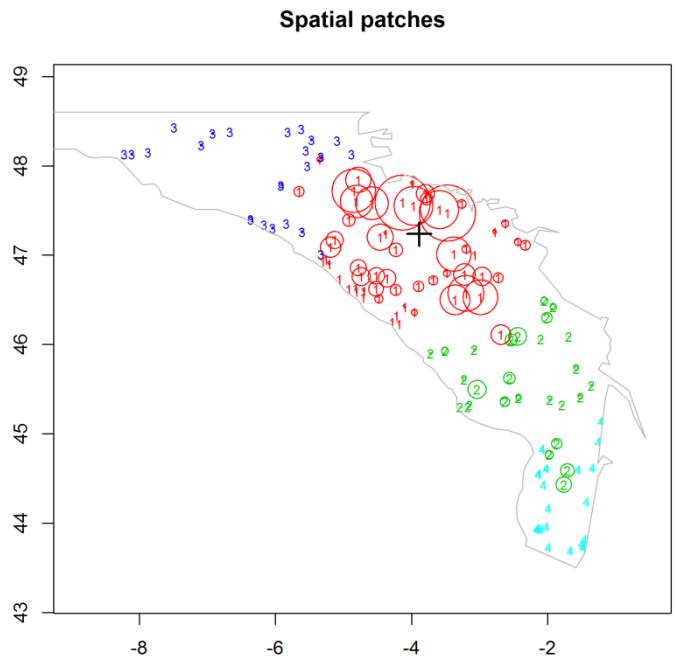


Figure 3.9. Map of the Bay of Biscay showing spatial patches of the age 0 hake population. Four patches were identified using a threshold distance D_{min} = 100 nautical miles. Only the patch in red represents more than A_{min} = 10% of the total abundance. It contains 91% of the abundance and occupies 35% of the area.

4 Structural analysis and variography

The goal of the structural analysis in the geostatistical sense is to capture, describe, and model the way a regionalized variable is spatially structured. For this, geostatistics proposes "structural tools". The best known is the variogram (Figure 4.1), which measures the mean variability between any two points as a function of the distance vector between these points (Matheron, 1971; Chilès and Delfiner, 2012). Indeed, this variability may be expected to vary with this distance between points. First, we will see how to compute the variogram experimentally; then, we will see how to fit the variogram and model the regionalized variable. At the end of this chapter, we will see a variant structural tool, the transitive covariogram.

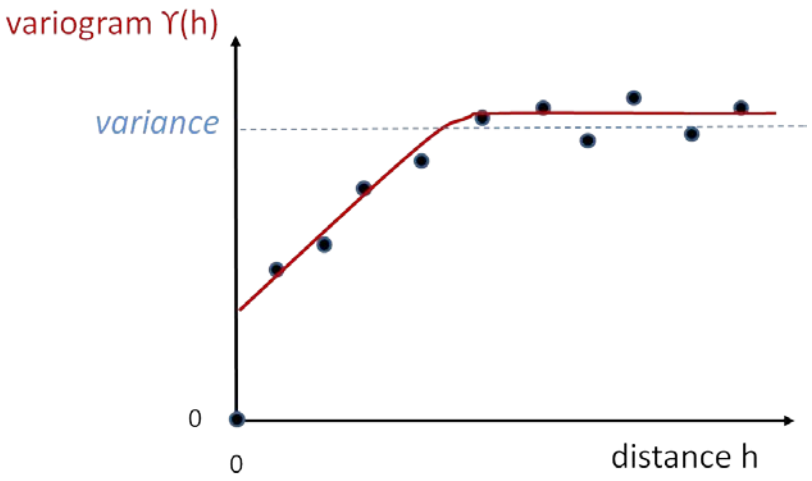


Figure 4.1. Typical variogram representing the mean variability between two points as a function of the distance between these. The bullets represent the experimental variogram computed from data. In this example, they show first a jump from the origin, then an increase in the variability when the separating distance increases, and finally a stabilization at larger distances. This is fitted by a variogram model (continuous line). The dashed line represents the variance of data values.

4.1 Computing the variogram

4.1.1 Regular sampling on a line

Consider N datapoints regularly spaced on a line with mesh size a, starting from an origin x_0, with values $z(x_0), z(x_0+a), \ldots, z(x_0+(N-1)a)$ (Figure 4.2). The variogram $\gamma(h)$ is a function of the distance h (often termed the "lag") between any two points.

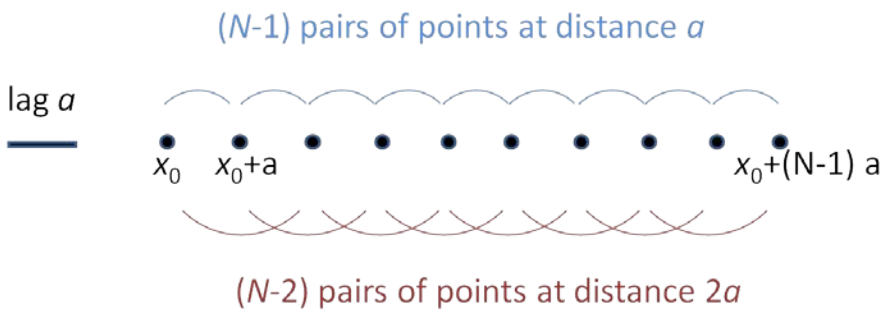


Figure 4.2. Datapoints regularly spaced on a line with mesh size *a*. Pairs of points separated by *a*, and by 2*a* are highlighted.

For the distance $h = a$, there are (N–1) pairs of datapoints with values:

$$\left[z(x_0), z(x_0 + a)\right], \left[z(x_0 + a), z(x_0 + 2a)\right], ..., \left[z(x_0 + (N-2)a), z(x_0 + (N-1)a)\right]$$

This corresponds to (N–1) increments:

$$\left[z(x_0 + a) - z(x_0)\right], \left[z(x_0 + 2a) - z(x_0 + a)\right], ..., \left[z(x_0 + (N-1)a) - z(x_0 + (N-2)a)\right]$$

The variogram at distance a is defined by half the mean of the squared increments and measures the mean variability between two points distant by a:

$$\gamma(a) = \frac{1}{2} \frac{\left[z(x_0 + a)\right) - \left(z(x_0)\right]^2 + \left[z(x_0 + 2a)\right) - z(x_0 + a)\right]^2 + ... + \left[z(x_0 + (N-1)a) - z(x_0 + (N-2)a)\right]^2}{(N-1)}$$

Similarly, there are (N–2) pairs of datapoints at distance $h = 2a$, with values:

$$\left[z(x_0), z(x_0 + 2a)\right], \left[z(x_0 + a), z(x_0 + 3a)\right], ...$$

and (N–k) pairs at distance $h = ka$ (k integer). The variogram at distance $h = ka$ is defined by considering the $N(h) = (N–k)$ pairs of points (x_i, x_j) distant by h through the general formula:

$$\gamma(h) = \frac{1}{2} \frac{\sum_{x_i - x_j = h} (z(x_i) - z(x_j))^2}{N(h)}$$

The larger the distance, the smaller the number of pairs on the line, and the less such an experimental variogram is representative of the variability of the sampled regionalized variable. Generally, an experimental variogram is computed for distances that do not exceed half of the largest dimension in the domain.

4.1.2 Regular sampling on parallel lines

Suppose we have several parallel lines sampled regularly every a along lines, but with a number of samples that can vary with lines. This corresponds typically to an acoustic survey where the EDSU (elementary distance sampling unit) = a. The variogram at distance $h = ka$ along lines, i.e. in the direction of the transects, will be computed from all pairs of points separated by h on any line. If all lines have the same number of samples, the variogram at distance $h = ka$ along lines will coincide with the average of the variograms at distance h along each line. However, in the case of lines having a different number of samples, a line with fewer samples will give fewer pairs. The variogram at distance $h = ka$ along lines will then coincide with the average of the variogram at distance h along each line, weighted by the number of corresponding pairs on this line.

4.1.3 Regular sampling in 2D (square or rectangular grid cell)

As the spatial variability may depend on the direction, the experimental variogram can be computed in different directions, e.g. the principal and the diagonal directions of the sampling grid. If the cell of the sampling grid is rectangular, say (a_1, a_2), the variogram will be computed at distances multiple of a_1 along the first axis, and a_2 along the second one. When the spatial variability revealed by the variogram does not depend

on the direction, an omnidirectional variogram can be computed from pairs in all directions (see further).

4.1.4 Irregular sampling

Variograms can be directional or omnidirectional. Note that an omnidirectional variogram may be more robust as it is computed from more pairs, but it cannot bring out a dependance of the spatial variability with the direction. The omnidirectional variogram will be computed from pairs in all directions. Then, a tolerance will be applied on the distance. Typically the variogram with "lag" a with $\pm a/2$ tolerance, so that each pair of points is allocated to a "bin" $\left(ka-a/2, ka+a/2\right)$. The first bin will be made from pairs at distance between 0 and $a/2$, if any (very often this first) point of the variogram is computed from few pairs and is not significant.

Each directional variogram will be computed similarly, but using only pairs along the direction considered, with a tolerance on the direction; for example, 2D directions at (if starting from 0° corresponding to the east by mathematical conventions) 0°, 45°, 90°, 135°, with tolerance ±22.5° so that any pair of points corresponds to one of these directions (Figure 4.3). In 2D, it is advised to compute the variogram in four directions (or more) rather than two, and not necessarily from 0°, in order to detect the directions presenting the highest spatial continuity. Weighted variograms may be used to take into account the area of influence (s_i) of each datapoint in irregular sampling:

$$\gamma(h) = \frac{1}{2} \frac{\sum_{x_i - x_j \sim h} s_i s_j [z(x_i) - z(x_j)]^2}{\sum_{x_i - x_j \sim h} s_i \, s_j}$$

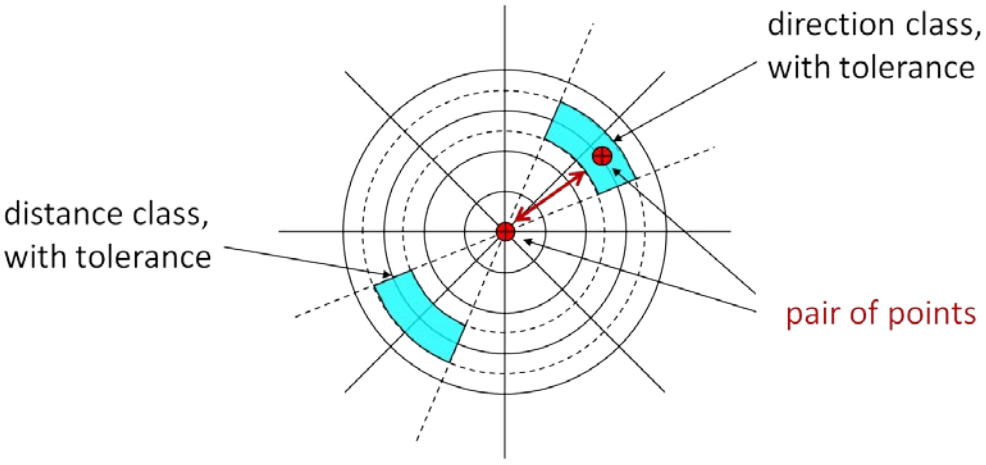


Figure 4.3. Computing variogram with tolerances on distance and direction: each pair of datapoints is allocated to a distance and direction "bin" taking into account the tolerances.

4.1.5 Representation of variograms

Variograms, whether directional or omnidirectional, are usually depicted as a function of distance. Outputs are expected to be robust to the choice of the computation parameters (distance lag, direction). Other representations, less important, can be mentioned. The variogram map is a radial and colored representation of a variogram in all directions. Finally, the variogram cloud represents all individual half-squared increments between datapoints before they are binned and averaged by distance. This sometimes allows identifying high variability pairs and possibly identifying data outliers.

When plotting an experimental variogram, it is helpful to add a line that gives the statistical variance of data values. This is a reference level for the variogram. Indeed, it will be seen in Chapter 5 that this variance coincides with the average of the experimental variogram values for all possible classes of distance, weighted by the number of pairs. This makes the link between statistical variability (variance) and spatial variability (variogram). If, for example, the variogram computed for small or medium distances is lower than the variance, this means that the variogram is necessarily higher than the variance for some larger distances.

Application 4.1. Omnidirectional variogram on demersal survey trawl data

The following R code estimates a distance lag and computes omnidirectional variograms. This application uses data of the MEDITS demersal survey series (data details in Annex 2). Demersal surveys are the typical case of sampling schemes without preferential directions. They usually follow a stratified random sampling protocol so that samples are uniformally distributed in each large stratum. We use here the hake densities expressed in number of individuals per km².

```
# pre-requisite
projec.toggle(0)
rg.load(filename="Demo.hake.med.db.data", objname="db.data")

# Data presentation
plot(db.sel(db.data,YEAR==1996),zmin=0.001,pch.low=3,cex.low=0.25,las
=1,pch=21,col=1,inches=5,title="Hake - 1996",asp=1)
map("worldHires",add=T)
```

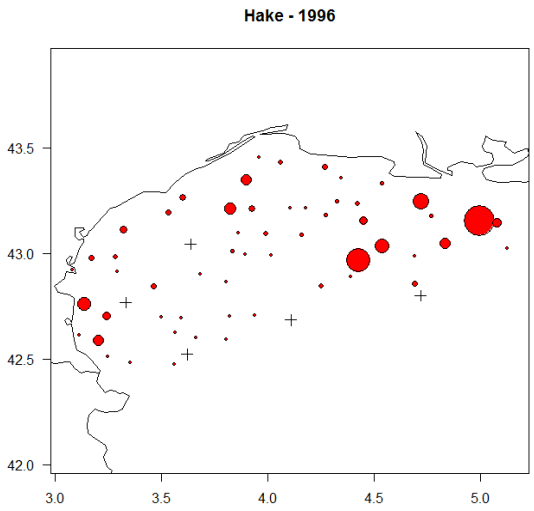


Figure 4.4. Map of the Mediterranean Sea showing bubbleplot of hake densities. Zero densities, represented by crosses, can be found on the edge as well as within the sampled area.

The geographical distribution of the sample data for the 1996 survey indicates that zero observations are rare and not homogeneously distributed at the edge of the sampling area. So, the whole sampled area will be considered, that is, all samples will be used.

The two main parameters for variogram computation are the value of distance lag and the maximum lag to compute to, which, in this case, is designated by the number of lags. The distance lag corresponds to the average distance between a sample and its nearest neighbour. This can be fully computed from sample location by looking, for

example, at the mean distance to the nearest neighbor or estimated from the map using the function dist.digit(). Beware that the map must be plotted in the projected space to get a distance measurement.

```
# Evaluate the distance lag (in projected units i.e. nautical miles)
# and the number of lags by clicking on points
projec.define("mean",db=db.data)
plot(db.sel(db.data,YEAR==1996),zmin=0.001,pch.low=3,cex.low=0.25,las
=1,pch=21,col=1,
     inches=5,title="Hake - 1996",asp=1)
     worldHires <- map("worldHires",plot=F,xlim=c(3,5),ylim=c(42,44))
lines(projec.operate(worldHires $x,worldHires $y))

lag <- dist.digit()

# click on the graph at two adjacent sample locations
Designate the two points
Distance =  5.1
```

It is recommended to compute variograms not exceeding half the diameter of the field. We can also estimate the largest dimension of the sampled area using the function dist.digit().

```
diagonal <- dist.digit()


# click on the graph at two sample locations separated by the greates
t distance
Designate the two points
Distance =  92.19
```

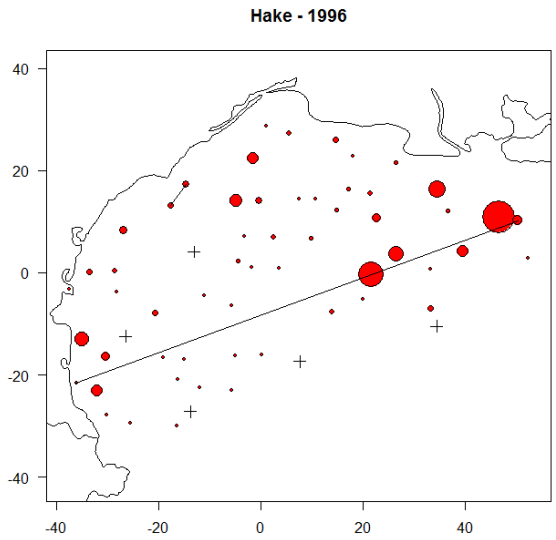


Figure 4.5. The distance between neighbouring samples is about 5 nautical miles (short black line), while the largest dimension of the field is about 90 nautical miles (long black line).

Half the largest dimension equals 45 nautical miles. It is thus covered by nine legs of 5 nautical miles.

```
# Compute and represent omnidirectional variogram
vg.data <- vario.calc(db.sel(db.data,YEAR==1996), lag=5,nlag=9)

# Edit the results
vg.data

Variogram characteristics
=========================
Number of variable(s)  = 1
Number of direction(s) = 1
Space dimension        = 2

Direction 1
-----------
Number of lags              = 10
Direction coefficients      = (     1.000     0.000)
Direction angles (degrees)  = (     0.000)
Tolerance on direction      = 90.000000 (deg)
Calculation lag             = 5.1
Tolerance on distance       = 50.000000% (of the Lag value)

For variable 1
Referenced value (variance,...) = 12378070.3
      Rank              Npairs            Distance                Valu
e
         1              54.000               5.505          7307421.2
1
         2             105.000               9.876          10529137.
4
         3             178.000              14.934          9075330.9
0
         4             180.000              20.247          9419314.9
1
         5             191.000              25.119          10681845.
0
         6             177.000              30.075          10736612.
7
         7             169.000              34.972          10407282.
7
         8             153.000              39.845          9003176.2
7
         9             129.000              44.734          12692429.
5


# Plot the results
plot(vg.data,las=1,xlab="Distance (n.mi.)")
```

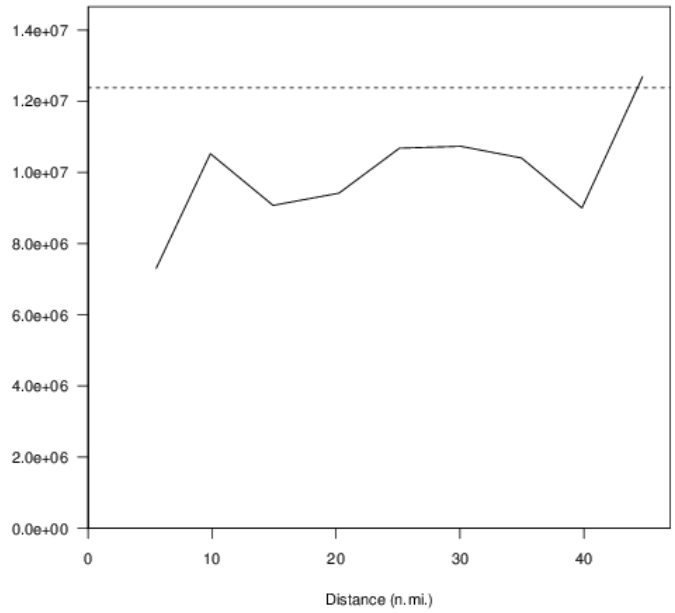


Figure 4.6. Omnidirectional variogram of hake density, up to a distance equal to one-half the diameter of the field.

```
plot(vg.data,npairdw=T,inches=0.1,las=1,xlab="Distance (n.mi.)")
```

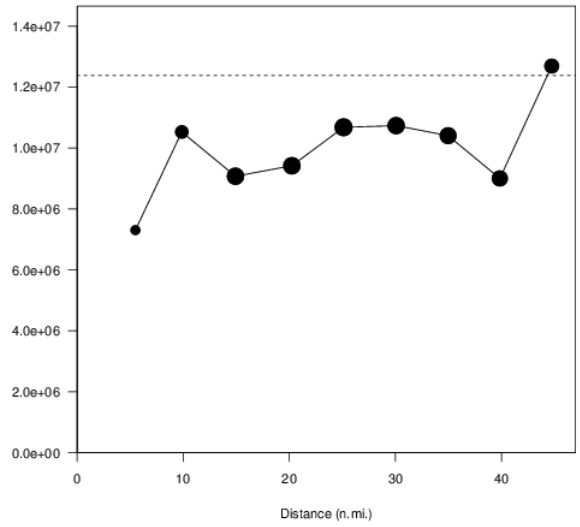


Figure 4.7. Omnidirectional variogram of hake density. The number of pairs is proportional to the size of the black dot (so the first variogram point is less significant). The spatial variability is high and increases only slightly with the distance.

The number of pairs of points can also be represented by a symbol whose size is proportional to this number (Figure 4.7). The variogram value at 0 distance is trivially equal to 0 and is generally not represented. The variogram value at the first distance lag is made of the average over 54 pairs of sample data. The rest of the variogram is supported by at least twice more number of pairs of points and will thus be heavier in the model choice. No graphical connection is done between the origin and the first point to avoid overoptimistic interpretation of the variogram at short distance. The horizontal line represents the variance of the data. As said before, this is the reference level for the variogram values as the mean of the variogram values for all possible distances (weighted by the number of pairs) is equal to the variance. Here, since all variogram values are below the variance up to a distance of 40 nautical miles, some variogram values larger than the variance must occur at larger distances.

4.1.6 Mean variogram

We have seen above that the mean variogram along lines was, for each distance, the average of the individual variograms of each line, weighted by the number of pairs. Similarly computing a "mean variogram" from several variograms can use a weighting by the numbers of pairs. This is typically useful when looking at a time-series of surveys (Fernandes and Rivoirard, 1999). Individual variograms can be made for each survey, but the mean variogram per survey can be a more robust description of the spatial structure. It is also possible to compute variograms from pairs made of two points coming from different surveys (e.g. interyear spatial variability).

As mentioned before, the variance of data gives the average level of the variogram. This level may be very different for the variograms of different datasets, so that variograms may be "normalized" or "standardized" (each variogram being divided by its variance, or equivalently each dataset being standardized by its standard deviation) before computing a mean variogram.

Application 4.2. Comparing and averaging omnidirectional variograms in demersal survey

The following R code (full script in Annex 3) compares omnidirectional (normalized) variograms and computes their average (Morfin *et al.*, 2012). The approach is applied on the MEDITS demersal survey data (data details in Annex 2).

```
# pre-requisite
projec.toggle(0)
rg.load(filename="Demo.hake.med.data", objname="db.data")
projec.define(projection="mean",db=db.data)

# Compute annual variograms and superimpose them
for(i in unique(db.data[,"YEAR"])){
  vg.data <- vario.calc(db.sel(db.data,YEAR==i), lag=5, nlag=9)
  plot(vg.data,npairdw=T,inches=0.05,col=rgb(0,0,0,0.25),add=!(i==199
6),
las=1,xlab="Distance (n.mi.)",ylim=c(0,1e+08))
}
```

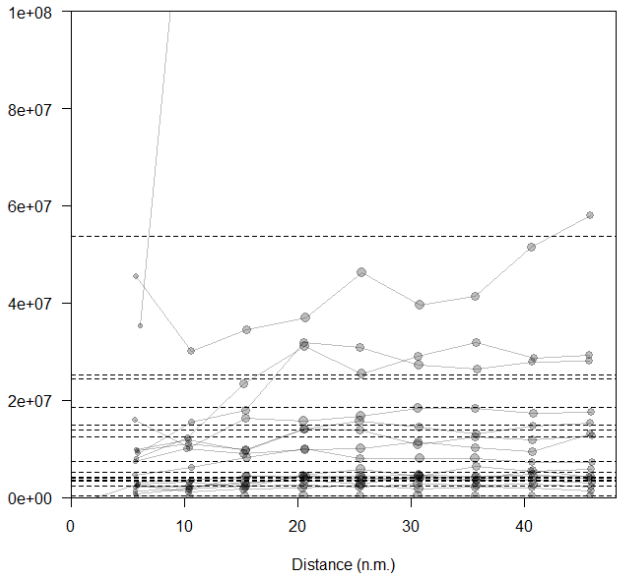


Figure 4.8. Multiple annual omnidirectional variograms of hake density. The variation in the annual variance makes it impossible to evidence a common spatial structure.

Interpretation of the series of annual variograms is made impossible by the differences between the annual variances. Density-dependent mechanisms are known to be present in most ecosystems. They lead to relationships between mean and variance of fish densities, the two increasing together, though not necessarily linearly. It is, however, expected that, within a reasonable range of biomass, aggregative behaviors are the same, leading to the same spatial structures relative to the average biomass and thus to its variance. We will see further that kriging estimates (contrary to the kriging estimation variances) are only dependent on the shape of the variogram and not on the level of the variogram. A standardized mean variogram can help idendify recurrent spatial structure. Standardizing individual variograms can be done by previously standardizing individual samples by their standard deviation or by dividing the variogram values by the variance using the argument flag.norm=T.

```
# superimpose several omnidirectional standardized variograms
for(i in unique(db.data[,"YEAR"])){
  vg.data <- vario.calc(db.sel(db.data,YEAR==i), lag=5, nlag=9)
  plot(vg.data,npairdw=T,inches=0.1,col=rgb(0,0,0,0.25),add=!(i==1996
),
       flag.norm=T,las=1,xlab="Distance (n.mi.)",ylim=c(0,2))
}
```

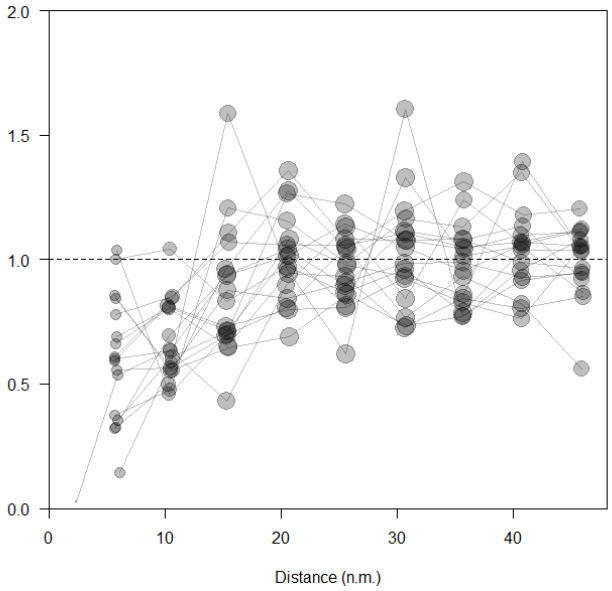


Figure 4.9. Multiple standardized annual omnidirectional variograms of hake density. A common spatial structure emerges, though with statistical fluctuations.

One can see that, for one survey, some sample points were less 2.5 nautical miles apart. In this case, a first point appears in the experimental variogram at short distance corresponding, however, to a very small number of pairs of points with small explanatory power.

Using all data together can be misleading as spatial and temporal variabilities would be mixed up. In particular, the variability depicted at short distance would correspond mainly to temporal variability; the same samples being performed at the same location survey after survey. In RGeostats, computing the mean variogram can be made directly by using a variable coding for the survey or the year (locator "code") and by looking at pairs of points for which the difference in the coding variable is null. Prior to doing this, sample data must be standardized survey by survey by divid ing sample values by the standard deviation of the survey.

```
# Standardize the density by the annual standard deviation and create
a new file.
# Note that 1/n is used for variance calculation and not 1/(n-1)
# YEAR has locator "code" for selecting pairs of points from the same
year
db.data.std <- db.data
for(i in unique(db.data[,"YEAR"])){
  sel <- db.data.std[,"YEAR"]==i
  sd.year <- sqrt(mean(db.data.std[,"MERLMER"][sel]^2) -
                  mean(db.data.std[,"MERLMER"][sel])^2)
  db.data.std[,"MERLMER"][sel] <- db.data.std[,"MERLMER"][sel]/sd.yea
r
```

```
}
db.data.std <- db.locate(db.data.std,"YEAR","code")

# Compute annual variograms (which are normalized because of the stan
dardization of
the density values)
for(i in unique(db.data[,"YEAR"])){
  vg.data <- vario.calc(db.sel(db.data.std,YEAR==i), lag=5, nlag=9)
  plot(vg.data,npairdw=T,inches=0.1,col=rgb(0,0,0,0.25),add=!(i==1996
),
       las=1,xlab="Distance (n.mi.)",ylim=c(0,2))
}


# Compute the mean annual variogram: Pairs are retained if their code
s are the same
# i.e. if their difference is equal to 0
vg.data.std <- vario.calc(db.data.std, lag=5, nlag=9,opt.code=1,tolco
de=0)
plot(vg.data.std,npairdw=T,inches=0.1,las=1,add=T,col=2,lwd=2)
```

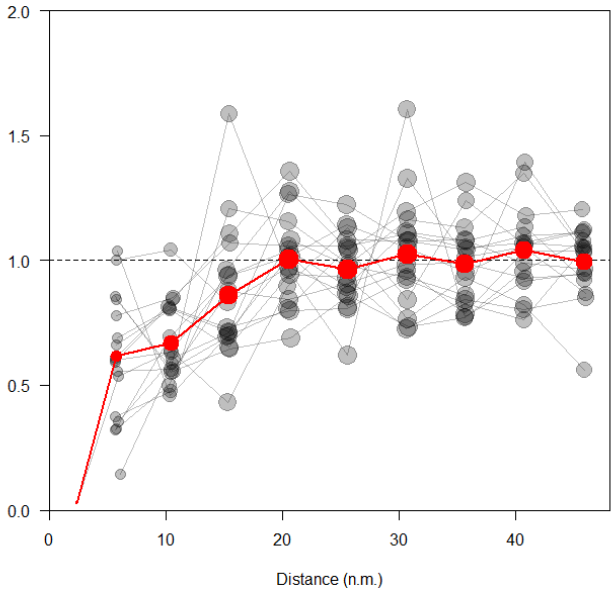


Figure 4.10. The mean annual variogram estimates the spatial structure common to all observations

4.2 Intrinsic model

The variogram is central for what is called the *intrinsic approach* in geostatistics, which aims at describing the behavior of the variable (e.g. fish density) within a given spatial domain. Because of its variability, the regionalized variable $z(x)$ is conveniently represented by a random function (or random process) model $Z(x)$ (note capitalization). The inference of the model is made possible by a hypothesis of stationarity (invariance under translation) on the variable or its increments. In the following, we will first consider the intrinsic approach and then the transitive one.

Note that the expectation or expected value of a variable Z, considered as random, corresponds to its mean, and is denoted as $E[Z]$. The variance is the expectation of the

squared variable centered by its mean: var $Z = E\{(Z - E[Z])^2\}$. When the mean is 0, the variance is the expectation of the squared variable var $Z = E[Z^2]$.

In the intrinsic approach, common models are the intrinsic and the stationary random function models (Matheron, 1971; Chilès and Delfiner, 2012). In the *intrinsic model*, the increments $Z(x + h) - Z(x)$ between two points separated by distance vector h are (order 2) stationary:

- their expectation is 0: $E[Z(x + h) - Z(x)] = 0$ which means that there is no drift (no systematic variation);
- their variance depend on h (not on x). The variogram is the function of h defined as half this variance:

$$\gamma(h) = \frac{1}{2} E\left\{\left[Z(x+h) - Z(x)\right]^2\right\}$$

It satisfies $\gamma(-h) = \gamma(h)$ and is positive, except for $\gamma(0) = 0$.

This structural tool characterizes the intrinsic random function model. The intrinsic model allows computing the variance of any linear combination $\sum \lambda_i Z(x_i)$ with weights summing to 0 (the expectation of such a linear combination is 0):

$$\text{var}\left[\sum_i \lambda_i Z(x_i)\right] = -\sum_{ij} \lambda_i \lambda_j \gamma(x_i - x_j) \text{ with } \sum_i \lambda_i = 0$$

Only such linear combinations are defined and so authorized in this model (the simplest ones being the increments). The variogram cannot be any mathematical function as it must ensure the positivity of the variance of such linear combinations. In the next chapters, the intrinsic random function model with its variogram model will be used to computed variances and perform kriging.

In the *stationary model*, $Z(x)$ is (order 2) stationary, with a constant mean $m = E[Z(x)]$ and a covariance depending on h:

$$C(h) = \text{Cov}\left[Z(x), Z(x+h)\right] = E\left[\left(Z(x) - m\right)\left(Z(x+h) - m\right)\right] = E\left[Z(x)Z(x+h)\right] - m^2$$

which satisfies $C(-h) = C(h)$.

This model allows computing the variance of any linear combination $\sum \lambda_i Z(x_i)$:

$$\text{var}\left[\sum_i \lambda_i Z(x_i)\right] = \sum_{ij} \lambda_i \lambda_j C(x_i - x_j)$$

$C(0)$ represents the variance of $Z(x)$ in the model, and we have $|C(h)| \le C(0)$. The ratio $C(h) / C(0)$ gives the autocorrelation function or correlogram $\rho(h)$.

If $Z(x)$ is stationary, it is also intrinsic (and then the variogram $\gamma(h) = C(0) - C(h)$ is bounded). The reverse is false, and the variogram is a more general structural tool than the covariance. The stationary model is regulated around its constant mean (Figure 4.11). This is not necessarily the case with the intrinsic model, which has more flexibility, although it does not include any drift or trend. The case of a trend will be considered further in the multivariate chapter.

In practice, variograms are computed and fitted for distances that do not exceed half the domain, often less. This means that the stationarity (of the variable or of its increments) may be local only.

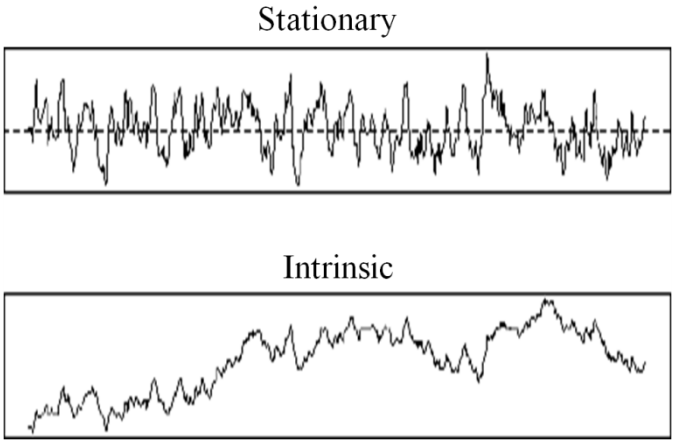


Figure 4.11. A stationary process is regulated around its constant mean. An intrinsic process can be more flexible.

4.3 Variogram properties and variogram fitting

The behavior of the variogram at the origin characterizes the more or less important spatial continuity of the variable. The variogram may stabilize at a "sill", from a distance called the range [in this case, the variogram can be related to a covariance $C(h)$, with its sill equal to $C(0)$]. There is anisotropy if the behavior of the variogram depends on the direction of h, isotropy otherwise. Clearly, directional variograms are required to make evidence of an anisotropy.

In practice, an experimental variogram is computed on data as seen above. Then, it must be fitted by an appropriate model. This is done (manually or automatically) using a mathematically authorized function. This is usually obtained with a simple single structure (e.g. nugget, spherical, exponential, linear) or a sum of such structural components (nested structures) (see, for example, Chilès and Delfiner, 2012 or Rivoirard *et al.*, 2000).

The nugget effect corresponds to a discontinuity at the origin. It may correspond to a small-scale component which is not resolved by the sampling design. Random errors when measuring a variable are also responsible for a nugget component. A "pure" nugget effect, i.e. a nugget effect without any additional structure, corresponds to the variance.

The exponential and spherical models have a linear behavior at the origin, corresponding to more continuity than the nugget effect. They stabilize on a sill at a separation distance (lag) termed the “range”. So, they are bounded and can be associated with a stationary covariance.

The linear variogram is unbounded. It corresponds to an intrinsic random function, but not a stationary one. In 1D, the Brownian motion is an example of a process with linear variogram.

In the case of anisotropy, the different directional variograms must be fitted using a unique variogram model that includes the anisotropy. Anisotropy is often modeled through a "geometrical" anisotropy. This is equivalent to a linear transformation of the coordinates. For example, a spherical model with the largest range along the north–south direction and the shortest range in the east–west direction would correspond to an isotropic structure if appropriately reducing the north–south distances or increasing

the east–west distances in the geographical space. In the case of a "zonal" anisotropy, there is a structural component in the variogram which only acts in some direction(s), e.g. along the vertical in 3D. This can be detected from the variogram by a sill which depends on the direction (e.g. smaller along horizontal planes if the zonal component is vertical).

Application 4.3. Directional variograms on acoustic data

In acoustic surveys, the sampling has high-resolution data along transects and transects separated by tens of nautical miles. The following R code (full script in Annex 3) computes the variogram along and across transects with different lag distances to check for structural anisotropy in the fish distribution. The example dataset is that of anchovy (*Engraulis encrasicolus*) in the Bay of Biscay (Annex 2).

Before computing the variogram, we load the data, define a projection, and check for duplicated locations in the sample points.

```
# pre-requisite
projec.toggle(0)
rg.load(filename="Demo.anchovy.bob.2d.db.data", objname="db.data")
rg.load(filename="Demo.anchovy.bob.2d.poly.data",objname="poly.data")
projec.define(projection="mean")
db.data <- duplicate(db.data)

# Calculate directional variogram
vg2 <- vario.calc(db.data,lag=c(2,15),dirvect=c(35,145), nlag=c(40,7)
)
plot(vg2,npairpt=0,npairdw=TRUE,title="",inches=.05)
```

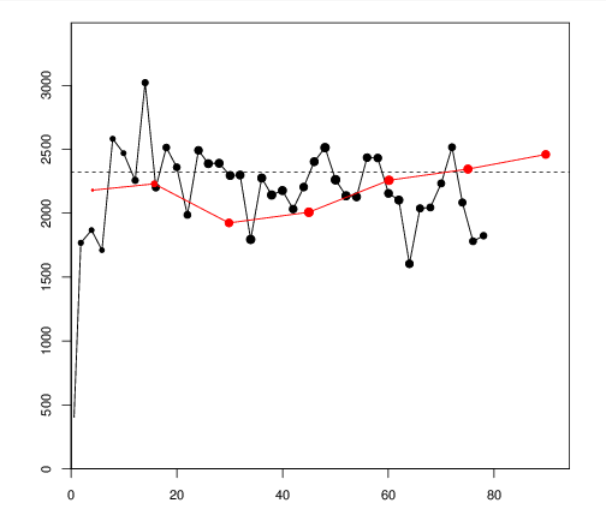


Figure 4.12. Calculation of experimental variograms, with lag depending of the direction (in black along transects, in red accross transects).

The variogram is calculated with function vario.calc() in the direction 35° (along transects) counted with a lag distance of 2 nautical miles (black colour) and in the direction 145° (across transects) with a lag distance of 15 nautical miles (red colour). Directions are counted trigonometrically. The variogram in the across-transect direction attains the sill at its first lag. The variogram structure is assumed isotropic as no structural difference is evident from the directional variograms.

The variogram is then reestimated in omnidirection mode and fitted automatically with a nugget effect and a spherical model using the function model.auto(). The variogram range is close to the intertransect distance.

```
# omnidirectional variogram
vg <- vario.calc(db=db.data,lag=2, nlag=40)

# fit isotropic variogram
vg.mod <- model.auto(vario=vg,struct=melem.name(c(1,3,3)),npairpt=0,npairdw=TRUE,
                     title="",inches=.05)
```

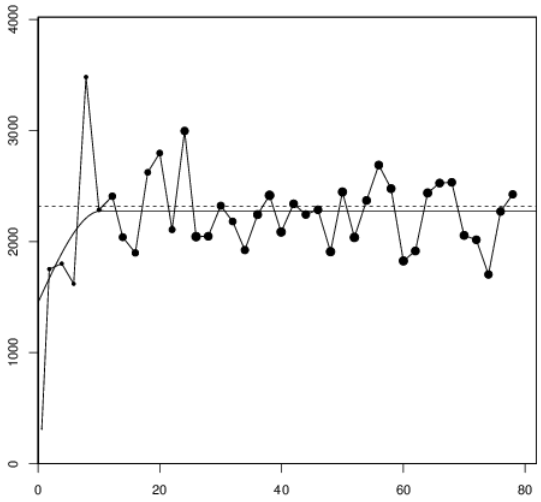


Figure 4.13. Automatic fit of an isotropic variogram model with function model.auto().

4.4 Transitive covariogram

An alternative approach to the intrinsic one exists in geostatistics, which is the *transitive approach* (Matheron, 1971; Bez and Rivoirard, 2001; Chilès and Delfiner, 2012). In this approach, the spatial distribution of a population is studied as a whole, not distinguishing the frontiers from its core. In particular, the delimitation of a domain is not required, and one does not have the problem of whether to include the 0 fish density values. It is based on another structural tool (the transitive covariogram, Figure 4.17). The transitive approach requires fewer hypotheses than the often used intrinsic approach (in particular, there is no reference to a random function model). It is more robust, but less powerful. In particular, it is unable to capture what could be the inner behavior of a regionalized variable within its domain (intrinsic behavior).

In the *transitive approach*, the structure of the regionalized variable is described by the transitive covariogram, function of the distance vector *h*:

$$g(h) = \int z(x)z(x+h)dx$$

This satisfies $g(-h) = g(h)$ and $|g(h)| \le g(0)$. It is positive or null if $z(x) \ge 0$. It satisfies $\int g(h)\,dh = Q^2$ where $Q = \int z(x)\,dx$ [Q is the abundance if $z(x)$ is a fish density]. At large distances, the transitive covariogram stabilizes at 0 from a distance which is the range,

usually depending on the direction. This corresponds to the size (diameter) of the area of presence of the population along this direction. The transitive covariogram has some similarities with a covariance or a variogram. However, it is not an expectation or a mean, but a sum. This makes the transitive covariogram more robust, particularly to outliers. In the case of a regular grid with cell a, it can be estimated for distances $h = k\,a$ multiple of the cell by:

$$g(k\,a) = \mid a \mid \sum_{x_i} z(x_i)\, z(x_i + k\,a)$$

In 2D, with 2D notations $a = (a_1, a_2)$, the covariogram at distances $h = k_1\, a_1$ (k_1 integer) along a_1 direction is, for example:

$$g(k_1\, a_1) = \mid a_1 \mid\mid a_2 \mid \sum_{x_i} z(x_i)\, z(x_i + k_1\, a_1)$$

When the sample grid is not regular, the experimental computation of the covariogram must be weighted by the area of influence of each datapoint (Bez, 1997):

$$g(h) = \frac{1}{2}\left(\sum_{x_i} s_i\, z(x_i) \frac{\sum_{x_i - x_j \sim h} s_j z(x_j)}{\sum_{x_i - x_j \sim h} s_j} + \sum_{x_i} s_i\, z(x_i) \frac{\sum_{x_i - x_j \sim -h} s_j z(x_j)}{\sum_{x_i - x_j \sim -h} s_j} \right)$$

Otherwise, the computation is similar to that of the variogram. The mathematical models to be fitted are similar to the stationary covariance models.

Application 4.4. Transitive covariogram of cephalopod concentrations

The following R code (full script in Annex 3) is an example of transitive covariogram computation in 2D. The data used here correspond to a regular stratified sampling where one sample is taken at random in each square of a 11 x 11 nautical mile regular grid. They correspond to the cephalopod survey carried by INRH (Institut National de Recherche Halieutique) – Casablanca – Morocco (Faraj and Bez, 2007). See Annex 2 for data details.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.octopus.morocco.db.data","db.data")
rg.load("Demo.octopus.morocco.poly.data","poly.data")
worldHires <- map("worldHires",plot=F,add=T)
projec.define(projection="mean",db=db.data)

# Data presentation
plot(db.data, zmin=0.01, pch.low=3, cex.low=0.25, las=1, pch=21, col=
1, inches=5, title="Octopus", xlab="Longitude (°)", ylab="Latitude (°
)", asp=1, flag.proj=FALSE)
plot(poly.data,add=T, flag.proj=FALSE)
map("worldHires",add=T)
legend.proportion(db.data[,7], position="bottomright", zmin=0.01, zrm
in=0.01, zamin=0.01, zamax=max(db.data[,7]), zrmax=max(db.data[,7]),
pch.low=3, cex.low=0.25, pch=21, col=1, bg=2, cex0=0.1, cex1=1, inche
s=5)
```

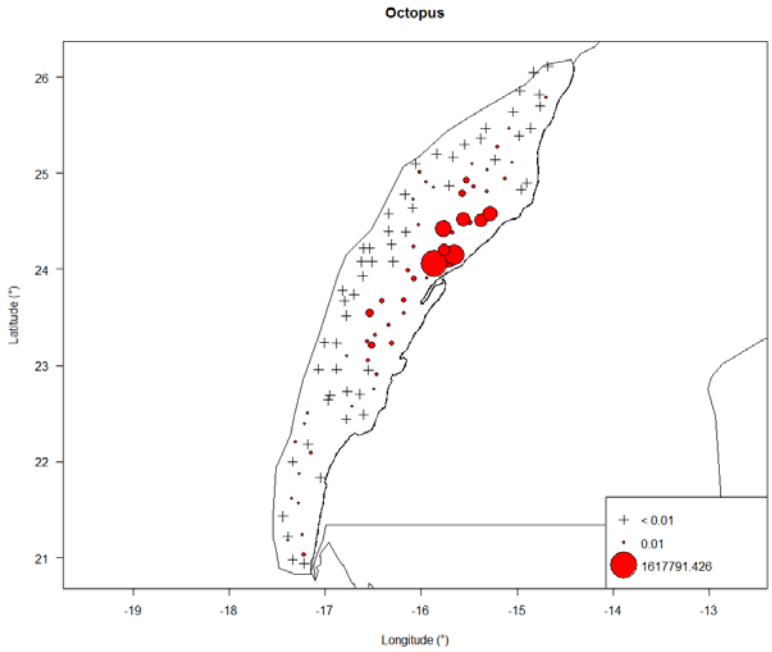


Figure 4.14. Map showing a bubbleplot of the octopus (*Octopus vulgaris*) densities. Many null densities are observed around the heart of the distribution showing that the entire habitat has been surveyed. Large densities are clumped near the coast.

This is a case where there is a relationship between the geometry of the domain and the regionalized variable and where it is not possible to assume that the sampled area is a window of observation which allows having access to statistics that are intrinsic to the phenomenon and relevant anywhere in the field. Stationarity, either of the variable or of its increments, is clearly questionable and transitive geostatistics a relevant approach.

To compute the transitive covariogram, one must first estimate the surfaces of influence of each sample using the function infl(). To be consistent, this must be done in the projected geographical system. The polygon allows restricting the surface of influence of samples that are at the edge of the sampling area.

```
db.data <- infl(db.data, nodes=400, extend=c(6,6), origin=c(-18,20.5)
,
polygon=poly.data, plot=T, asp=1, xlab="Longitude (n.mi.)", ylab="Lat
itude (n.mi.)")
lines(projec.operate(worldHires $x,worldHires $y))
```

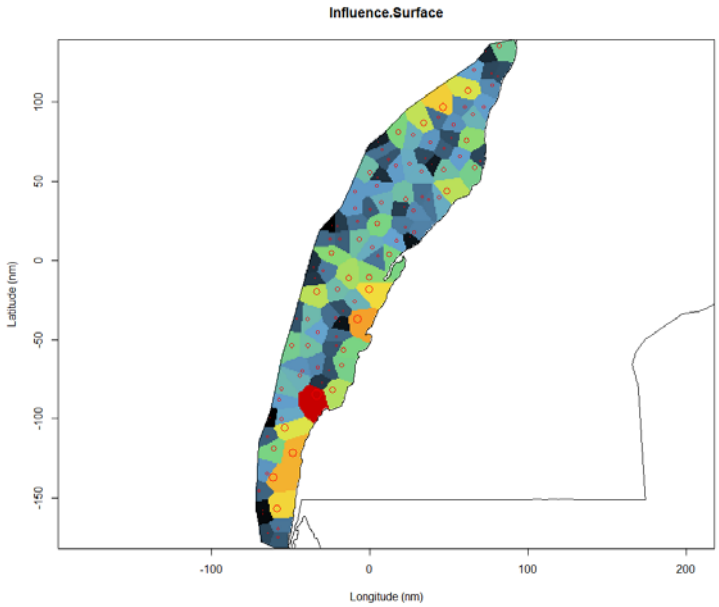


Figure 4.15. Tesselation of the sampling area to estimate the surface of influence of each sample restricted to the study polygon. Bubbles are proportional to the surface area. Colours range from red and yellow for the largest surface areas to blue and dark for the smallest ones.

Lag parameter is equal to the size of the regular strata, i.e. lag = 11 nautical miles. The number of lags must insure that the entire area of presence is covered. The two directi ons of computation follow the main axes of the polygon, i.e. 50° and 140°. The same function is used to compute a variogram or a covariogram. The argument "calcul" sel ects the type of computation that is performed.

```
covario.data <- vario.calc(db.data, breaks=seq(5.5,250,by=11), calcul
="covg", tolang=45, dirvect=c(50,140))


plot(covario.data,npairdw=1,xlim=c(0,250),inches=0.075,pos.legend=7,
xlab="Distance (n.mi.)")
```

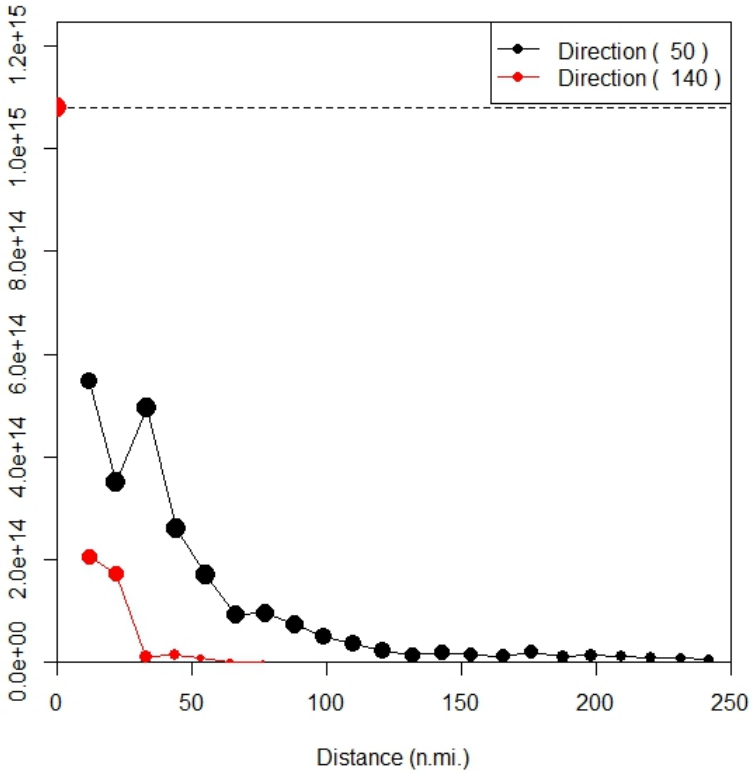


Figure 4.16. Transitive covariogram in directions 50° (black) and 140° (red). The dashed line corresponds to the value of the covariogram at distance zero. As is often the case, the empirical covariogram is strongly anisotropic. This comes from the geometrical anisotropy of the area of presence which impacts the covariogram. Ranges, i.e. diameters of the area of presence in the direction of computation, are approximately equal to 30 and 150 nautical miles, respectively, in the directions 140° and 50°.

We considered adjusting the empirical covariogram with a combination of nugget effect and exponential functions. For the model to remain positive, one must use the argument constraints = covario.data@vars in the function model.auto().

```
# Superimpose the model to the empirical covariogram.
model.covario <- model.auto(covario.data,struct=c(1,3,3),constraints=
covario.data@vars,draw=F)
plot(covario.data,npairdw=1,xlim=c(0,250),inches=0.075,pos.legend=7,
xlab="Distance (n.mi.)")
plot(model.covario, vario=covario.data,lwd=2,add=T)
```

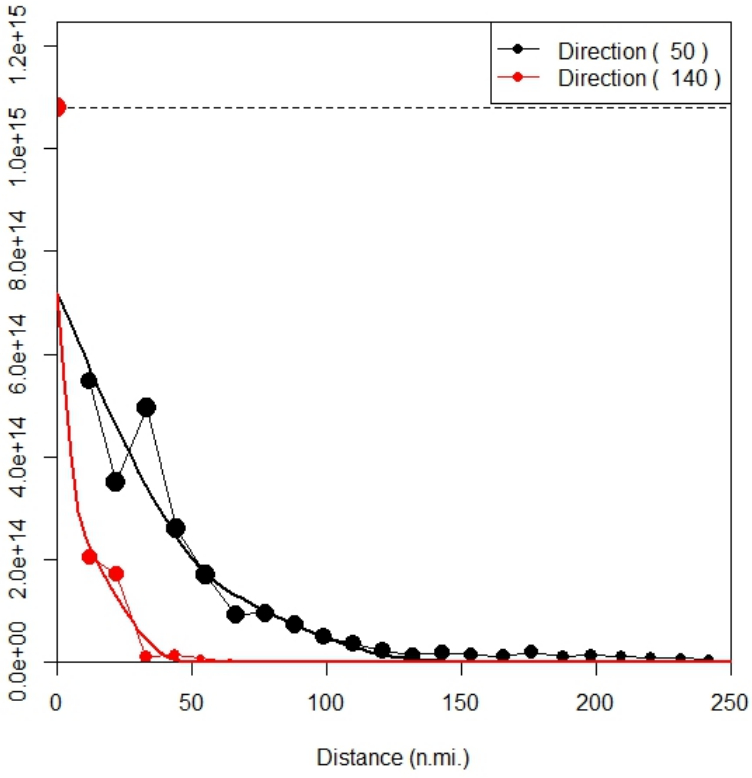


Figure 4.17. Transitive covariogram model superimposed to the empirical one.

5 Dispersion and estimation variances

5.1 Dispersion variance

Geostatistics makes the distinction between two types of variances: dispersion variances and estimation variances (Matheron, 1971; Chilès and Delfiner, 2012). The dispersion variances focus on the dispersion of values taken by the regionalized variable. The estimation variances focus on the errors which are made when estimating the variable at target locations.

Outside of any model, the *dispersion variance* corresponds to the classical statistical variance of values (see Chapter 2). Consider a domain V divided into N equal blocks v_i corresponding to the support v. The regionalized variable is supposed additive, so that the value $z(V)$ is the arithmetic mean of the $z(v_i)$:

$$z(V) = \frac{1}{N}\sum_i z(v_i)\,.$$

The "dispersion variance of v within V" is nothing but the variance of the $z(v_i)$ within V:

$$s^2(v \mid V) = \frac{1}{N}\sum_i [z(v_i) - z(V)]^2$$

Similarly $z(V)$ is the average of point values $z(x)$ within V, and the "dispersion variance of a point within V" is the variance of the $z(x)$ within V denoted as:

$$s^2(o \mid V) = \frac{1}{V}\int_V \left[z(x) - z(V)\right]^2 dx$$

The dispersion variance of a point within each v_i can be defined similarly, and their average represents the "dispersion variance of a point within v" $s^2(o \mid v)$. Interestingly, all such variances are related by an additivity relationship:

$$s^2(o \mid V) = s^2(o \mid v) + s^2(v \mid V)$$

More generally, whatever the domain V partitioned into equal vs, the dispersion variance of v in V depends on the support v (in general, it decreases when v increases), and on the domain V. Finally, it can be shown that the variance of data values (that is, the dispersion variance of points within the set of datapoints) is the average of the experimental variogram values, weighted by the number of pairs. This makes the link between the statistical variability (variance) and the spatial variability (variogram).

Let us now introduce the model. One interest of the intrinsic random function model represented by a variogram model $\gamma(h)$, is its ability to compute and predict the theoretical version (expectation) of dispersion variances. For example, the expected dispersion variance of a point within a set of N points is equal to $\overline{\gamma}(N,N)$, mean value of the variogram $\gamma(y-x)$ when the points x and y describe independently the N points:

$$D^2(o/N) = E\left[S^2(o/N)\right] = \overline{\gamma}(N,N) = \frac{1}{N^2}\sum_{i,j}\gamma(x_i - x_j)\,.$$

If the N points x_i represent the sample points, this corresponds, in the model, to the variance of the values at sample points.

The dispersion variance of a point in V is equal to $\overline{\gamma}(V,V)$, mean value of the variogram $\gamma(y-x)$ when the points x and y independently describe V:

$$D^2(o/V) = E\left[S^2(o/V)\right] = \overline{\gamma}(V,V) = \frac{1}{|V|^2}\int_V \int_V \gamma(x-y)\,dx\,dy\,.$$

The dispersion variance of v in V is equal to $D^2(v/V) = \overline{\gamma}(V,V) - \overline{\gamma}(v,v)$, difference between the dispersion variance of a point within V and the dispersion variance of a point within v.

The dispersion variance of a point within (all datapoints of) the sampled domain V, representing the variance of data, equals $\overline{\gamma}(V,V)$. If there is a sill $C(0)$ and if the domain is large compared to the range, this variance equals $C(0)$. So, it is desirable that the fitted variogram model ensures that $\overline{\gamma}(V,V)$ or $C(0)$ match with the variance of data. However, in practice, it may not be so, because the variogram is not computed on largest distances, and its model does not represent the reality at large distances (stationarity is only local).

When the support changes, the statistical variability changes (dispersion variance) as does the geostatistical variability (see examples in Rivoirard *et al.*, 2000). When its support increases, the variable is generally more regular. If $Z(x)$ has a variogram $\gamma(h)$, its regularized value $Z(v)$ over support v has the regularized variogram:

$$\gamma_v(h) = \overline{\gamma}(v,v_h) - \overline{\gamma}(v,v)$$

and the regularized covariance is

$$C_v(h) = \overline{C}(v,v_h)$$

where v_h is v translated by h.

5.2 Estimation variance from the variogram

The *estimation variance* corresponds to the variance of an estimation error. So, it gives the precision of the estimation and can be predicted by the model. Consider the estimation of $Z(V)$ by $Z(v)$, whatever V or v. The error is $Z(V) - Z(v)$. Within an intrinsic random function model, the expectation of this error is zero (no bias), and the variance of this error (the "estimation variance") is equal to:

$$\sigma_E^2 = \mathrm{Var}\left(Z(V) - Z(v)\right) = 2\overline{\gamma}(V,v) - \overline{\gamma}(V,V) - \overline{\gamma}(v,v)$$

or in the stationary case:

$$\overline{C}(V,V) + \overline{C}(v,v) - 2\overline{C}(V,v)$$

In such formula, $\overline{\gamma}(V,v)$, for example, represents the mean value of the variogram $\gamma(y-x)$ between a point X describing V and a point y describing V. We can see that the estimation variance depends on the geometry of v and V and on the variogram. Note that v or V can be a set of isolated points.

The estimation variance when estimating $Z(v)$ by a punctual value $Z(x)$ equals:

$$2\overline{\gamma}(v,x) - \overline{\gamma}(v,v) - 0$$

Additionally, if the point x is randomly uniform within v (i.e. it can be anywhere with no preference), it reduces to the dispersion variance of a point in V: $\overline{\gamma}(v,v)$

In the case of N sample points random uniform within V and independent, the estimation variance is $\gamma(v,v)/N$.

In the general formula above, V can be a set of isolated points:

$$\sigma_E^2 = E\left[Z(V) - \frac{1}{N}\sum_i Z(x_i) \right]^2 = 2\frac{1}{N}\sum_i \overline{\gamma}(x_i,V) - \frac{1}{N^2}\sum_{ij}\gamma(x_i,x_j) - \overline{\gamma}(V,V)$$

This formula can be used to compute the estimation variance when estimating Z over a domain by the arithmetic mean of values over a set of points. It makes it possible to predict the estimation variance using a given sampling design (particularly for the global estimation of a domain). A variant of the previous equation allows giving different weights to the points and is used in ordinary kriging (next chapter):

$$\sigma_E^2 = E\left[Z(V) - \sum_i \lambda_i Z(x_i) \right]^2 = 2\sum_i \lambda_i \overline{\gamma}(x_i,V) - \sum_{ij}\lambda_i\lambda_j\gamma(x_i,x_j) - \overline{\gamma}(V,V) \text{ with } \sum_i \lambda_i = 1$$

The global estimation variance of a domain can also often be obtained by combining the estimation variances of subdomains dividing the domain. This is so when each subdomain is estimated by inner samples and when the estimation errors of the subdomains are not correlated or when their correlation can be neglected. Random stratified sampling is a good example of this (e.g. IBTS surveys); the domain is divided into blocks, and sample points are taken independently and random uniform within blocks. Because of this, there is no correlation between the errors. Such a principle of composition of variances is also valid with a regular sampling design and usual variogram models. The domain V is divided into equal blocks or cells v, with one sample at the center of each block. The global estimation variance can then be approximated by $\sigma_E^2(V) = \frac{\sigma_E^2(v)}{N}$. This also applies in 1D with a regular sampling on a segment (e.g. when working in 1D on accumulations). A similar composition can also be applied when dividing a 2D domain into slices centred on sampled transects (here the transects, and so the slices, can have different length).

Finally, note that global estimation variances depend on the variogram. Knowing this allows predicting the variances for different envisaged survey designs and, therefore, allows optimizing the sampling design. As it depends on the variogram model, this geostatistical approach can be qualified as model-based, compared with the classical sampling techniques (Cochran, 1977). The latter do not use such a variogram model, but require the locations of samples to be randomized. Typically, variances can be computed for each stratum (providing it contains a minimum of two points), and variances can be combined over a domain divided into strata (having possibly different geometries). The calculation of variances is straightforward, but requires a simple or stratified random sampling design. A model-based approach such as the geostatistical one is necessary in the case of regular sampling designs, which can be more precise due to a more even distribution of sample points.

In practice, the estimation variance attached to an estimate of abundance is more comfortably interpreted through a coefficient of variation (*CV*), square root of the relative estimation variance, see Chapter 2. Then, under a Gaussian hypothesis, for example, the probability to deviate by more than r (e.g. $r = 15\%$) can be deduced by $p = 2G(-r/CV)$, where G is the standard Gaussian cumulative distribution function.

Application 5.1. Global estimation with a variogram

The estimation variance of the mean over domain V can be computed using the function global(). The following R code (full script in Annex 3) performs the global estimation of herring (*Clupea harengus*) eggs over a spawning bed. The survey design (Annex 2) is made of dredge hauls dispersed more or less evenly over the spawning bed. It is considered that there is no error on the delineation of the limits of the domain V, which are defined by a polygon.

Before performing the global estimation, we must read the data and polygon, select the data inside the polygon, and calculate and model the variogram. The variogram model is isotropic and made of a nugget effect and an exponential model.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herreggs.scot.db.data","db.data")
rg.load("Demo.herreggs.scot.poly.data","poly.data")
rg.load("Demo.herreggs.scot.vario.data","vg")
rg.load("Demo.herreggs.scot.model.vario","vg.fit")

# Data Presentation (left figure)
x1lim<-25.9; x2lim<-26.5; y1lim<-17.0; y2lim<-17.58
plot(db.data,name.prop="eggs",xlab="",ylab="",title="Eggs",
xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim))
plot(poly.data,add=T,lty=1,density=0)


# Variogram and Model Presentation (right figure
plot(vg,xlab="Distance (km)",ylab="Variogram")
plot(vg.fit,add=T)
```

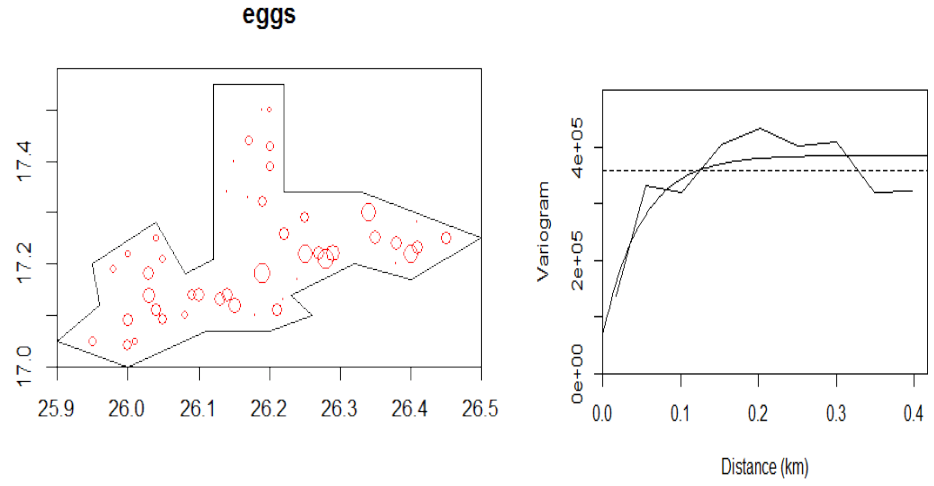


Figure 5.1. Left: proportional representation of herring egg data on a spawning bed delineated by a polygon (data supplied by Marine Scotland Science at the Marine Laboratory, Aberdeen). Right: variogram and its fitted model.

Now, we estimate the zone mean over polygon V and its estimation variance. The estimator considered is the simple data average. The estimation variance has three terms, two of which involve integrals of the variogram, $\gamma(V,V)$ and $\gamma(V,v)$, where v denotes the set of datapoint locations. To compute these terms, a fine grid discretizing polygon V is necessary.

```
# Define the discretization grid
gnx <- 100; gny <- 100
gd.disc <- db.grid.init(obj=poly.data,nodes=c(gnx,gny))
gd.disc <- db.polygon(gd.disc,poly.data)

# Display discretization grid
plot(gd.disc,pch=3,col=1,title="")
plot(db.data,add=T,pch=21)
plot(poly.data,add=T)
```

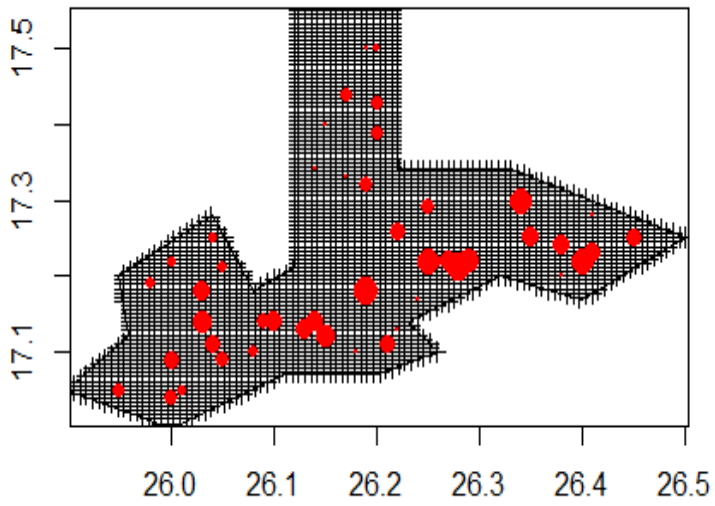


Figure 5.2. Map of a herring spawning bed with a bubbleplot of herring egg density (in red) over a spawning ground. Discretization grid (black) inside the polygon.

```
# Global estimate = arithmetic mean
global.ma <- global(dbin=db.data, dbout=gd.disc, model = vg.fit, uc=c
("1"),
polygon = poly.data, calcul = "arith", verbose=0)
cat("simple mean: ", global.ma$zest," CV.geo: ", global.ma$cv,"\n")

## simple mean:  963.26  CV.geo:  0.08
```

Alternative regular design

Given new data locations, the estimation variance for an alternative sampling design can be calculated using the function global(). For that, a new database is defined that consists of a regular design. Note that we add a dummy variable (zm: data mean), which is not used in the variance calculation, into the new database.

```
# Regular grid design: create

x0 <- 25.9; y0 <- 17.0
dx <- 0.06; dy <- 0.06
nx <- 13;   ny <- 12
db.nw <- db.create(x0=c(x0,y0),dx=c(dx,dy),nx=c(nx,ny))
db.nw <- db.add(db.nw,loctype="z")
db.nw <- db.locate(db.nw,2:3,loctype="x")
db.nw <- db.add(db.nw,z1=rep(0,db.nw$nech)) # dummy
```

```
db.nw <- db.locate(db.nw,4,loctype="z")
db.nw <- db.polygon(db.nw,poly.data)
```

```
# Regular grid design: display
plot(db.nw,pch=3,xlab="km",ylab="km",flag.aspoint=TRUE,name.post=1,
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),title="")
plot(poly.data,add=T,lty=1,density=0)
```

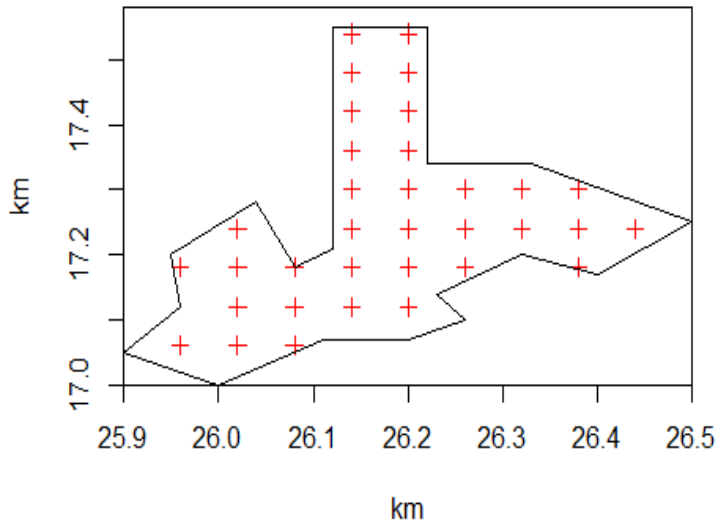


Figure 5.3. Global estimation of herring eggs over a spawning ground. Alternative regular grid design (red crosses).

```
# Regular grid design: global estimation
global.syst <- global(dbin=db.nw, dbout=gd.disc, model = vg.fit, uc=c
("1"),
polygon = poly.data, calcul = "arith", verbose=0)

# Summary of results

tab2 <- rbind(c(global.ma$zest,global.ma$cv,sum(db.data[,5])),
              c(zm,global.syst$sse/zm,sum(db.nw[,5])),
              c(zm,sqrt(zv/sum(db.data[,5]))/zm,sum(db.data[,5])) )
dimnames(tab2) <- list(c("Data","Regular","Random"),c("Mean","CV","NB
"))
print(round(tab2,3))
```

```
##            Mean    CV NB
## Data    963.261 0.080 46
## Regular 963.261 0.074 34
## Random  963.261 0.092 46
```

The alternative regular grid design has 34 stations within the polygon and is more precise ($CV = 0.074$) than the current survey design with 46 stations ($CV = 0.080$). A design made of 46 randomly positioned stations would have the lowest precision ($CV = 0.092$).

5.3 Estimation variance in transitive geostatistics

Transitive geostatistics (Matheron, 1971; Chilès and Delfiner, 2012) can be used to compute the global estimation variance of an abundance $Q = \int z(x)dx$, when the domain is not delineated (e.g. because of diffuse limits), assuming a regular grid (or a random stratified sampling in regular blocks). The grid is supposed to extend beyond the limits and to have a random origin. Transitive geostatistics can be used in 2D (Bez, 2002), but also in 1D (e.g. parallel transects projected on an orthogonal line; Petitgas, 1993a).

The formula of the estimation variance for a regular grid with cell a is, in 1D or using "short" notations:

$$\sigma_Q^2 = |a| \sum g(ka) - \int g(h)dh$$

where $g(h)$ is the transitive covariogram, k are integers, and ka describe the distances between grid points. This formula can be expanded in 2D with 2D notations as follows, the cell being (a_1, a_2):

$$\sigma_Q^2 = |a_1 a_2| \sum\sum g(k_1 a_1, k_2 a_2) - \int\int g(h_1, h_2)\, dh_1 dh_2$$

The formula of the estimation variance for a random stratified sampling in regular blocks with cell a is:

$$\sigma_Q^2 = |a| \left[g(0) - \overline{g(a)} \right]$$

with

$$\overline{g(a)} = \frac{1}{|a|} \int_a \int_a g(y - x)\, dx\, dy,$$

or with 2D notations:

$$\sigma_Q^2 = |a_1 \| a_2| \left[g(0) - \overline{g((a_1, a_2))} \right]$$

Application 5.2. Global estimation of cephalopod abundance with transitive method

The following R lines (full script in Annex 3) show an example of transitive global estimation. The data used here correspond to a regular stratified sampling where one sample is taken at random in each square of a 11 x 11 nautical mile regular grid. They correspond to the cephalopod survey carried by INRH (Institut National de Recherche Halieutique) - Casablanca – Morocco (Faraj and Bez, 2007). See Annex 2 for data details.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.octopus.morocco.db.data","db.data")
rg.load("Demo.octopus.morocco.poly.data","poly.data")
projec.define(projection="mean",db=db.data)


# Compute and fit the covariogram (see sections above for details)
db.data <- infl(db.data, nodes=400, extend=c(6,6), origin=c(-18,20.5)
,
```

```
                polygon=poly.data, plot=F, asp=1,
                xlab="Longitude (n.mi.)", ylab="Latitude (n.mi.)")

lag <- 11 ; nlag <- 20 ; dirvect = 50+c(0,90)
covario.data <- vario.calc(db.data, lag=lag, nlag=nlag, calcul="covg"
, tolang=45,
dirvect=dirvect)
iad0 = covario.data[1]$npas + 1
covario.data[1,1]$sw[iad0] <- 0
covario.data[2,1]$sw[iad0] <- 0
model.covario <- model.auto(covario.data,struct=c(1,2),constraints=co
vario.data@vars,draw=F)

# Global estimate
Q <- sum(db.data[,7]*db.data[,9])

# Estimation variance and coefficient of variation
# Turn OFF the projection not to project the strata dimensions provid
ed in projected units (here nautical miles)
projec.toggle(0)
var.est <- 11 * 11 * (model.eval(model.covario, h=0, as.cov=T) -
           model.cvv(v.mesh=11, model=model.covario, seed=110366, ndi
sc=20))
CV <- round(sqrt(var.est)*100/Q,2)
cat("CV = ",CV,"%\n")

## CV = 18.63%
```

The estimation variance is highly sensitive to the amount of nugget effect. As a matter of fact, the estimation variance is the difference between the value of the covariogram at 0 distance and an average of it over a grid mesh, i.e. using the values of the covariogram model without the nugget effet. A very good approximation of the estimation variance can thus be obtained by using only the nugget effect, provided it is large. Particular care is thus recommended when fitting/choosing the model to not be optimistic in terms of nugget effect and thus in terms of estimation variance; too small a nugget effect leads to an estimation variance that is too small.

5.4 Case of an indicator: the geometric error

The transitive method can be used in the particular case of the indicator of presence $1_{z(x)>0}$. In 2D, the sum of this indicator corresponds to the area of presence $A = \int 1_{z(x)>0}\ dx$. Using regular sampling with mesh size (a1,a2), this area can be estimated by the number N of sample points hitting the area A* = N a1 a2 (Figure 5.4).

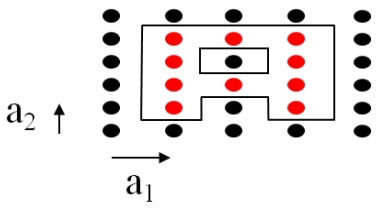


Figure 5.4. Estimating the area of presence from a regular sampling pattern. The sample points in red hit the area, those in black do not. The estimation (in nautical miles) of the area of presence is

the same as the area corresponding to the "positive" cells, i.e. tenfold the area of a cell. The *CV* of this estimation, computed by transitive geostatistics, is 11%.

This estimation is accompanied with an error, due to the real limits of the geometry being unknown between the sample points. In this case the transitive estimation variance can be developed and gives the following relative variance (Matheron, 1971; Chilès and Delfiner, 2012):

$$\frac{\sigma^2_A}{A^2} = \frac{1}{N^2}\left(\frac{1}{6}N_2 + 0.061\frac{N_1^2}{N_2}\right) \text{ with } N_2 \le N_1 \text{ and}$$

$2N_1$ and $2N_2$ the number of linear elements a_1 and a_2 that make the perimeter of the area A^* containing the N points.

In the particular case of Figure 5.4, this gives:

$$N_1 = 6$$
$$N_2 = 4$$
$$\frac{\sigma^2_A}{A^2} = \frac{1}{100}\left(\frac{4}{6} + 0.061\frac{36}{4}\right) = \frac{1.21}{100}$$
$$CV_A = \frac{\sigma_A}{A} = 11\%$$

The method can also be applied in 1D. For example, in the case of parallel transects projected on an orthogonal line, the sum of densities along each transect gives the transect abundance, and the 1D indicator on the orthogonal line says if the transect has a positive abundance or not (Figure 5.5). Summing the indicator along the line gives the 1D extension of the abundance on the line. This can be estimated by the number of positive transects $A^* = N\,a$. The exact location of the two extremities of the extension is unknown. The resulting estimation variance of this geometric error is equal to $\sigma^2_A = \frac{a^2}{6}$ (Matheron, 1971; Petitgas, 1993a), giving $CV_A = \frac{\sigma_A}{A} = \frac{1}{N\sqrt{6}}$

Figure 5.5. 1D example for parallel transects with inter-transect distance *a*.

Each point represents a transect projected on this orthogonal line. Each red point corresponds to a transect having a positive abundance, and each black point is a transect without abundance. The 1D extension of the abundance on the line can be estimated by the number of positive transects. The transitive method can give the variance of this estimation, due to the fact that the exact location of the two extremities of this 1D extension is unknown.

5.5 On the different methods for global estimation variances

The transitive approach allows one to compute the global estimation variance of a sampled abundance $Q = \int z(x)dx$ from a regular sampling design. It does not require the delineation of a domain, but assumes that the population has been sampled to its outer limits.

On the other hand, the intrinsic approach based on the variogram allows to compute the global estimation variance σ_E^2 of the average regionalized variable $Z(V)$ (density in 2D, transect biomass in 1D) within a given domain V. Since the corresponding abundance is $Q = V\,Z(V)$, the estimation variance of Q is simply $V^2\ \sigma_E^2$, and the estimation CV of Q and $Z(V)$ are the same.

The intrinsic approach assumes that the variogram describes the inner behavior of the variable within the domain. However, this approach does not take into account the possible uncertainty on the domain. This is relevant when the domain has to be estimated from the samples. There are cases where the domain over which the abundance is estimated is known in advance, e.g. an ICES statistical square or a management area in which case no uncertainty has to be considered for the domain.

This uncertainty can be assessed by the relative variance of the geometric error of the domain $\frac{\sigma_V^2}{V^2}$ which can be computed from the transitive approach in the case of a regular sampling design (see Section 6.4). It can be used first to complement the estimation variance of $Z(V)$ which becomes $\sigma_E^2 + D^2(o/V)\frac{\sigma_V^2}{V^2}$, where $D^2(o/V)$ is the dispersion variance of the variable within V, represented by its sample variance (Matheron, 1971; Journel and Huijbregts, 1978). The relative estimation variance of $Z(V)$ becomes $\frac{\sigma_E^2}{Z(V)^2} + \frac{D^2(o/V)}{Z(V)^2}\frac{\sigma_V^2}{V^2}$.

Secondly, if there is an uncertainty on the domain V, the CV of $Z(V)$ and of Q are not the same. The relative estimation variance of $Q = Z(V)V$ can be approximated by the sum of the relative estimation variances of $Z(V)$ and of V:

$$\frac{\sigma_Q^2}{Q^2} = \frac{\sigma_E^2}{Z(V)^2} + \frac{D^2(o/V)}{Z(V)^2}\frac{\sigma_V^2}{V^2} + \frac{\sigma_V^2}{V^2} = \frac{\sigma_E^2}{Z(V)^2} + \frac{\sigma_V^2}{V^2}\left(1 + \frac{D^2(o/V)}{Z(V)^2}\right)$$

Application 5.3. Global estimation in 1D for acoustic surveys

In acoustic surveys, acoustic data are recorded continuously along the ship's sailing track. When the survey design is made of parallel, regularly spaced transects, the (global) estimation of population abundance can be performed in one dimension. It suffices to sum fish concentrations along the transect lines and work on the one-dimensional dataset made of fish biomass per transect (Petitgas, 1993a). Here are the main elements for performing the 1D global estimation of an acoustic survey on anchovy. The full demonstration Rscript is in Annex 3, and the data details are in Annex 2.

Before performing the global estimation, the 1D data are read (28 values) and their variogram is computed and modelled.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.anchovy.bob.1d.db.data","db.data")
rg.load("Demo.anchovy.bob.1d.vario.data","vg")
```

```
rg.load("Demo.anchovy.bob.1d.model.vario","vgmod")

# Data Presentation (left figure)
nrad  <- db.data$nsamples                    # Nb of transects
aa <- 1                                      # Inter-transect (arbi-
trary) distance

#  Transform db.data into regular grid
db.datagrid <- db.grid.init(db.data,nodes=nrad,flag.regular=T)
db.datagrid <- migrate(db.data,db.datagrid,flag.fill=2,name="Tr.bio-
mass")

# Display information
plot(db.data,pch=20,type="b",title="Biomass",
     xlab="S <---- Transects ----> N", ylab="Biomass per transect")
plot(db.datagrid,add=TRUE,col="red")

# Display Variogram and Model (right figure)
plot(vg,xlab="Distance",ylab="Variogram")
plot(vgmod,add=T,col="red")
```

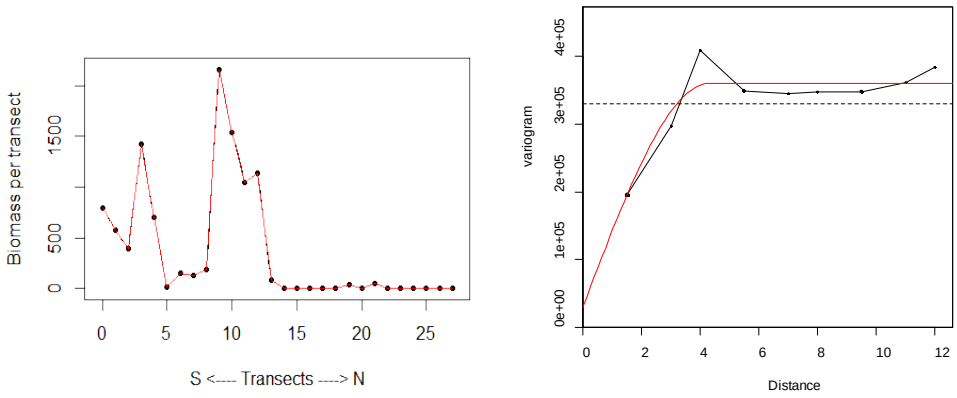


Figure 5.6. Bay of Biscay anchovy global estimation in 1D for an acoustic survey made of regularly spaced parallel transects. Left: fish concentration is summed along east–west transect lines resulting in a 1D dataset of regularly aligned biomass per transect values. Right: variogram and its fitted model (red) for the 1D data. Distance is expressed as a multiplier of the intertransect distance.

The global mean estimate is the simple 1D data average. The 1D domain for the estimation is the 1D segment made of the 27 intertransect distances to which we add one-half a transect-distance away from each extremity. The estimation variance within the 1D domain is computed using the function global(), and the discretization grid is made of 100 points. But the limits of the 1D domain are unknown. Therefore, we calculate a geometric error variance term, considering that the limits are uniformly located within a transect distance away from each extremity (see above). The geometric error variance term adds to the estimation variance within the domain. Here, the geometric error variance represented 4% of the total estimation variance. The total estimation *CV* is 0.107.

```
# Estimation variance 1D
gloa <- global(dbin=db.datagrid, calcul="arith", model=vgmod, ndisc=1
00, verbose=0)

# Geometric error variance: nrad= nb.transect, aa= inter-transect dis
t-ance
```

```
s2 <- var(db.data[,"Tr.biomass"],na.rm=T)*(nrad-1)/nrad
d2geom <- s2*(aa^2/6)/(aa*nrad)^2
# Total estimation CV
cv.tot  <- sqrt(gloa$sse^2+d2geom)/gloa$zest  # Total error CV
cat("Mean=",round(gloa$zest,3), "CVest=",round(cv.tot,3),"\n")

## Mean= 371.735 CVest= 0.126
```

Alternative sampling efforts

The intertransect distance is now varied to evaluate how survey precision changes. For each new intertransect distance, a 1D line grid is defined over the 1D domain, and the estimation variance is computed using the function global(). Note that for each intertransect distance, the geometric error variance term is also computed. The total estimation *CV* increases linearly with intertransect distance. If a survey precision of 0.15 was acceptable, the intertransect distance could be increased.

```
nk <- 9

# Loop on alternative interTransect distances
sse <- numeric(nk)
for (k in 1:nk) {
  ak <- k*0.25*aa                       # new intertransect distance
  nrk <- round(nrad*aa/ak,0)            # new nb transects
  d2geom <- s2*(ak^2/6)/(ak*nrk)^2      # geometric error variance

  # variance of estimation error
  db.datagrid <- db.grid.init(db.data,nodes=nrk,flag.regular=T)
  db.datagrid <- migrate(db.data,db.datagrid,flag.fill=2,name="Tr.bi-
omass")
  d2estim <- global(dbin=db.datagrid,calcul="arith",
                    model=vgmod,ndisc=100,verbose=0)$sse^2
  sse[k] <- sqrt(d2estim+d2geom)
}
plot(0.25*aa*(1:nk),sse/gloa$zest,type="b",
     xlab="Multiplier of Intertransect Distance",ylab="Estimation
CV")
```

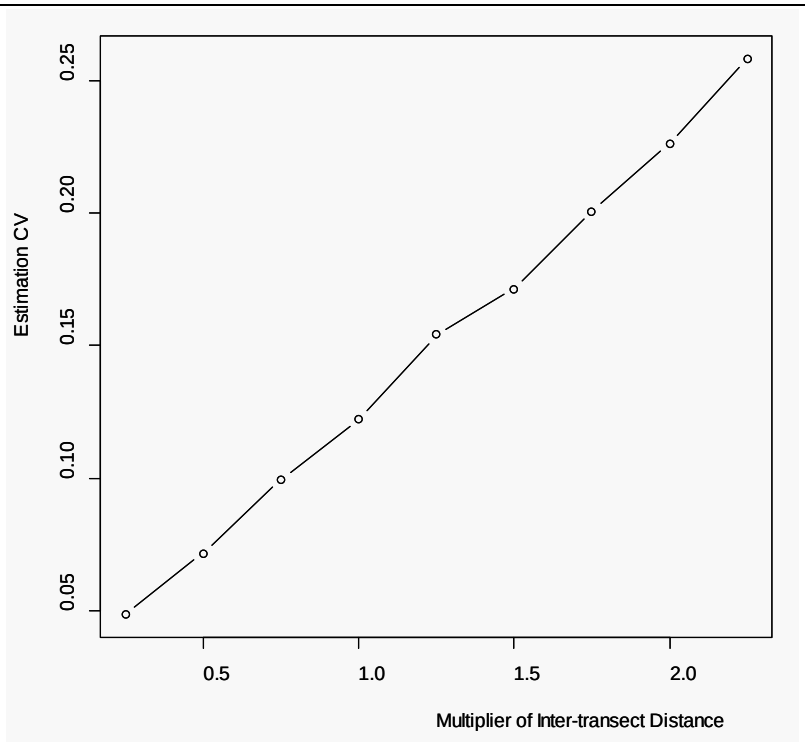


Figure 5.7. Bay of Biscay anchovy global estimation in 1D for an acoustic survey made of regularly spaced parallel transects. Estimation of *CV* as a function of intertransect distance. The current survey corresponds to a multiplier equal to 1.

6 Kriging

Kriging is the best linear unbiased estimator and can be used to estimate the value of a regionalized variable at a target point, the average value of a block, the values at the nodes of a regular grid (for mapping), or the average value over a domain (e.g. a polygon). In particular, kriging can be used for global estimation when data values are to be weighted due to irregular sampling design (the data must not be too numerous).

As a point estimator, kriging is an interpolator obtained as a linear combination of the values measured at sample points. Several other linear interpolation techniques exist which are deterministic approaches to interpolating, for example:

- Moving average: average of the sample values within a neighborhood of the target point;
- Inverse distance: neighbouring sample values are weighted proportionally to the inverse distance (or distance squared) to target point.

Rather than adopting a conventional weighting of the sample data, kriging will choose the weights that make it unbiased and with minimum estimation variance (optimal) given the structural model. Kriging is a probabilistic approach to interpolating between sample points (Matheron, 1971; Chilès and Delfiner, 2012).

Let Z_0 be the target to be estimated (value at a point, or block value, etc). There are different types of kriging, depending on the hypotheses of the model. However, in each case, kriging is a linear combination of data values $Z(x_\alpha)$:

$$Z_0^* = \sum \lambda_\alpha Z(x_\alpha) + \lambda_0$$

where the index α is that of the sample points, λ_0 can be 0 or not, and the weights λ_α must be chosen so that the error has a zero expectation (no bias) and a minimum variance (optimality). The data may be selected within a neighborhood of the target location (see further).

6.1 Simple kriging

"Simple kriging" corresponds to the stationary case with known mean $E[Z(x)] = m$. This mean is the mean of the process, which is also the mean over a very large domain compared to the range. Here, it is supposed to be known through many data over such a domain. If the sampled domain is not large, or if data are not sufficient, such a mean is not known.

Writing that the expectation of the error is 0:

$$E\left(Z_0 - Z_0^K\right) = m - \sum_\alpha \lambda_\alpha m - \lambda_0 = 0 \Rightarrow \lambda_0 = m\left(1 - \sum_\alpha \lambda_\alpha\right)$$

allows for the determination of the constant term λ_0. We can then see that the estimator is a weighted average of the data and of the process mean m (which receives the complementary weight $\lambda_m = 1 - \sum_\alpha \lambda_\alpha$):

$$Z_0^K = \sum \lambda_\alpha Z(x_\alpha) + \lambda_0 = \sum \lambda_\alpha Z(x_\alpha) + \left[1 - \sum \lambda_\alpha\right] m$$

The variance of the error follows from:

$$\text{var}\left[\sum_i \lambda_i Z(x_i)\right] = -\sum_{ij} \lambda_i \lambda_j \gamma(x_i - x_j) \text{ with } \sum_i \lambda_i = 0$$

as:

$$Var\left[Z_0 - \sum_\alpha \lambda_\alpha Z_\alpha - \lambda_0\right] = C_{00} - 2\sum_\alpha \lambda_\alpha C_{\alpha 0} + \sum_\alpha \sum_\beta \lambda_\alpha \lambda_\beta C_{\alpha\beta}$$

where, for example, $C_{\alpha 0} = \text{cov}(Z(x_\alpha), Z_0)$ is either the covariance between datapoint x_α and a punctual target $C(x_0 - x_\alpha)$, or the mean covariance between the datapoint and a discretized target v, in the case of a block or domain $\frac{1}{v}\int_v C(x_0 - x)dx$. The indices α and β are that of the sample points, as β is used in the double summation over sample points.

Kriging weights are solutions of the linear system obtained by minimizing this variance:

$$\sum_\beta \lambda_\beta C_{\alpha\beta} = C_{\alpha 0} \; \forall \alpha$$

When minimized, the estimation variance can be written:

$$\sigma_K^2 = C_{00} - \sum_\alpha \lambda_\alpha C_{\alpha 0}$$

This is called "kriging variance". Note that this is the variance of the error and should not to be confused with the dispersion variance of the kriging estimator.

Because of the linearity of the kriging weight with respect to the right-hand term, kriging directly a set of points (e.g. block, domain) is equivalent to averaging the kriging of the points (when using the same set of data). This only holds for the estimated value. This does not hold for the estimation variance; the kriging variance of the mean density over a spatial domain is not the mean of the kriging variances of the points in this domain.

Some points are to be noted that are also valid for the other types of kriging:

- Kriging is an exact interpolation; it honors the data values at datapoints;
- Kriging is a smoothing interpolator;
- Kriging weights do not depend on data values;
- Multiplying the structure by a constant does not change the estimation, but changes proportionally the estimation variance.

6.2 Ordinary kriging

"Ordinary kriging" corresponds to the case where the mean is unknown or to the intrinsic model. The sum of weights $\sum \lambda_\alpha$ is constrained to be 100% and $\lambda_0 = 0$. This ensures that the error has a mean of 0 whatever the unknown mean, or that the error is defined in the intrinsic random function model characterized by the variogram. Minimizing the variance under the constraint on the sum of the weights gives the following linear system to be solved for the kriging weights:

$$\begin{cases} \sum_\beta \lambda_\beta C_{\alpha\beta} + \mu = C_{\alpha 0} \; \forall \alpha \\ \sum_\beta \lambda_\beta = 1 \end{cases}$$

where μ is a Lagrange parameter introduced for the constraint (the system can be written in terms of variogram instead of covariance, replacing μ by $-\mu$). The indices α and β are those of the sample points.

The kriging variance can be written as:

$$\sigma_K^2 = C_{00} - \sum_\alpha \lambda_\alpha C_{\alpha 0} - \mu$$

Ordinary kriging can, in particular, be used in global estimation to estimate the mean over a domain.

Application 6.1. Global estimation with a variogram, kriging the global mean over a polygon

We come back to the demonstration Rscript in Annex 3, which performs the global estimation of herring eggs over a spawning bed.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herreggs.scot.db.data","db.data")
rg.load("Demo.herreggs.scot.grid.disc","gd.disc")
rg.load("Demo.herreggs.scot.poly.data","poly.data")
rg.load("Demo.herreggs.scot.vario.data","vg")
rg.load("Demo.herreggs.scot.model.vario","vg.fit")

# Data Presentation (left figure)
x1lim<-25.9; x2lim<-26.5; y1lim<-17.0; y2lim<-17.58
plot(db.data,name.prop="eggs",xlab="",ylab="",title="Eggs",
xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim))
plot(poly.data,add=T,lty=1,density=0)

# Variogram and Model Presentation (right figure
plot(vg,xlab="Distance (km)",ylab="Variogram")
plot(vg.fit,add=T)
```

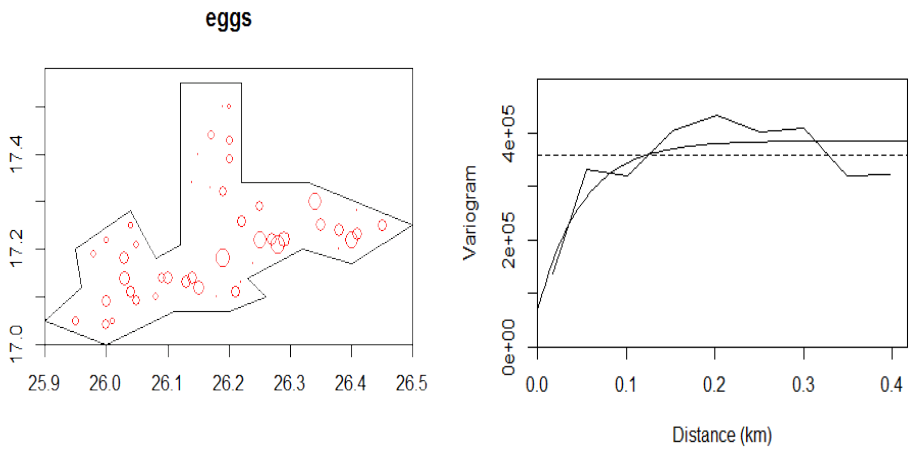


Figure 6.1. Left: proportional representation of herring egg data on a spawning bed delineated by a polygon (data supplied by Marine Scotland Science at the Marine Laboratory, Aberdeen). Right: variogram and its fitted model.

The mean over the bed in the polygon (regional mean) is now estimated by ordinary kriging. This is done using the function global() with appropriate input parameters.

We use the same discretization grid of the domain as previously. The kriging weights are saved and added into the data base for spatial representation.

```
# Global estimate = kriged mean
global.mk <- global(dbin=db.data, dbout=gd.disc, model = vg.fit, uc=c
("1"),
                    polygon = poly.data, calcul = "krige", flag.wgt=T
RUE,
                    verbose=0)
# Display kriging weights
db.data <- db.add(db.data,global.mk$wgt,loctype="w")
plot(db.data,name.prop="w",title="Global Kriging weights")
plot(poly.data,add=T)
```

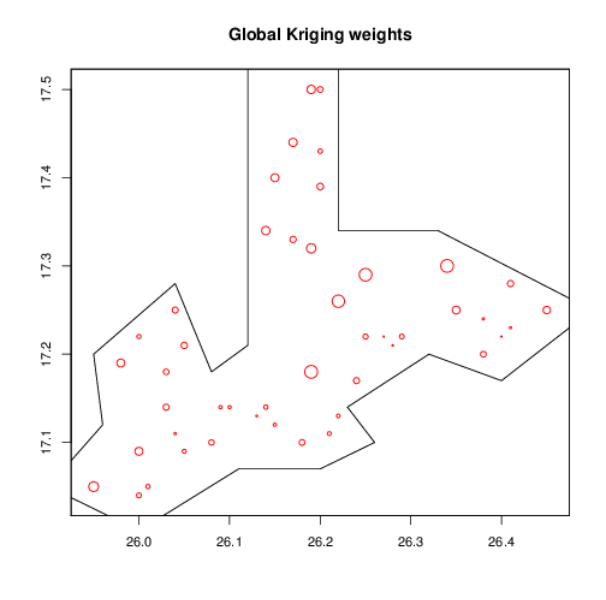


Figure 6.2. Herring eggs over a spawning ground. In red, the kriging weights when kriging the global mean over the delineated domain.

Kriging weights depend on the location of samples and also on the variographic structure between samples. Here, they are smaller where sample points are closer and larger in areas where there are less data. This results in slightly down-weighting large values, and thus the kriged mean is a bit lower than the simple data average. The kriging variance is also lower than the estimation variance when estimating the global mean by the data average.

```
cat("Kriged zone mean: ", global.mk$zest," CV.geo: ", global.mk$cv,"\
n")
Kriged zone mean:  946.6976  CV.geo:  0.07502062
```

6.3 Comparing simple kriging and ordinary kriging

Simple kriging is a weighted average of the data values and of the process mean $m = E[Z]$ of the random function. The possible sparseness of data, if any, is compensated by a weight given to the mean. The linear regression of Z_0 on its kriging Z_0^{SK} is equal to

Z_0^{SK} (line with slope 1 going through the origin). If the true regression [conditional expectation $E(Z_0 \mid Z_0^{SK})$] is linear, this guarantees, for example, that areas considered as "rich" (e.g. high kriged fish density) are, on average, as rich as predicted (absence of conditional bias: $E\left[Z(x) \mid Z(x)^K \right] = Z(x)^K$).

When the mean of the stationary random function is unknown, it can be estimated optimally by kriging as a weighted average of data values, similarly to ordinary kriging. This is the kriging of the mean m^K, with its kriging variance $Var(m^K)$. It can be shown that ordinary kriging is nothing but simple kriging in which the process mean is replaced by its kriged mean m^K:

$$Z_0^{OK} = \sum \lambda_\alpha^{SK} Z(x_\alpha) + \left[1 - \sum \lambda_\alpha^{SK} \right] m^K$$

If the weight of the mean in SK $\lambda_m = 1 - \sum_\alpha \lambda_\alpha^{SK}$ is close to 0, there is no difference between simple and ordinary krigings, as the mean (either known in simple kriging or kriged in ordinary kriging) is not used. When the weight of the mean in simple kriging is not close to 0, the slope of the regression of Z_0 on its kriging Z_0^{OK}, which can be written as:

$$a = \frac{Cov\left(Z_0, Z_0^{OK} \right)}{Var\left(Z_0^{OK} \right)} = 1 - \lambda_m \frac{Var(m^K)}{Var\left(Z_0^{OK} \right)}$$

will be different from 1, and generally less than 1. This means that areas considered as rich after their kriging are, on average, less rich than predicted, while areas considered as poor are less poor. Kriging does not smooth enough, and there is conditional bias: $E\left[Z(x) \mid Z(x)^K \right] \neq Z(x)^K$.

In practice, stationarity is often only local, so that ordinary kriging ("kriging with unknown mean") based on neighbouring data values is frequently preferred. However, it is recommended to use a large enough neighbourhood to make a sufficient smoothing and avoid, as much as possible, conditional bias.

6.4 Choosing the neighbourhood

Kriging can be performed using all data ("unique neighbourhood") or only the data in the neighbourhood of the target location (moving neighbourhood for a kriged map). Theoretically, it is better to use all data, but this is not always possible when data are numerous and not desirable when stationarity (of the variable or its increments) is only local. Then data must be selected from a neighbourhood of the target location. The choice of the neighbourhood in kriging may not be easy. Selecting the most appropriate subset of neighbouring samples can be done by their number, by the maximal distance from target, by angular sector from target, or by a mixture of criteria. The use of angular sectors ensures that data in different directions from the target are selected; this is particularly useful when datapoints are densely located along lines, e.g. acoustic transects. The presence of a "screen effect" (datapoints having 0 weights because they are screened by datapoints closer to the target location) can help choosing a neighbourhood without loss of information, but a nugget effect tends to suppress such screen effects. Looking at the weights of data away from the target, or at their influence on the kriging variance, may help choosing a sufficient neighbourhood. In ordinary kriging,

when the simple kriging weight of the mean is not low, a slope of regression of Z conditional on Z^* close to 1 should be preferred to avoid conditional bias.

6.5 Cross-validation

Cross-validation may help choosing a method (e.g. type of kriging), a variogram model or a neighbourhood. Its principle is the following. At each datapoint, remove the sample value and estimate its value from the other samples. Then, compare real to estimated values using statistics on the errors:

$$\varepsilon_\alpha = Z(x_\alpha) - Z(x_\alpha)^*$$

and on the normalized errors:

$$\varepsilon_\alpha^R = \frac{Z(x_\alpha) - Z(x_\alpha)^*}{\sigma_\alpha}.$$

In principle, the best choice in terms of prediction of values corresponds to the lowest mean squared error:

$$\frac{1}{n}\sum {\varepsilon_\alpha}^2 = \frac{1}{n}\sum \left[Z(x_\alpha) - Z(x_\alpha)^* \right]^2 .$$

Furthermore, the best choice in terms of prediction of variance corresponds to the mean squared normalized error:

$$\frac{1}{n}\sum \left(\varepsilon_\alpha^R\right)^2 = \frac{1}{n}\sum \frac{\left[Z(x_\alpha) - Z(x_\alpha)^* \right]^2}{\sigma_\alpha^2}$$

which is the closest to 1. Remember that multiplying a variogram model by a constant does not change the kriging weights and also the estimation of the values, whereas it directly impacts the prediction of the variance.

Application 6.2. Kriging herring eggs on a spawning bed, neighbourhood, cross-validation and mapping

We come back to the herring egg data over a spawning bed (full Rscript in Annex 3; data details in Annex 2), on which we performed global estimation previously. Now we consider kriging for mapping, and for that, different neighbourhoods are analyzed. Two criteria are used: the weight of the mean, and the decrease in kriging weights with distance from the target point to be kriged. The data comprise 46 egg counts at dredged locations (db.data) and a polygon (poly.data) delineating the spawning bed. The variogram model (vg.fit) has been calculated previously.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herreggs.scot.db.data","db.data")
rg.load("Demo.herreggs.scot.grid.disc","gd.disc")
rg.load("Demo.herreggs.scot.poly.data","poly.data")
rg.load("Demo.herreggs.scot.vario.data","vg")
rg.load("Demo.herreggs.scot.model.vario","vg.fit")

# Data Presentation (left figure)
x1lim<-25.9; x2lim<-26.5; y1lim<-17.0; y2lim<-17.58
plot(db.data,name.prop="eggs",xlab="",ylab="",title="Eggs",
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim))
```

```
plot(poly.data,add=T,lty=1,density=0)

# Variogram and Model Presentation (right figure
plot(vg,xlab="Distance (km)",ylab="Variogram")
plot(vg.fit,add=T)
```

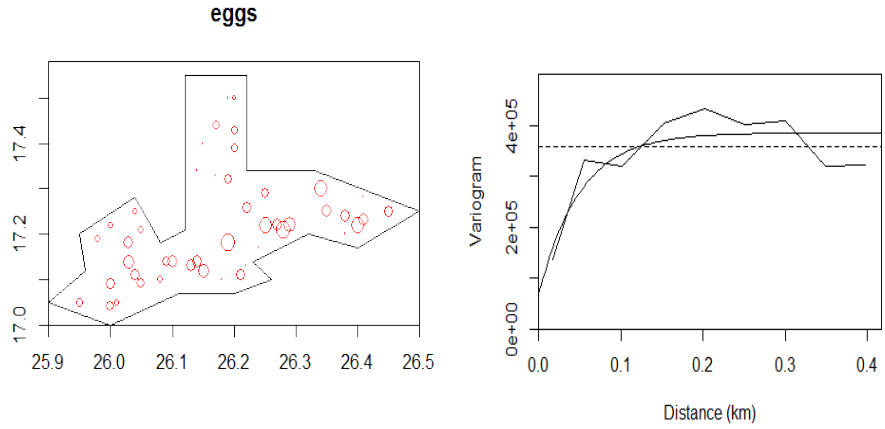


Figure 6.3. Left: proportional representation of herring egg data on a spawning bed delineated by a polygon (data supplied by Marine Scotland Science at the Marine Laboratory, Aberdeen). Right: variogram and its fitted model.

First, we define the grid on which to perform kriging and select its nodes inside the polygon.

```
# Grid on which to perform kriging
x0 <- 25.9; y0 <- 17.0; dx <- 0.05; dy <- 0.05; nx <- 13; ny <-13
db.grid <- db.create(flag.grid=T,x0=c(x0,y0),dx=c(dx,dy),nx=c(nx,ny))
db.grid <- db.polygon(db.grid,poly.data)

# Display grid, polygon and data
plot(db.grid, title="", xlim=c(x1lim, x2lim), ylim=c(y1lim, y2lim),
pch=3)
plot(poly.data,add=T,lty=1,density=0)
plot(db.data,pch=18,add=T,col="black",inches=1.5)
```

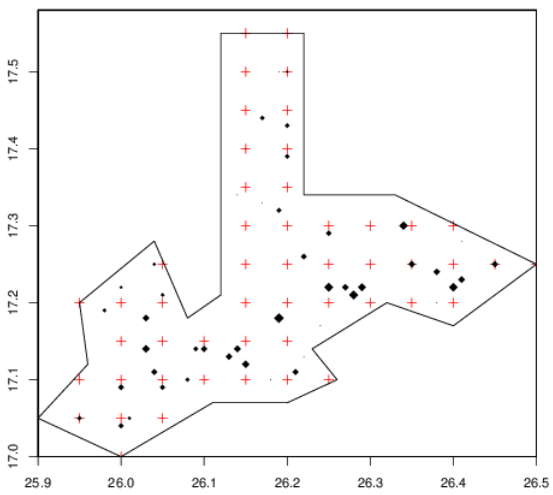


Figure 6.4. Mapping by kriging herring eggs on a spawning ground. Kriging discretized grid (red crosses) with proportional representation of the data (black diamonds) inside the polygon. The blue circles denote the grid points selected for testing different kriging neighbourhoods.

We choose two target points to test different kriging neighbourhoods; one is well surrounded by datapoints (grid point 30 in the southwest), thus well informed for kriging; the other is less well informed (grid point 87 in the northeast).

The process mean is estimated by kriging using function global().

```
mt <- global(dbin=db.data, dbout=gd.disc, model = vg.fit, uc=c("1"),
             polygon = poly.data, calcul = "mean", verbose=0)
```

Two different kriging neighbourhoods are considered: a unique neighbourhood (nei1) involving all datapoints and a moving neighbourhood (nei2) involving 2–8 points within a disc of radius 0.5 km from the target point.

```
nei1 <- neigh.create(ndim=2,type=0)                                    #
unique
nei2 <- neigh.create(ndim=2,type=2,nmini=2,nmaxi=8,radius=0.5) # mov-
ing
```

The function krigetest() provides all details on the kriging system for a given target point. For target point 30 and neighbourhood nei1, the following code allows for the estimation of the weight of the process mean and to access the kriging weights of the samples within nei1.

```
# Kriging parameters for neighbourhood nei1 at grid point 30

# simple kriging
kts <- krigtest(dbin=db.data, dbout=db.grid, model=vg.fit,
neigh=nei1, uc=NA,
                mean=mt$zest, calcul="point", iech0=30, target=NA)

# ordinary kriging
iech0 = 30
```

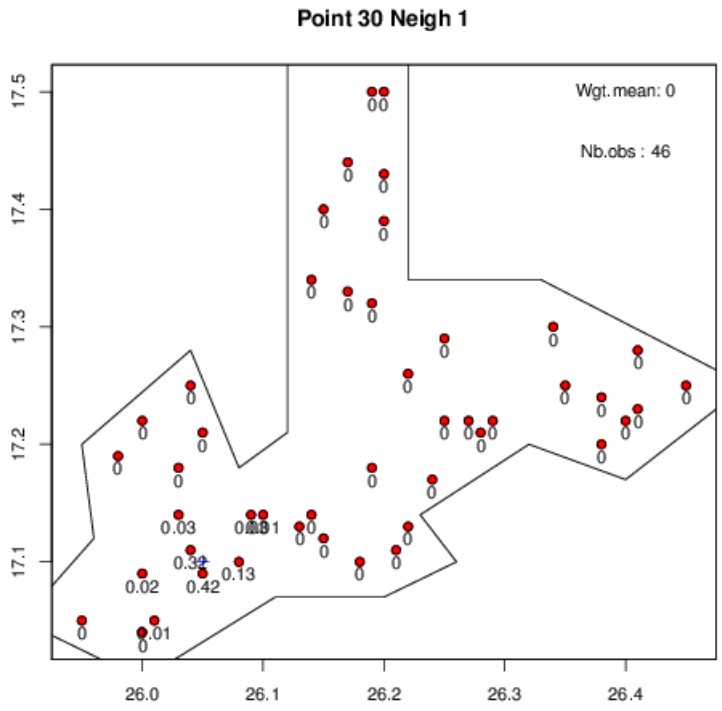
```
nei_rank = 1
kto <- krigtest(dbin=db.data, dbout=db.grid, model=vg.fit,neigh=nei1,
uc=c("1"),
                calcul="point", iech0=iech0, target=NA)

# weight of process mean in neighbourhood
wgt <- kto$wgt[1: kto$nech]
lm  <- 1-sum(wgt)

# nb samples in neighbourhood
ns <- length(wgt)

# Display Kriging parameters
plot(db.data, name.post=1, xlab="", ylab="",
     title=paste("Point",iech0,"Neigh",nei_rank))
points(kto$xyz)
text(x=26.4, y=17.5, paste("Wgt.mean:",round(lm,3)), cex=1)
text(x=26.4, y=17.45, paste("Nb.obs :",ns), cex=1)
plot(poly.data, add=T, lty=1, density=0)
points(x=kto$target[1], y=kto$target[2], pch=3, col="blue")
text(kto$xyz,label=round(wgt,2),cex=1,adj=c(0.5,1.5))
```

Inserting this code into a double loop on the two selected target points (30, 87) and the two neighbourhoods (nei1, nei2) gives the following results. For the well-informed target (point 30), the kriging weights become nil when datapoints are away from the target. For the less well-informed target (point 87), kriging weights decrease with distance to the target, but some distant data are given non-nil kriging weights. The weight of the process mean is larger for the less well-informed target (point 87: 0.27–0.29), while it is lower for the well-informed target (point 30: 0.03).



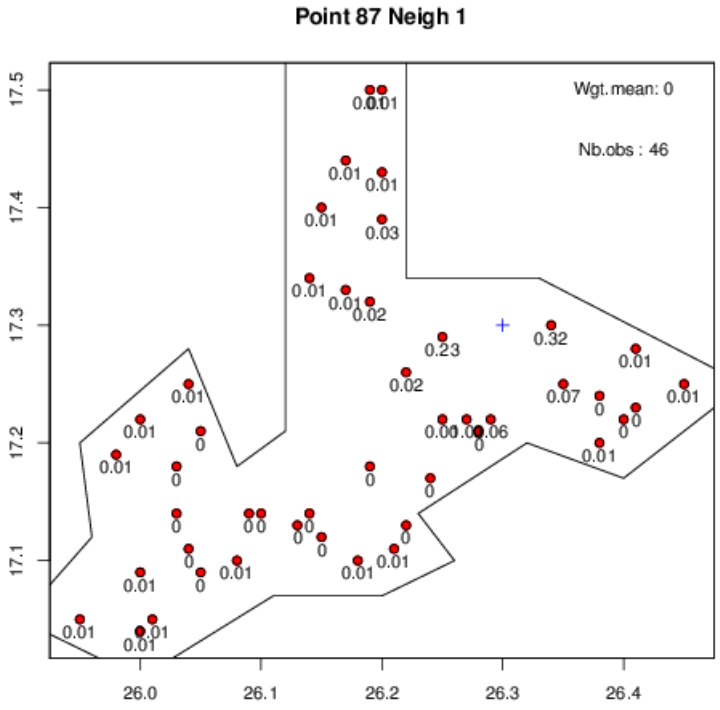
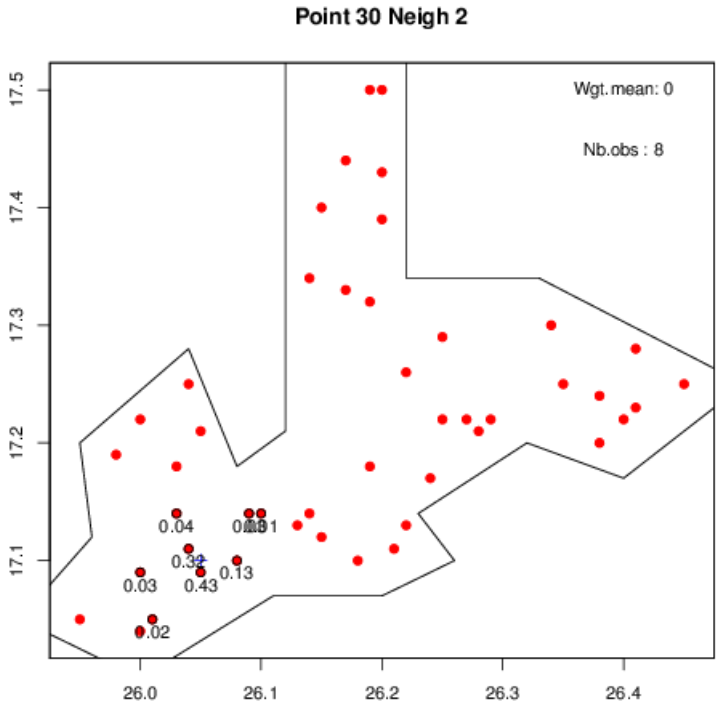


Figure 6.5. Weight of the process mean and ordinary kriging weights for a unique neighbourhood around target points (blue cross) 30 (top) and 87 (bottom).



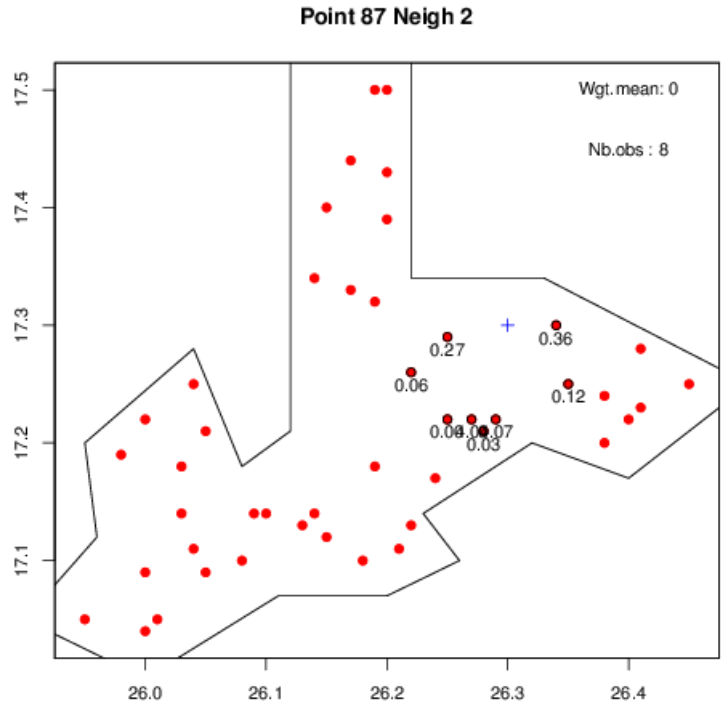


Figure 6.6. Weight of the process mean and ordinary kriging weights for a moving neighbourhood with eight datapoints maximum around target points (blue cross) 30 (top) and 87 (bottom).

The kriging weights of the data close to the target points are similar in unique and moving neighbourhoods and so are the weights of the mean.

Another test of the neighbourhood is cross-validating the data by kriging with that neighbourhood. For that, we use the function xvalid().

```
# Cross-validation
db.data.xv <- xvalid(db=db.data, model=vg.fit, neigh=nei2, uc=c("1"),
mean=NA)

# Mean error (should be close to 0) in percent of zone mean
mean(db.extract(db.data.xv, "Xvalid.eggs.esterr")) / 963.26
## 0.0453

# Mean standardized squared error (should be close to 1)
mean(db.extract(db.data.xv, "Xvalid.eggs.stderr")^2)
## 1.2591
```

Mapping by kriging is now performed with function krige(). We use ordinary point kriging in moving neighbourhood with neighbourhood nei2. The global mean in the map is close to the simple data average. The kriging variance expresses how the sample locations inform the kriging grid.

```
# Ordinary point kriging in moving neighbourhood
kres <- kriging(dbin=db.data, dbout=db.grid, model=vg.fit,
neigh=nei2, uc=c("1"), mean=NA, calcul="point")

# Plot kriged estimates: K.estim
plot(kres,name.image=5,title="K.estim",col=topo.colors(20),xlab="",
     ylab="",xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),pos.legend=5)
```

```
plot(db.data,pch=18,add=T,col="black",inches=1.5)
plot(poly.data,add=T)

# Plot kriging errors: K.std
plot(kres,name.image=6,title="K.std",col=rev(gray((0:100)/100)),
     xlab="",ylab="", xlim=c(x1lim,x2lim),
ylim=c(y1lim,y2lim),pos.legend=5)
plot(db.data,pch=18,add=T,col="black",inches=1.5)
plot(poly.data,add=T)
```

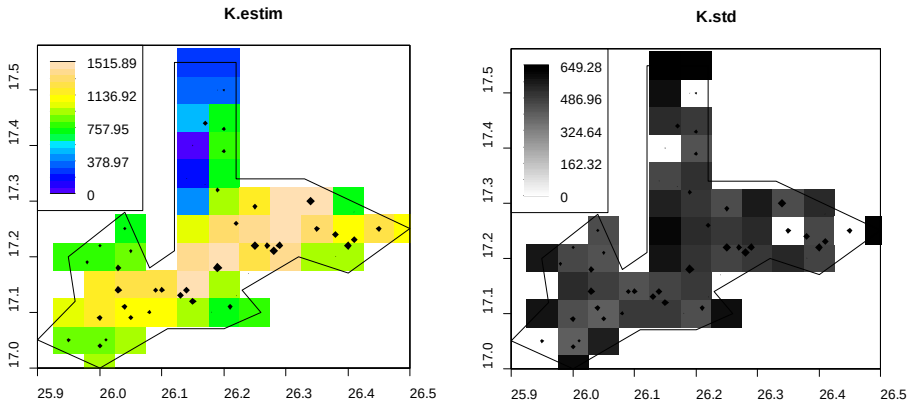


Figure 6.7. Map of herring eggs over a spawning ground obtained by ordinary point kriging with moving neighbourhood (left) and map of the corresponding kriging standard deviation (right).

6.6 Transitive kriging

Although not frequently used, kriging can also be performed in the transitive approach (with $\lambda_0 = 0$). Kriging weights are obtained by minimizing the sum of squared errors, supposing that the target is translated over space with the same relative configuration of data. The sum of kriging weights λ_α can be set to 1, so that the sum of the estimates equals the sum of the true values (non bias). Kriging weights are obtained as the solution of a linear system similar to ordinary kriging, but replacing the variogram or covariance by the transitive covariogram.

Application 6.3. Mapping cephalopod concentrations by transitive kriging

The following R code (full script in Annex 3) is for an example of transitive kriging (Faraj and Bez, 2007). The data used here correspond to a random stratified sampling where one sample is taken at random in each square of a 11 x 11 nautical mile regular grid (see Annex 2). They correspond to the cephalopod survey carried out by INRH (Institut National de Recherche Halieutique) - Casablanca - Morocco. Before performing the kriging estimation, we must calculate and model the covariogram and define a kriging grid and a kriging neighbourhood.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.octopus.morocco.db.data","db.data")
rg.load("Demo.octopus.morocco.poly.data","poly.data")
projec.define(projection="mean",db=db.data)
```

```
# Compute and fit the covariogram
db.data <- infl(db.data, nodes=400, extend=c(6,6), origin=c(-18,20.5)
,
                polygon=poly.data, plot=F, asp=1,
                xlab="Longitude (n.mi.)", ylab="Latitude (n.mi.)")
dirvect = c(50,140)
covario.data <- vario.calc(db.data, breaks=seq(5.5,250,by=11), calcul
="covg", tolang=45,dirvect=dirvect)

model.covario <- model.auto(covario.data,struct=c(1,3,3),constraints=
covario.data@vars,draw=F)
```

The kriging grid is defined in the geographical space. The function db.grid.init() allows covering the data with a regular grid with a given number of discretized points. Then, a selection with the function db.polygon() allows for the selection of gri d points that are in the polygon to avoid spending time kriging outside the polygon. T he grid points will be projected into a set of points to perform transitive kriging in the projected space with a consistent covariogram model. The final output will thus be re presented in the geographical system.

```
projec.toggle(0)
grid.kri <- db.grid.init(db.data,nodes=400,extend=c(6,6),origin=c(-18
,20.5))
grid.kri <- db.polygon(grid.kri,poly.data)
projec.toggle(1)
```

The function neigh.create() allows defining the neighbourhood. Here, we chose to use a moving neighbourhood (type = 2). A minimum of 10 points are required in the neig hbourhood to perform kriging (nmini = 10) and a maximum of 30 is enough (nmaxi = 30). To honour the anisotropy present in the spatial structure and in the covariogram model, we used an anisotropic neighbourhood, i.e. an ellipse. The long axis is radius = 150 nautical miles in direction 50° and the other is 50 nautical miles in the orthogonal direction.

```
# Define rotation matrix consistent with main directions used
rotmat <- matrix(c(cosd(dirvect[1]),sind(dirvect[1]),
                   -sind(dirvect[1]),cosd(dirvect[1])),ncol=2)
neigh.kri <- neigh.create(ndim = 2,type = 2,flag.aniso = T, flag.rota
tion = T,
              nmini = 10, nmaxi = 30, radius = c(150,50), rotmat=rotma
t)

# Transitive kriging
res <- kriging(db.data,grid.kri,model.covario,neigh.kri)

# Threshold possible negative estimates
res[,5][res[,5]<0] <- 0

# Map of the result
projec.toggle(0)
plot(res,col=rainbow(6,start=0.2,end=1),las=1,title="Octopus density
- 1999",asp=0.8)
plot(poly.data,las=1,add=T); map("worldHires", fill=T,col=grey(0.8),a
dd=T)
plot(db.data,las=1,add=T,col=1)
```

```
legend.image(range(db.extract(res,"Kriging.JUV.estim"),na.rm=T),posit
ion="bottomright",col=rainbow(6,start=0.2,end=1),ntdec=0,cex=0.75)
```

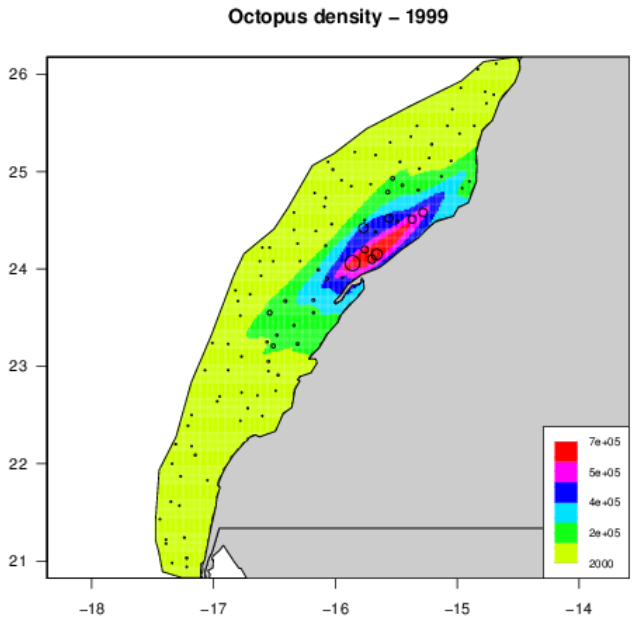


Figure 6.8. Distribution map of octopus density obtained by transitive kriging. Local heterogeneities are smoothed given the amount of nugget effect present in the model. Anisotropy is generated according to the model.

7 Multivariate geostatistics

7.1 Multivariate structural tools

Relations often exist between different regionalized variables in the same domain. For example, fish densities at different ages are often related, or fish attributes (e.g. its length) may be linked to bottom depth, etc.

Basic tools such as scatterplots between two variables are excellent ways to illustrate the relations that can exist between variables. They can be completed by linear regressions, non-linear regressions, conditional distributions, and extended to the case of several variables. However, very often these statistical tools describe the relationships between variables measured at the same location. For example, in a scatter diagram, each fish length value will be plotted against the corresponding bottom depth value at the same location (Figure 7.1).

Application 7.1. Correlating two variables: herring mean length and bottom depth

Multivariate geostatistics makes use of relationships between different regionalized variables. Here is an example on herring mean length and bottom depth collected at the same (trawl) stations around the Shetland (full Rscript in Annex 3, data details in Annex 2). The correlation between herring mean length and bottom depth at trawl stations is measured and displayed using the function correlation(). The (Pearson) correlation coefficient is 0.62.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herring.len.scot.db.data","db.data")
rg.load("Demo.herring.len.scot.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

# Figure: correlation
correlation(db.data,name1="depth",name2="m.length")
## [1] 0.6190425
```

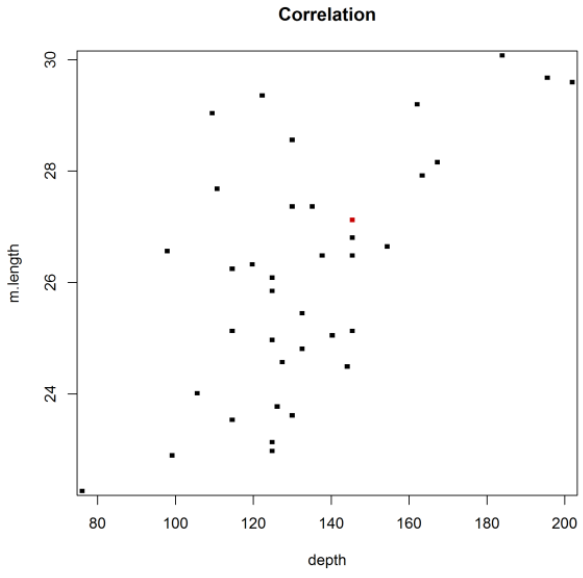


Figure 7.1. Scatterplot between mean length and bottom depth at trawl stations.

The tools of multivariate geostatistics aim at highlighting the structural relations between variables, which includes the relations that may exist between different points. They can be used to improve the estimation or the mapping of one variable using an auxiliary variable, either sampled at the same locations ("isotopic" case, not to be confused with "isotropic") or not ("heterotopic" case). They can also be used to estimate or simulate consistently a set of variables. In particular, these tools will be very useful to estimate consistently the indicators above different thresholds corresponding to a regionalized variable in Chapter 8 (Thresholding and indicators).

Multivariate linear models (intrinsic, stationary) generalize the univariate ones. Simple variograms or covariances are then complemented with their bivariate versions. In an order 2 stationary model, each variable is stationary and the link between, say, variables Z_1 and Z_2, with means m_1 and m_2, is depicted by the cross-covariance, supposed to be a function of h:

$$C_{12}(h) = E\left\{\left[Z_1(x+h) - m_1\right]\left[Z_2(x) - m_2\right]\right\}$$

We have $C_{12}(h) = C_{21}(-h)$, but this is not necessarily equal to $C_{12}(-h) = C_{21}(h)$. In case of a "delay" in space (or time) between variables, the maximum of correlation may correspond to a non-zero distance. Note also that two variables that are statistically uncorrelated ($C_{12}(0) = 0$) may still be spatially correlated ($C_{12}(h)$ non zero for some *h*. This happens particularly when making a statistical factorization of variables; by construction, the factors are uncorrelated only at the same point.

In an intrinsic model, each variable is intrinsic and the link between Z_1 and Z_2 is described by the cross-variogram, supposed to be a function of *h*:

$$\gamma_{12}(h) = 0.5\ E\left\{\left[Z_1(x+h) - Z_1(x)\right]\left[Z_2(x+h) - Z_2(x)\right]\right\}$$

As the expectation of each increment is 0, the cross-variogram is also the covariance between the increments. The cross-variogram is symmetrical in *h*: $\gamma_{12}(h) = \gamma_{12}(-h)$ and cannot reveal a possible delay. It satisfies $\gamma_{12}(0) = 0$, but unlike the variogram, it can present negative values (e.g. in case of substitution of one variable by the other, one decreases when the other increases). Note that the cross-variogram can only be established from datapoints where both variables are known.

7.2 Linear model of coregionalization

Once experimental simple and cross covariances or variograms have been computed from data, they must be fitted with a consistent multivariate model in order to ensure the positivity of the variances of all linear combinations. The linear model of coregionalization is the natural extension of the nested model for one variable. All simple and cross-variograms are modeled using the same basic structural components $\gamma^k(h)$ (indexed by *k* and corresponding, for example, to different ranges):

$$\gamma_{ij}(h) = \sum_k b_{ij}^k \gamma^k(h)$$

For consistency, each matrix $\left(b_{ij}^k\right)$ must be definite positive. In particular, when considering a pair of variables, a component can be present in the simple structure of one or both variables and be absent from their cross-structure, but a component appearing in a cross-structure must necessarily be present in the two simple structures.

Application 7.2. Fitting a linear model of coregionalization on herring mean length and bottom depth

Mapping with cokriging required a multivariate geostatistical model to be fitted to the simple and cross-variograms. The following R code (full script in Annex 3) illustrates how to fit a linear model of coregionalization on herring mean length data and bottom depth around Shetland (data details in Annex 2), these two variables being correlated.

After a projection and a proper assignment of locators in the RGeostats database db.data, simple and cross-variograms are computed from data at stations and fitted using the function vario.calc() and model.auto().

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herring.len.scot.db.data","db.data")
rg.load("Demo.herring.len.scot.poly.data","poly.data")
projec.define(projection="mean",db=db.data)
db.data = db.locate(db.data,"depth","z",1)
db.data = db.locate(db.data,"m.length","z",2)

# Calculate experimental omni-directional variograms and fit a model
vario.data <- vario.calc(db.data)
model.vario <- model.auto(vario=vario.data,struc=c("Nugget Effect","S
pherical"),
                wmode=2,flag.goulard=TRUE,npairpt=F,npairdw=T,inches=0
.08,
                opt.varname=1)
```

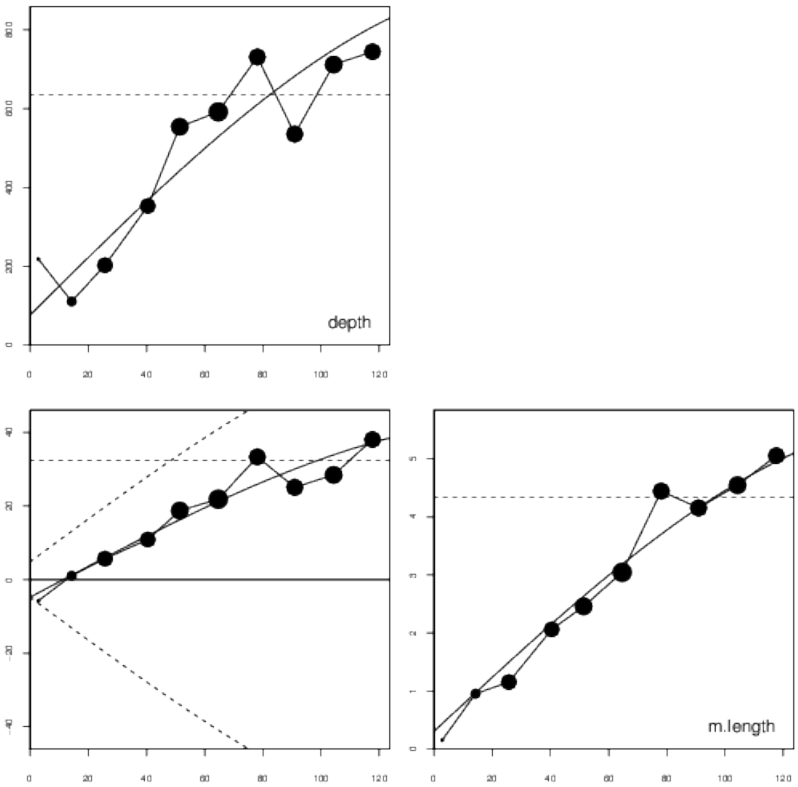


Figure 7.2. Fit of a linear model of coregionalization on bottom depth data and mean fish length. Simple variogams are displayed on the diagonal with the depth data at the top and mean length at the bottom. The cross-variogram is displayed at the bottom left (the oblique dashed lines represent an envelope where the cross-variogram model must lie, given the simple variogram models).

7.3 Cokriging

Cokriging is the extension of kriging to the multivariate case. For example, consider two variables Z_1 and Z_2 informed on two sets of points S_1 and S_2 identical (isotopic case) or not (heterotopic case), and suppose we want to estimate $Z_1(x_0)$. Its cokriging is the best estimation by a linear combination of data:

$$Z_1^*(x_0) = \sum_{S_1} \lambda_{1\alpha} Z_1(x_\alpha) + \sum_{S_2} \lambda_{2\alpha} Z_2(x_\alpha) + \lambda_0$$

Different types of cokriging exist, but in any case, it is unbiased (the expectation of the error is zero) and optimal (the variance of the error is minimized). Simple cokriging corresponds to kriging in a stationary model where the means are known. Ordinary cokriging corresponds to the case where the means are supposedly unknown or to the intrinsic case, then λ_0 is 0, the sum of weights of the target variable λ_1 is set to 1, while the sum of weights of the other variable λ_2 is set to 0. Other variants exist when the means are unknown, but related. In any case, the optimal cokriging weights are the solution of a linear cokriging system of equations.

It is worth paying attention to the order of magnitude of the weights; the weights of Z_1 have no unit, but the weights of Z_2 have the unit: Z_1 unit / Z_2 unit. Moreover, in ordinary cokriging, the weights of Z_2 data sum to 0, hence the presence of negative weights. When associated with large values of Z_2, these can easily lead to a negative estimation for a positive variable Z_1.

Cokriging ensures the consistency of estimates. For example, cokriging a difference $Z_2 - Z_1$ is the difference of the cokrigings of Z_2 and Z_1. In kriging, this is only ensured when the kriging weights are the same for all variables. There is a similar consistency for the cokriging of a sum of variables or of any linear combination of variables.

Application 7.3. Mapping herring mean length by cokriging

Once a linear model of coregionalization is fitted, maps can be produced using cokriging. Here, the following R lines show an example of cokriging using the herring mean length and bottom depth data at trawl stations around Shetland (full script in Annex 3, data details in Annex 2).

Herring mean length and bottom depth are two correlated variables (see above). After a projection and a proper assignment of locators in the RGeostats database db.data, a multivariate variogram model was fitted (see above). Here, it is used for cokriging, and maps were produced (Figure 7.3). The kriging grid and a neighbourhood were defined prior to kriging. As different multivariate kriging methods will be considered later, a common scale was used for the mean length and standard deviation maps for comparing methods results. Cokriging is performed using the function kriging(). Note that as there are multiple locators in the database and the variogram model being multivariate, cokriging is performed automatically.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herring.len.scot.db.data","db.data")
rg.load("Demo.herring.len.scot.grid.kriging","grid.kriging")
rg.load("Demo.herring.len.scot.model.vario","model.vario")
rg.load("Demo.herring.len.scot.neigh.krig","neigh.kriging")
```

```
rg.load("Demo.herring.len.scot.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

db.data = db.locate(db.data,"depth","z",1)
db.data = db.locate(db.data,"m.length","z",2)

# Perform ordinary cokriging
grid.kriging <- kriging(dbin=db.data,dbout=grid.kriging,model=model.v
ario,
                        neigh=neigh.kriging,uc=c("1"),mean=NA,calcul=
"point",
                        radix="CK")

# Perform the various displays
plot(grid.kriging,flag.proj=FALSE,xlim=c(-5,3),pos.legend=5,
     name="CK.m.length.estim",zlim=c(19,40))
map("worldHires",add=T,fill=T,col=8)
plot(db.data,add=T,pch=19,col="black",flag.proj=FALSE)
plot(grid.kriging,flag.proj=FALSE,xlim=c(-5,3),pos.legend=5,
     name="CK.m.length.stdev",zlim=c(0,5))
map("worldHires",add=T,fill=T,col=8)
plot(db.data,add=T,pch=19,col="black",flag.proj=FALSE)
plot(grid.kriging,flag.proj=FALSE,xlim=c(-5,3),pos.legend=5,
     name="CK.depth.estim",zlim=c(80,200))
map("worldHires",add=T,fill=T,col=8)
plot(db.data,add=T,pch=19,col="black",flag.proj=FALSE)
plot(grid.kriging,flag.proj=FALSE,xlim=c(-5,3),pos.legend=5,
     name="CK.depth.stdev",zlim=c(0,30))
map("worldHires",add=T,fill=T,col=8)
plot(db.data,add=T,pch=19,col="black",flag.proj=FALSE)
```

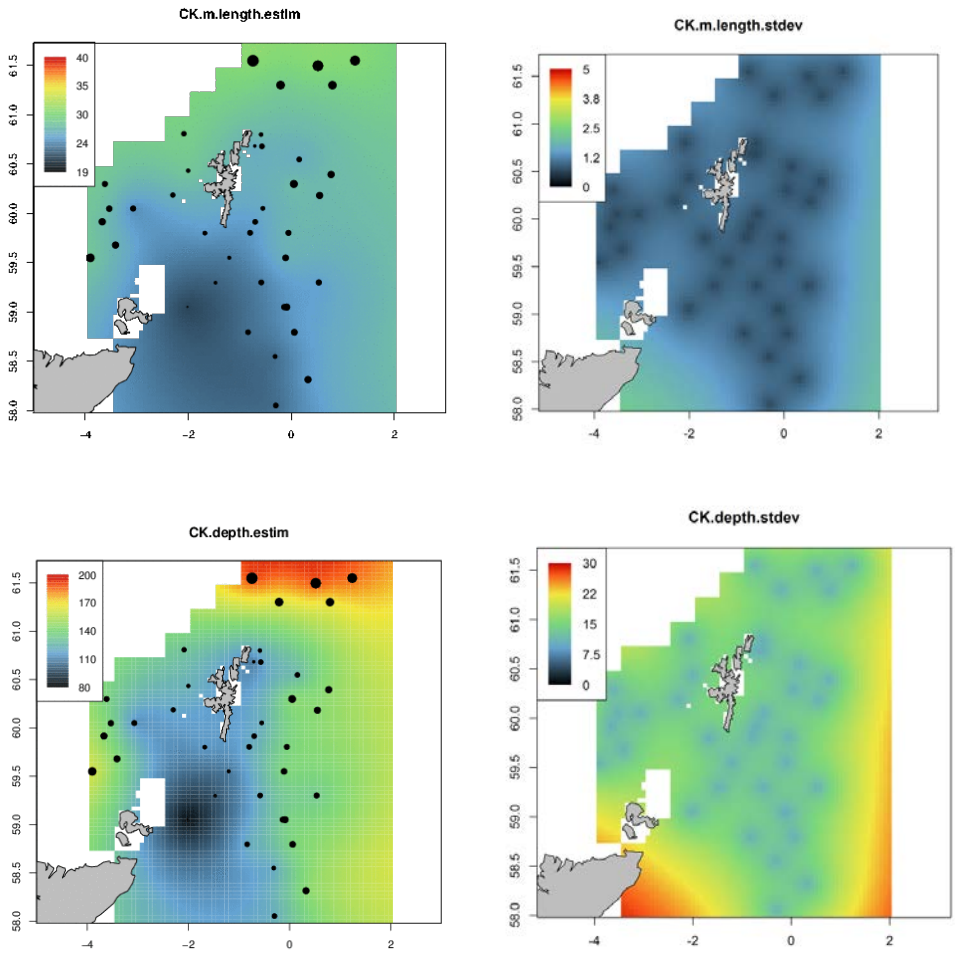


Figure 7.3. Maps of mean length (top left) and bottom depth (bottom left) obtained by cokriging the data collected at the same trawl stations (isotopic multivariate case). Cokriging standard deviation map for mean length (top right) and bottom depth (bottom right).

7.4 Cokriging simplification

Cokriging is more demanding than kriging in terms of calculations. This is particularly true when variables or datapoints are numerous. Then, it can be interesting to use models allowing simplifications, if they are compatible with the structure and data locations.

A first case of simplification corresponds to the "intrinsic correlation" model, characterized by all simple and cross-variograms being similar (proportional to each other). Then, when the variables are known at the same datapoints (isotopic case), cokriging is equivalent to kriging, and all variables have the same kriging weights. In this model, the correlation coefficient between $Z_1(v)$ and $Z_2(v)$ within V can be shown to be constant. It does not depend on the support v and domain V and, therefore, is intrinsic (hence the name of the model). In this model, cokriging coincides with kriging in isotopic cases; hence, it will not do better. By contrast, in isotopic cases, cokriging can be expected to be more precise only when the structures of the different variables are contrasted.

Yet, another case of simplification corresponds to the absence of spatial correlation between variables. Then, cokriging each variable reduces to its kriging (unless the means of the variables are unknown, but related). Such a situation may be met after a geostatistical factorization. The set of variables is transformed into a set of spatially uncorrelated factors, and cokriging can be obtained by kriging the factors.

In particular, this is the case of the so-called model with residual, a hierarchical model where one variable is subordinated to another variable (in the case of two variables). Consider two variables $Y(x)$ and $Z(x)$, the linear regression of $Z(x)$ on $Y(x)$ at the same point X, and the residual $R(x)$ of this regression, so that we have:

$$Z(x) = aY(x) + b + R(x)$$

By construction, this residual has a mean of 0 and is not correlated with $Y(x)$ at the same point, but this is trivial. In the so-called model with residual, the residual R is spatially uncorrelated with Y (i.e. between any pair of points), which is a strong hypothesis. Then Z is subordinated to Y through this additional residual. In practice, this model can be identified by a cross-structure between $Y(x)$ and $Z(x)$ being similar to the structure of $Y(x)$:

$$\gamma_{YZ} = a\gamma_Y$$

The model can be factorized in Y and $R = Z - aY - b$:

$$Z(x) = aY(x) + b + R(x)$$

$$\gamma_Z = a^2\gamma_Y + \gamma_R$$

This leads to a simplified cokriging when Z is known only at datapoints where Y is known:

$$\begin{cases} Y^{CK} = Y^K \\ Z^{CK} = a\ \mathrm{Y}^K + b + R^K \end{cases}$$

In the case where $Y(x)$ is known everywhere (densely sampled), cokriging (simple or ordinary) is equivalent to the kriging of the residual:

$$Z^{CK} = a\ \mathrm{Y} + b + R^K$$

The cokriging neighborhood is "collocated"; it uses Y only at the target and at the Z datapoints.

Collocated cokriging (i.e. cokriging using such a collocated neighbourhood) is a general way to simplify cokriging by restricting the neighbourhood. It is particularly helpful when taking into account an auxiliary variable which is practically known everywhere. Note, however, in general, there is a loss of information in using a collocated neighbourhood. Only in the model with residual is this cokriging optimal (but then it is equivalent to kriging the residuals).

Application 7.4. Mapping herring mean length by collocated cokriging

The following R code shows an example of collocated cokriging for the herring mean length and bottom depth data around Shetland (full script in Annex 3, data details in Annex 2).

We have seen that herring mean length and bottom depth are two correlated variables. We have seen previously how to perform cokriging for mapping using bottom depth at trawl stations where herring mean length was available. However, bottom depth is available at a higher resolution over the area. Here, it is used as an auxiliary variable known everywhere, particularly at the trawl stations and at all the grid nodes. Maps of herring mean length and its standard deviation are now produced by collocated cokriging after adequately assigning locators in the grid database. Collocated cokriging makes use of bottom depth at each target grid point so that the estimated mean length is now driven by bottom depth at every grid node (Figure 7.4). Standard deviation is lower than previously when bottom depth was used at datapoints only.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herring.len.scot.db.data","db.data")
rg.load("Demo.herring.len.scot.grid.kriging","grid.kriging")
rg.load("Demo.herring.len.scot.model.vario","model.vario")
rg.load("Demo.herring.len.scot.neigh.krig","neigh.kriging")
rg.load("Demo.herring.len.scot.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

db.data = db.locate(db.data,"depth","z",1)
db.data = db.locate(db.data,"m.length","z",2)

# Grid management
grid.kriging <- db.locerase(grid.kriging, loctype="z")
grid.kriging <- db.locate(grid.kriging,"depth",loctype="z")

# Perform collocated cokriging
grid.kriging <- kriging(dbin=db.data,dbout=grid.kriging,model=model.v
ario,
                        neigh=neigh.kriging,uc=c("1"),
                        radix="CCK",rank.colcok=c("depth",NA))

# Perform the various displays
plot(grid.kriging,flag.proj=FALSE,xlim=c(-5,3),pos.legend=5,
     name="CCK.m.length.estim",zlim=c(19,40))
map("worldHires",add=T,fill=T,col=8)
plot(db.data,add=T,pch=19,col="black",flag.proj=FALSE)
plot(grid.kriging,flag.proj=FALSE,xlim=c(-5,3),pos.legend=5,
     name="CCK.m.length.stdev",zlim=c(0,5))
map("worldHires",add=T,fill=T,col=8)
plot(db.data,add=T,pch=19,col="black",flag.proj=FALSE)
```

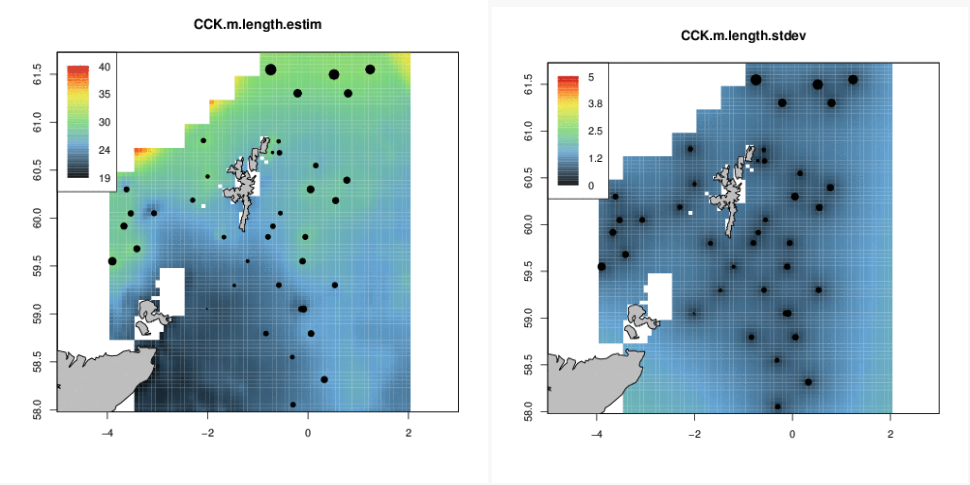


Figure 7.4. Maps of herring mean length obtained by collocated cokriging using bottom depth at each grid node point (left) and the associated standard deviation (right). The map of the mean length is locally driven by the high-resolution bottom depth covariate. The standard deviation for collocated cokriging is lower than for ordinary cokriging due to bottom depth considered as being known exactly everywhere.

7.5 External drift kriging and universal kriging

Kriging of the residuals can be extended to the case where the mean of the variable is a known varying function of X:

$$Z(x) = m(x) + R(x)$$

and when the variability of the residual $R(x) = Z(x) - m(x)$ is independent from the function $m(x)$:

$$Z(x)^K = m(x) + R(x)^K$$

For example $m(x)$ may be the result of a multiple regression of $z(x)$ on a set of environmental variables at the same location. More simply, $m(x)$ may depend linearly on a known auxiliary variable $f(x)$ or on coordinates:

$m(x,y) = a\,x + b\,y + c$ (with 2D notations) or $m(x) = a\,f(x) + b$.

A popular variant of residual kriging in the case where: $m(x) = a\,f(x) + b$. This is external drift kriging (Chilès and Delfiner, 2012). This is not exactly cokriging, but has more flexibility than residual kriging, as it considers that the coefficients of the regression a and b are unknown (e.g. they could vary slowly in space). These are filtered out by setting appropriate conditions on the kriging weights:

$$\sum_{\alpha} \lambda_{\alpha} = 1$$
$$\sum_{\alpha} \lambda_{\alpha} f(x_{\alpha}) = f(x_0)$$

In this external drift kriging, $Z(x)$ is driven by $f(x)$ through the drift $a\,f(x) + b$.

Universal kriging (Matheron, 1971) can be seen as a variant of external drift kriging when the drift depends linearly from the coordinates, e.g. with 2D notations:

$$m(x,y) = a\,x + b\,y + c$$

Appropriate conditions are imposed on the kriging weights in order to filter out the coefficients *a*, *b*, and *c*, of the drift:

$$\sum_{\alpha} \lambda_{\alpha} = 1$$
$$\sum_{\alpha} \lambda_{\alpha} x_{\alpha} = x_0$$
$$\sum_{\alpha} \lambda_{\alpha} y_{\alpha} = y_0$$

External drift kriging and universal kriging are common ways to handle a drift or trend in so-called non-stationary geostatistics. In particular, external drift kriging is capable of taking advantage of driving auxiliary variables that are known everywhere. Fisheries applications of kriging with external drift include the use of a functional relationship between fish mean length and bottom depth (Rivoirard *et al*., 2000) and fish concentration and time of day (Rivoirard and Wieland, 2001).

Application 7.5. Mapping herring mean length by kriging with external drift

The following R code shows an example of kriging herring mean length around Shetland using bottom depth as external drift (full script in Annex 3, data details in Annex 2).

First, a linear regression of mean length on bottom depth is performed using the function regression() (Figure 7.5). Then, a variogram model is fitted on the residuals (Figure 7.6).

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herring.len.scot.db.data","db.data")
rg.load("Demo.herring.len.scot.grid.kriging","grid.kriging")
rg.load("Demo.herring.len.scot.model.vario","model.vario")
rg.load("Demo.herring.len.scot.neigh.krig","neigh.kriging")
rg.load("Demo.herring.len.scot.poly.data","poly.data")
projec.define(projection="mean",db=db.data)

db.data = db.locerase(db.data,"z")
db.data = db.locate(db.data,"depth","z",1)

# Regression
correlation(db.data,"depth","m.length",flag.regr=T,
            title="Regression",ylab="Mean length",xlab="Depth")
## [1] 0.6190425

db.data <- regression(db.data, name1="m.length",names="depth", flag.d
raw=F)
vario.data <- vario.calc(db.data)
model.vario <- model.auto(vario.data,str=c(1,3),npairdw=T,inches=0.08
)
```

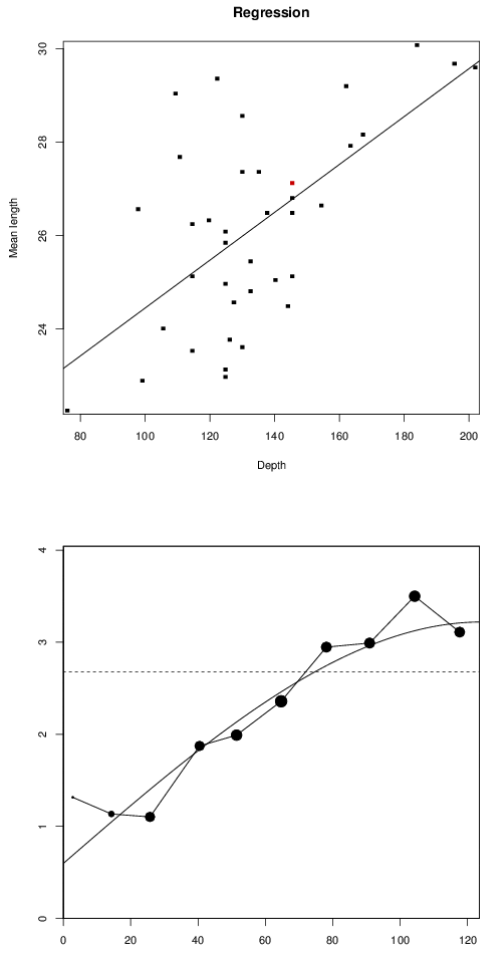


Figure 7.5. Linear regression of mean length on bottom depth and variogram model of the residual.

Then, locators need to be set adequately. In the database, herring mean length is the variable under study with locator "z", while bottom depth is the drift with locator "f". In the kriging grid database, bottom depth is valued at every node and has locator "f" also. Then, kriging with external drift is performed using the function kriging(), and maps of herring mean length and its standard deviation are produced (Figure 7.6). Note that the parameterization of the universality conditions, `uc=c("1","f1")`, in function kriging(), is compatible with the external drift being a linear regression. In comparison to collocated cokriging, the external drift technique allows the parameters of the regression to vary over space. It produces a map of herring mean length that is spatially driven by a functional relationship with the covariate (here a linear regression), rather than just by the values of the covariate, as in collocated cokriging. However, the disadvantage is a higher standard deviation map, due to more flexibility and parameters in the model.

```
# Data management
db.data <- db.locate(db.data,7,loctype=NA)
db.data <- db.locate(db.data,6,loctype="z")
db.data <- db.locate(db.data,5,loctype="f")

# Grid management
grid.kriging <- db.locerase(grid.kriging,"z")
grid.kriging <- db.locate(grid.kriging,"depth","f")
```

```
# Perform kriging with external drift
grid.kriging <- kriging(dbin=db.data,dbout=grid.kriging,model=model.v
ario,
                        neigh=neigh.kriging,uc=c("1","f1"),mean=NA,ca
lcul="point",
                        radix="KED")

# Perform the various displays
plot(grid.kriging,flag.proj=FALSE,xlim=c(-5,3),pos.legend=5,
     name="KED.m.length.estim",zlim=c(19,40))
map("worldHires",add=T,fill=T,col=8)
plot(db.data,add=T,pch=19,col="black",flag.proj=FALSE)
plot(grid.kriging,flag.proj=FALSE,xlim=c(-5,3),pos.legend=5,
     name="KED.m.length.stdev",zlim=c(0,5))
map("worldHires",add=T,fill=T,col=8)
plot(db.data,add=T,pch=19,col="black",flag.proj=FALSE)
```

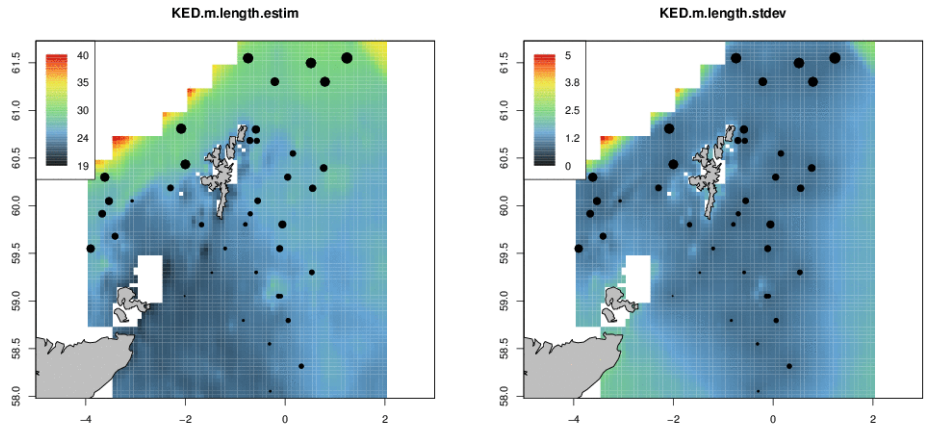


Figure 7.6. Map of mean fish length (left) and standard deviation (right) obtained by kriging with a linear external drift on bottom depth. In comparison to collocated cokriging, the map of the mean length is spatially driven by its regression on the bottom depth at the cost of only a small deterioration in standard deviation.

8 Thresholding and indicators

The geostatistical tools presented in the previous chapters (variogram, kriging, etc) correspond to what is called "linear geostatistics". In particular, estimates of a regionalized variable Z are linear estimates obtained by linear combinations of data. This linear approach may be insufficient in some cases. For example, when we want an estimator which is not linear in Z (e.g. because of high values) or when we want to estimate a non-linear function of Z (e.g. a 0/1 indicator of presence $1_{Z(x)>0}$ or the exceedance over a threshold Z, that is $1_{Z(x)>z}$). In this chapter, we will consider the indicator approach for such non-linear issues. The transformed Gaussian model, which is another approach particularly suited for simulations, will be developed in Chapter 9 on geostatistical simulations.

8.1 Indicator of a set

The indicator of a geometric set A (e.g. presence of fish or a rocky seabed) is a variable equal to 1 (presence) or 0 (absence) at a point. Its average over an area gives the proportion of the set A in this area (e.g. proportion of block occupied by fish).

This indicator variable can be modeled by a 0/1 random function $1_A(x)$, indicator of the random set A (set whose shape and location are considered as random). The expectation of $1_A(x)$ corresponds to the probability that x belongs to A. In the stationary case that will be considered here, this probability p_A does not depend on x and is the same everywhere. We have:

non-centered covariance:

$$K_A(h) = E[1_A(x)1_A(x+h)] = P(x \in A,\ x+h \in A)$$

centred covariance:

$$C_A(h) = \mathrm{cov}[1_A(x), 1_A(x+h)] = K_A(h) - \left(p_A\right)^2$$

variogram:

$$\gamma_A(h) = \frac{1}{2} E([1_A(x+h) - 1_A(x)]^2) = P(\text{one point } \in A,\ \text{the other } \notin A)$$

In particular, the non-centered covariance gives the probability that two points separated by h belong to A, and the cross-variogram gives the probability that one point belongs to A, but not the other one.

The estimator of an indicator at a target point from datapoints has the meaning of a conditional probability of being in the set, even if it is not always really a probability. For example, the kriging of an indicator can take values lower than 0 or larger than 1.

8.2 Indicators of several sets

It is important to make the distinction between:

- independent sets (being independent, they can overlap);
- disjoint sets (they cannot overlap and, therefore, cannot be independent);
- nested sets (one is included in the other).

Geometric sets can be defined from a continuous variable Z(x), such as a concentration, fish density, and so on. For example, one can consider the set made by the values ex-

ceeding a given cutoff or the set of values between two cutoffs. More generally, discretizing such a variable into disjoint classes $[0, z_1[, [z_1, z_2[, [z_2, z_3[, ..., [z_n, \infty[$ provides an example of spatial sets that are disjoint (each set corresponding to the points belonging to a class). Considering cumulated classes $[0, \infty[, [z_1, \infty[, [z_2, \infty[, ..., [z_n, \infty[$ gives sets that are nested.

In the case of disjoint sets (*A,B,C,..*), the cross-variograms are negative, and if $K_{AB}(h) = E[1_A(x)1_B(x+h)] = P(x \in A,\ x+h \in B)$ is symmetrical in *h*, then the opposite of the cross-variogram:

$$\gamma_{AB}(h) = \frac{1}{2} E[1_A(x+h) - 1_A(x)][1_B(x+h) - 1_B(x)]$$

is the probability that one point belongs to *A* and the other to *B*. The opposite of the ratio between the cross-variogram γ_{AB} and the variogram γ_A is the conditional probability of "meeting *B* when leaving *A*" (Rivoirard, 1994):

$$\frac{-\gamma_{AB}(h)}{\gamma_A(h)} = P(x+h \in B \mid x \in A, x+h \notin A)$$

When this probability does not depend on *h* and, therefore, is constant, the chances of meeting *B* when leaving *A* are the same everywhere outside *A*; we say that there is no edge effect in the distribution of *B* outside *A*. On the contrary, the set *B* tends to be positioned close to the borders of the set *A* if the probability decreases with *h*, and away from the borders of the set *A* if the probability increases with *h.* Now consider nested sets, e.g. $A_2 \subset A_1 \subset A_0$. If $K_{A_1A_2}(h)$ is symmetrical, the cross-variogram $\gamma_{A_1A_2}(h)$ is the probability that one point is inside A_2 and the other outside A_1. The ratio between the cross-variogram $\gamma_{A_1A_2}(h)$ and the variogram $\gamma_{A_1}(h)$ is the conditional probability of meeting A_2 when entering A_1:

$$\frac{\gamma_{A_1A_2}(h)}{\gamma_{A_1}(h)} = P(x+h \in A_2 \mid x \notin A_1, x+h \in A_1)$$

When this probability does not depend on *h* and, therefore, is constant, the chances to meet A_2 when entering in A_1 are the same everywhere in A_1; there is no edge effect in the distribution of A_2 within A_1. On the contrary, the set A_2 tends to be positioned close to the borders of the set A_1 if the probability decreases with *h*, and is away from the borders of the set A_1 if the probability increases with *h.*

Hence, such ratios between cross-variogram and simple variogram for disjoint or nested sets can be used to describe the joint arrangement of different sets.

When sets correspond to a discretized concentration (e.g. the disjoint classes corresponding to low, medium, or high values, or equivalently their cumulated nested classes), different types of arrangements can be distinguished.

When there is no edge effect at all, the structures of all indicators are the same. The probability to meet a given class when leaving another one does not depend on the distance *h*. This corresponds to a mosaic model with independent valuations; in this model, the domain is partitioned into tiles and each tile is given a value according to the same probability distribution and independently of the other tiles.

If there is no edge effect upward, the probability to meet a high value when leaving low values does not depend on the distance *h*. This corresponds to a hierarchical model from low to high values. First comes the spatial distinction between low values and all other values; then, where values are not low, the high values can be anywhere. Note that edge effects downward exist in this model; when leaving high values, medium values are preferentially met. In this model, there is a destructuring of high values (more spatial variability than the union of medium and high values), as often observed for skewed distribution concentrations. From this point of view, this model with no edge effect upward is more realistic than the reverse model with no edge effect downwards.

Edge effects upward and downward is typical of diffusion-type models (e.g. Gaussian model), where medium values are met when going from low to high or from high to low values.

Application 8.1. Exploring border effects upward among a range of indicator sets

In survey data, a few high values often represent a large percentage of the total abundance. The indicator approach is helpful to understand the spatial arrangement of these values among other medium and low values. Hence, the use of ratios between cross-variogram and simple variogram for indicator sets to describe the joint arrangement of different sets. Here, we show how the approach can be implemented on data. The full demonstration Rscript is in Annex 3. The data are the acoustic survey data on anchovy in the Bay of Biscay (Annex 2).

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.anchovy.bob.2d.db.data","db.data")
rg.load("Demo.anchovy.bob.2d.poly.data","poly.data")

# Data display (left figure)
y1lim <- 43.3; y2lim <- 47; x1lim <- -4.5; x2lim <- -1
plot(db.data,name="ENGR.ENC",pch=1,asp=1.2,inches=5,col="black",
      xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),ti-
tle="",flag.proj=FALSE)
plot(poly.data, add=T, lty=1, density=0)
map("worldHires",add=T,fill=T,col=8)

# Histogram (right figure)
hist(db.data[,"ENGR.ENC"],breaks=100,col="grey",
```

```
    xlab="Biomass of anchovy (tonns/n.mi.^2)",main="")
```

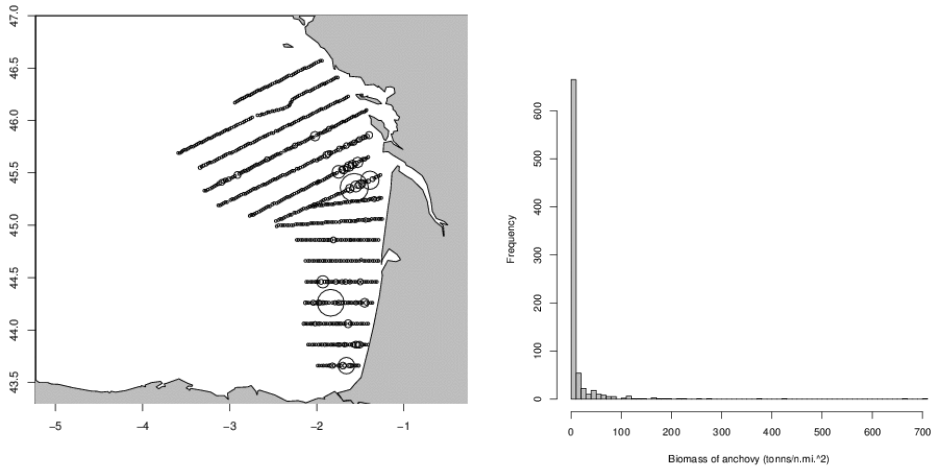


Figure 8.1. Anchovy concentrations ("ENGR.ENC" tonnes nautical mile^{-2}) in the Bay of Biscay. How are the high values positioned relative to medium and low values? Left: proportional representation of the data showing spatial arrangement of high values relative to other values. Right: skewed histogram of the data

We first start by discretizing the data and defining cumulated nested classes. Based on data percentiles, six classes are defined. The first threshold defines the set of strictly positive values. The last class is defined by a top threshold.

```
# Create indicator variables into the RGeostats database
zi <- 150 # top threshold
zcut <- c(quantile(db.data[,"ENGR.ENC"][db.data[,"ENGR.ENC"]!=0],
          seq(0,0.8,0.2)),zi)
my.limits <- limits.create(zcut=zcut,flag.zcut.int=F)
```

```
my.limits
```

```
## Number of classes =  6
## Class  1  : [ 0.0005610691 ,+Inf[
## Class  2  : [ 0.1 ,+Inf[
## Class  3  : [ 1.692 ,+Inf[
## Class  4  : [ 7.706 ,+Inf[
## Class  5  : [ 22.734 ,+Inf[
## Class  6  : [ 150 ,+Inf[
```

The six class indicators are added into the database db.data with multiple locators from "z1" to "z6".

```
db.data <- db.indicator(db.data, limits=my.limits)
```

After projecting the data, we now compute variograms and cross-variograms of the indicators using the function vario.calc(). Note that as there are multiple locators ("z1" to "z6") in db.data, variograms and cross-variograms are calculated automatically.

```
# Define the projection
projec.define(projection="mean",db=db.data)
```

```
# Compute simple and cross variograms of indicator variables
lag <- 5; nlag <- 20; dirvect <- 0
vg <- vario.calc(db.data,lag=lag,nlag=nlag,dirvect=dirvect)
```

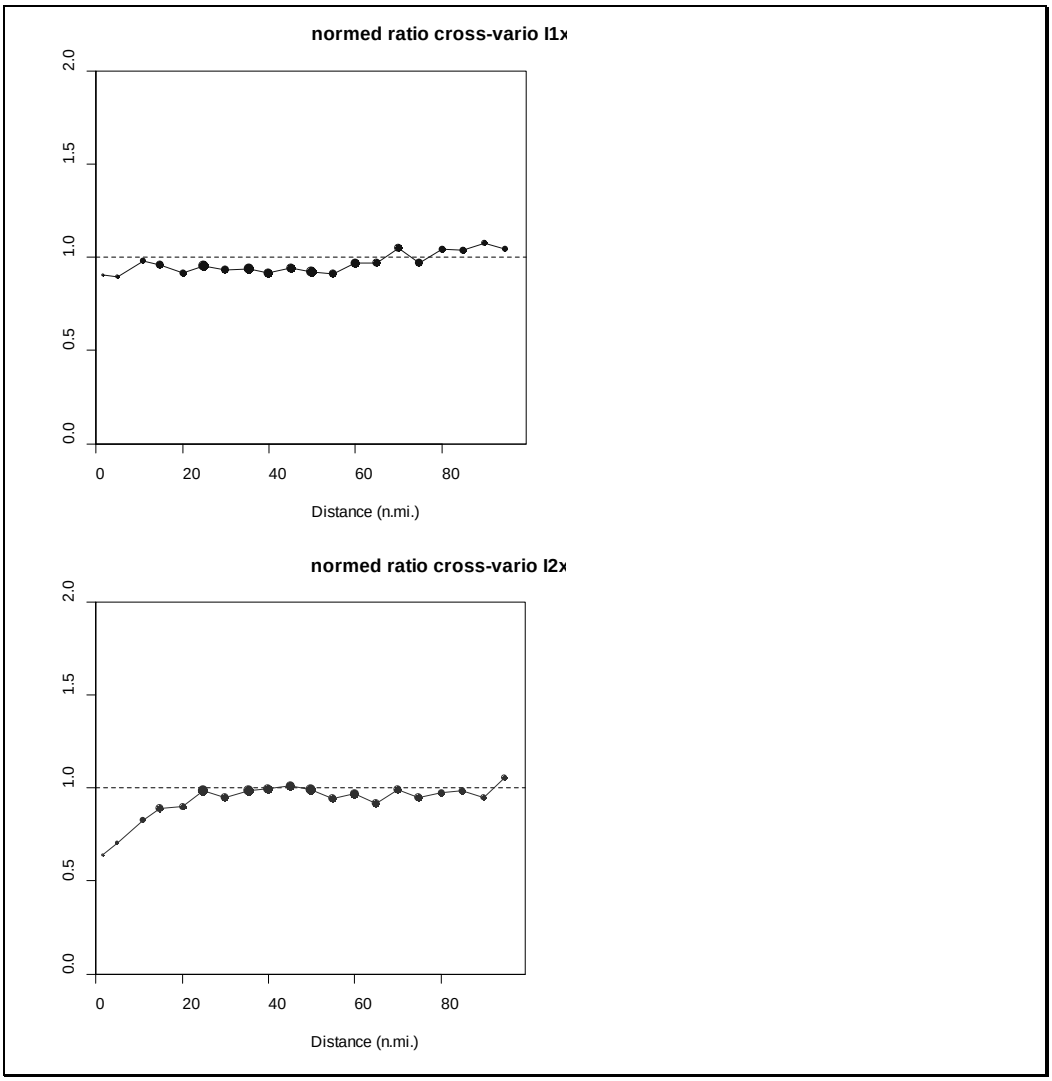
To explore for border effects upward, we compute the ratios for each cross-variogram $\gamma_{i,(i+1)}$ divided by the variogram of the lower cutoff $\gamma_{i,i}$. This is done with the function vario.transfo().

```
# Compute variogram ratios: cross variogram / first simple variogram
vgr <- vario.transfo("v1", vario1=vg, oper="g12/g1")
```

The result is a matrix of cross-variogram/variogram ratios. We are interested in visualizing the first subdiagonal line under the diagonal. The variogram ratios are standardized (sills equal to one).

```
# Visualization of normed cross-variogram/variogram ratios
for (i in 1:(length(zcut)-1)) {
   x11()
   j <- i+1
   plot(vgr, varcols=i, varcols2=j, inches=0.05, flag.norm=T,
npairdw=T, npairpt=F,
        col=grey(seq(0,1,.1))[j], ylim=c(0,2),
        main=paste("normed ratio cross-vario I",i,"xI",j,"/ vario I",
i,sep=""))
}
```

A flat ratio indicates that the set *Ai* defined by the higher cutoff is positioned at random within the set of the lower cutoff (no border effect). In contrast, when the ratio increases to reach a sill, the set defined by the higher cutoff tends to be positioned away from the borders of the set of the lower cutoff, meaning that spatial transitions in values are progressive (border effect). The figure shows that the sets A3 within the sets A2, A3 within A4, and A5 within A4 are spatially arranged with border effects that are smaller and smaller upward, from 25 to 10 nautical miles. In contrast, sets A6 seem to be positioned within A5, with virtually no border effect. Such spatial arrangements are compatible with a progressive spatial destructuring of higher values. Sets A2 within A1 also show no border effects, compatible with an independence between field limits and random function non-zero values.


normed ratio cross-vario l1x
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
Distance (n.mi.)
normed ratio cross-vario l2x
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
Distance (n.mi.)

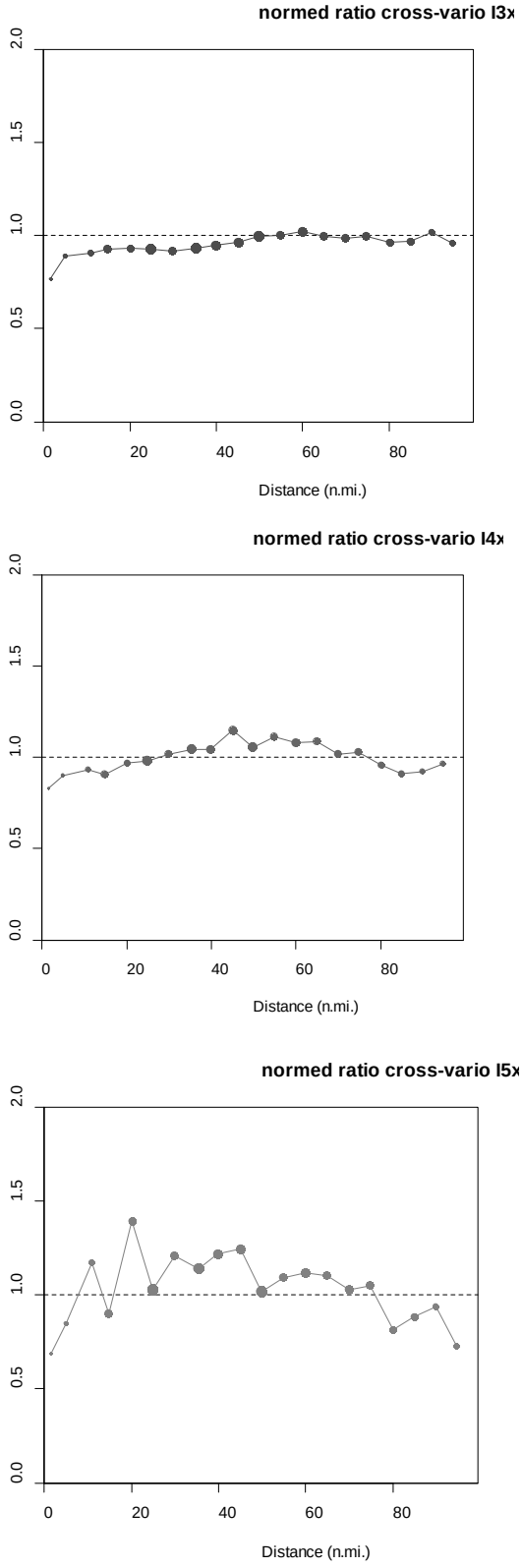


Figure 8.2. Ratios of cross-variograms/variograms exploring the spatial arrangement between nested indicator sets.

8.3 Indicator cokriging

From a theoretical point of view, it is equivalent to discretize a variable into disjoint classes or cumulated nested classes, but the second option is preferable in practice (structures are more visible). Indicator cokriging (also known as disjunctive kriging: Rivoirard, 1994) requires the modeling of simple and cross structures. This may be done using a linear model of coregionalization. However, such a linear model may not be adapted (e.g. when the continuity at the origin of the cross-structures is higher than that of the simple ones). Moreover, as:

$$E[1_{Y(x)\geq i}] = P[Y(x)\geq i]$$

$$Cov[1_{Y(x)\geq i}, 1_{Y(x+h)\geq j}] = P[Y(x)\geq i, Y(x+h)\geq j] - P[Y(x)\geq i]P[Y(x+h)\geq j]$$

modeling simple and cross structures is theoretically equivalent to modeling the bivariate distributions $\left(Z(x), Z(x+h)\right)\ \forall h$. However, linear models of coregionalization of indicators are not necessarily consistent with bivariate distributions (e.g. they could give probabilities to be above a threshold that increases when the threshold increases). Hence, the current use of more appropriate models. Note, however, that, just like indicator kriging, indicator cokriging can provide estimated indicators that fall outside [0, 1].

In the case of the mosaic model with independent valuations, since all structures are the same, cokriging reduces to kriging.

When there is no edge effect upward, the indicators can be factorized into "indicator residuals", and cokriging is obtained by kriging these factors (see applications to pelagic acoustic survey data in Petitgas, 1993b; Petitgas *et al.*, 2016). Let $\left(z_1, z_2, z_3, ..., z_n\right)$ be the discretizing thresholds, $\left(zm_1, zm_2, zm_3, ..., zm_n\right)$ be the mean values of $Z(x)$ within the classes $[0, z_1[, [z_1, z_2[, [z_2, z_3[, ..., [z_n, \infty[$ and A_j be the geometrical set corresponding to $Z(x)\geq z_j$. Each residual is obtained as:

$$R_i(x) = \left[\frac{1_{A_i}(x)}{p_{A_i}} - \frac{1_{A_{i-1}}(x)}{p_{A_{i-1}}}\right]$$

where $p_{A_i} = E\left[1_{A_i}(x)\right] = P[Z(x)\geq z_i]$ except for the most important one in the hierarchy, the first one, deduced from the first indicator:

$$R_1(x) = \left[\frac{1_{A_1}(x)}{p_{A_1}} - 1\right]$$

Each indicator can be obtained from the previous indicators through:

$$\begin{aligned}\frac{1_{A_{i+1}}(x)}{p_{A_{i+1}}} &= 1 + \left[\frac{1_{A_1}(x)}{p_{A_1}} - 1\right] + \left[\frac{1_{A_2}(x)}{p_{A_2}} - \frac{1_{A_1}(x)}{p_{A_1}}\right] + ... + \left[\frac{1_{A_{i+1}}(x)}{p_{A_{i+1}}} - \frac{1_{A_i}(x)}{p_{A_i}}\right] \\ &= 1 + R_1(x) + R_2(x) + ... + R_{i+1}(x) \\ &= \frac{1_{A_i}(x)}{p_{A_i}} + R_{i+1}(x)\end{aligned}$$

This can be used to obtain, successively, the cokriging of each indicator from indicator residuals kriging. The indicators for disjoint classes can be deduced from:

$$1_{z_i \le Z(x) < z_{i+1}} = 1_{Z(x) \ge z_i} - 1_{Z(x) \ge z_{i+1}}$$

and $\sum zm_i 1_{z_i \le Z(x) < z_{i+1}}$ gives the corresponding estimate of $Z(x)$.

The indicator residual model is an example of a so-called isofactorial model, where indicator cokriging, i.e. disjunctive kriging, is obtained by kriging factors. Other isofactorial models exist in non-linear geostatistics corresponding to diffusion-type models (discrete diffusion, gauss, gamma, etc). All of these models correspond to models for bivariate distributions. Indicator cokriging has the meaning of a conditional probability, but the estimated probability can fall outside [0, 1]. The case of the Gaussian model is, however, particular, as it can give access directly to consistent conditional probabilities, as will be seen in the next chapter.

Application 8.2. Multivariate analysis of the indicators of pelagic fish densities

The following R code (full script in Annex 3, data details in Annex 2) shows an example of multivariate structural analysis of indicators (Bez and Braham, 2014). These are built by discretizing acoustic densities (acoustic backscatter of all pelagic fish). The survey was carried by IMROP (Institut Mauritanien de Recherche en Océanographie et des Pêche) - Nouadhibou – Mauritania.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.acoustic.maur.db.data","db.data")
rg.load("Demo.acoustic.maur.poly.data","poly.data")
projec.define(projection="mean",db=db.data)
```

Indicator variables corresponding to fish density exceeding cutoffs are defined using the function limits.create(). In this function, the argument flag.zcut.int allows choosing either indicators of intervals (flag.zcut.int = TRUE) or of exceeding particular values (flag.zcut.int = FALSE)). The cutoffs considered correspond to the quartiles of the histogram of the positive densities, thus to four sets of small, medium small, medium large, and large values.

```
zcut <- as.numeric(quantile(db.data [,5][db.data[,5] >0]))
my.limits <- limits.create(zcut=zcut[-5],flag.zcut.int = F)
db.data <- db.indicator(db.data ,my.limits)

# Variography in two directions (along and across acoustic transects)
projec.toggle(1)
lag <- c(5,10) ; nlag=15 ; dirvect <- c(0,90)

# Mean annual variogram

vario.data <- vario.calc(db.data, lag=lag, nlag=nlag,
dirvect=dirvect,
              opt.code=1, tolcode=0)

# Fit a Linear Model of Coregionalization.
# Use of functions that are linear at the origin e.g. spherical or
exponential
model.vario <- model.auto(vario.data, struct = c(1,3,3,2,2), wmode=2,
draw=F)
plot(vario.data,opt.varname=1,cex.varname=0.8)
plot(model.vario, vario=vario.data, add=T)
```

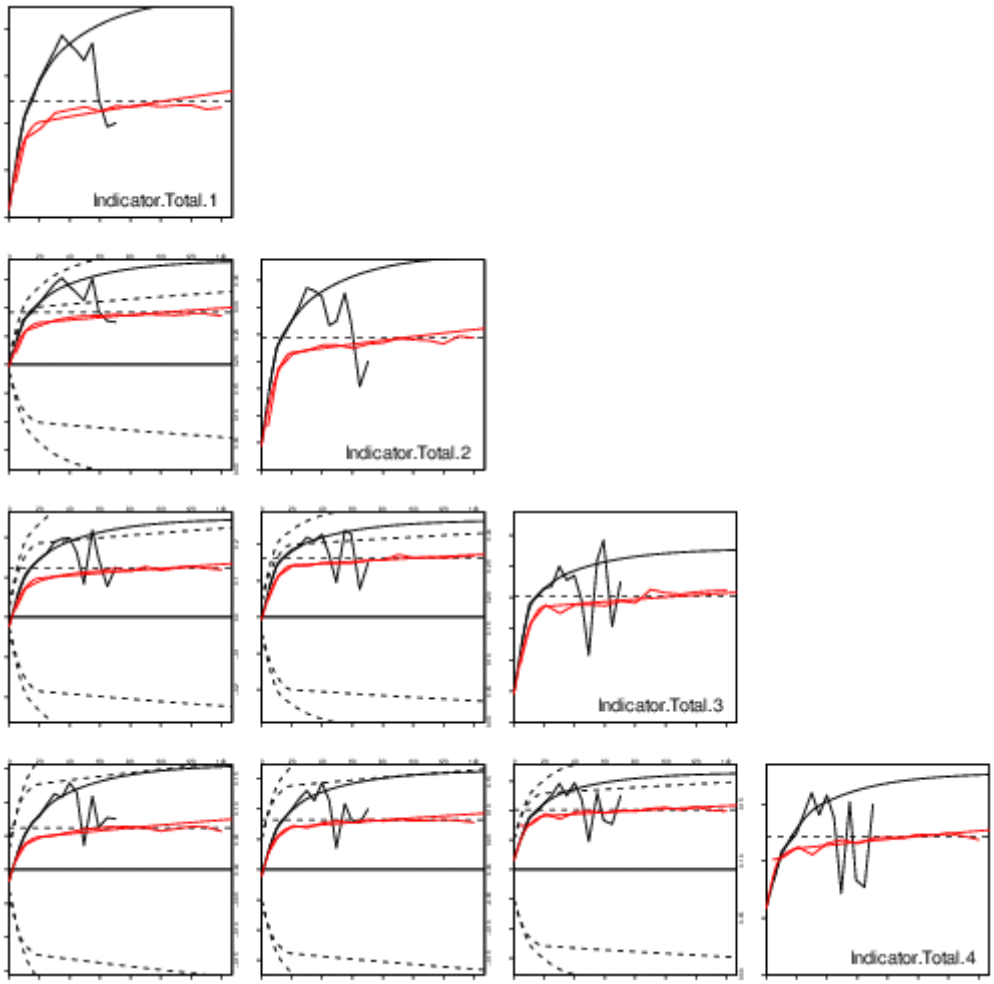


Figure 8.3. Empirical variograms and cross-variograms of indicators and corresponding linear model of coregionalization. Anisotropy in the computation is driven by the orientation of the acoustic transects (along transect in black and across transect in red). Simple variograms are on the diagonal (from the first indicator top left to the fourth one bottom right). The cross-variogram in row *i* and column *j* describes the spatial structure common to indicator *i* and indicator *j*.

One can see that the percentage of nugget effect increases with the rank of the indicator variable. This is part of the standard phenomenon of destructuring of the high grades; the sets corresponding to the very large fish densities being more erratic than the sets made of the densities above the median or the first quartile. Cross-structures show clear cross-correlations between indicators, justifying the recourse to multivariate geostatistics. These are very much driven by simple structures, which could lead to one of the possible simplifications of cokriging mentioned in the text (not done here). Cross-structures show no or very little nugget effects. This means that short scale structures are not correlated between density levels.

```
# Co-Kriging: definition of a kriging grid and a neighbourhood
# Co-Kriging is perfomed for the data of the first year.
projec.toggle(0)
grid.kri <- db.grid.init(poly.data,margin=10,nodes=150)

grid.kri <- db.polygon(grid.kri,poly.data)
neigh.kri <- neigh.create(type=2,ndim=2,nmini=10,nmaxi=50,radius=60)
```

```
# Performs kriging (might take 1-2 minutes)
projec.toggle(1)
kri.1 <- kriging(db.sel(db.data,an==1), grid.kri, model.vario,
neigh.kri)


# truncation of estimations to [0,1]
for(i in 1:model.vario$nvar){
    rank = db.ident(kri.1,paste0("*Indicator*",i,"*estim"))
    kri.1[,rank][kri.1[,rank] < 0] <- 0
    kri.1[,rank][kri.1[,rank] > 1] <- 1
}

# Mapping the results for the first and the fourth indicator varia-
bles
plot(kri.1,asp=1,zlim=c(0,1),col=rain-
bow(4,start=0.2,end=1),flag.proj=FALSE,
     xlab="Longitude (°)",ylab="Latitude (°)")
plot(poly.data,add=T)
map("worldHires",add=T)
legend.image(c(0,1),position="bottomleft",col=rain-
bow(4,start=0.2,end=1),
             ntdec=2,cex=0.75)
```

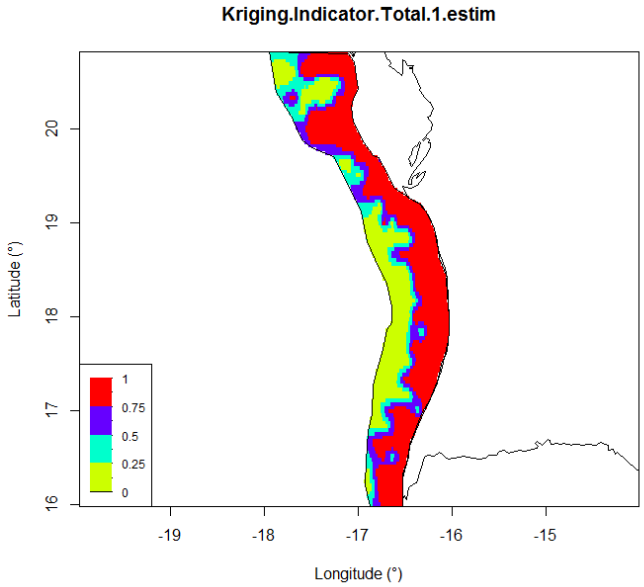


Figure 8.4. Cokriged estimate of the first indicator, i.e. estimation of the probability for the acoustic energy to be positive. The area of presence is clearly next to the coastline, with some excursions to the shelf edge.

```
plot(kri.1,name.image=8,asp=1,zlim=c(0,1),col=rainbow(4,start=0.2,end
=1),
     xlab="Longitude (°)",ylab="Latitude (°)",flag.proj=FALSE)
plot(poly.data,add=T)
map("worldHires",add=T)
legend.image(c(0,1),position="bottomleft",col=rainbow(4,start=0.2,end
=1),
```

```
ntdec=2,cex=0.75)
```

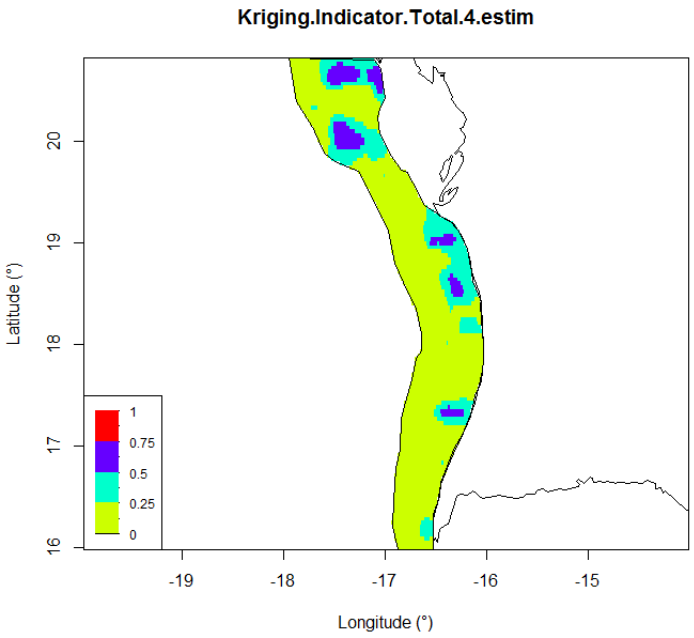


Figure 8.5. Cokriged estimate of the fourth indicator, i.e. estimation of the probability for the acoustic energy to above the 75% quantile. The geometrical sets associated with the largest cutoff are clumped and near the coast. The westward excursions of the habitat in the northern part of the distribution correspond to an area of dense fish distribution.

8.4 Topcut model

Much less sophisticated than the previous non-linear models, the topcut model provides a valuable substitute to linear kriging in the case of a skewed distribution with a few high values.

Very often, the histogram of a concentration (e.g. fish density) is skewed, and in some cases, there are a small number of high values that can make statistics not robust. Structural analysis and mapping of the variable Z may be improved using the topcut model at a threshold z (Rivoirard *et al.*, 2013):

$$Z = \min(Z, z) + [m(z) - z]\, 1_{Z(x)>z} + R$$

In this formula, $m(z)$ is the mean of data values above threshold z. Compared to the sole use of min(Z,z), which would result in a reduction of Z, the addition of the indicator term avoids a bias and allows distributing the average excess of concentrations above z where it is the most probable. The threshold z is chosen so that the residual R, with mean 0 by construction, takes into account the high nugget variability of Z. The estimation of $Z(x)$ depends on the estimation of min(Z,z) and $1_{Z(x)>z}$ (which is robust) and on the estimation of the residual, taken as 0, for example (then data values higher than z are only represented by their mean $m(z)$ and are not present individually in the estimation).

Application 8.3. Mapping anchovy with a topcut model

Acoustic survey data of schooling pelagic stocks often show high concentration values, which seem to appear randomly on a background of medium values. Here, we map anchovy in the Bay of Biscay using a topcut model and compare the results with ordinary kriging. The full demonstration Rscript is in Annex 3 and the data are presented in Annex 2.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.anchovy.bob.2d.db.data","db.data")
rg.load("Demo.anchovy.bob.2d.poly.data","poly.data")

# Data display (left figure)
y1lim <- 43.3; y2lim <- 47; x1lim <- -4.5; x2lim <- -1
plot(db.data,name="ENGR.ENC",pch=1,asp=1.2,inches=5,col="black",
      xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),ti-
tle="",flag.proj=FALSE)
plot(poly.data, add=T, lty=1, density=0)
map("worldHires",add=T,fill=T,col=8)
```

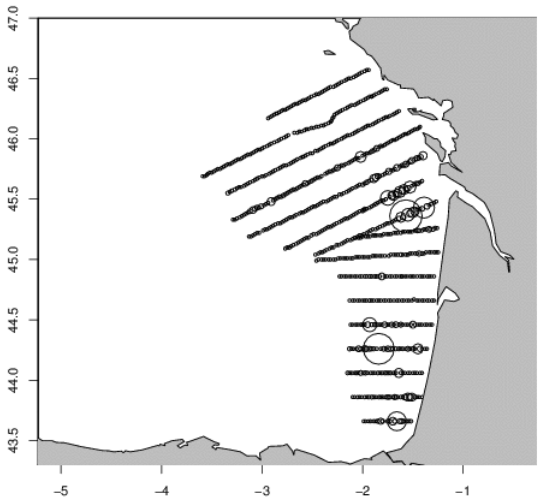


Figure 8.6. Proportional representation of anchovy concentrations ("ENGR.ENC" tonnes nautical mile^{-2}) in the Bay of Biscay. A few high concentration values appear on top of low to medium values.

After the variographic analysis of a range indicator, the topcut threshold chosen was 150 tonnes nautical mile^{-2}. New variables are constructed, which are added in the database; indicator I of values above 150, truncated variable $z1$, mean excess $z2$ above truncation, and residual $z3$ around the mean excess. Note that the mean excess is close to an indicator variable; it equals zero where the data are below the cutoff and a constant (mean above cutoff) where they are above.

```
# Topcut threshold
zi <- 150

# Mean above threshold
mi <- mean(db.data[db.data[,"ENGR.ENC"]>=zi,"ENGR.ENC"])
```

```
# Topcut indicator (I)
db.data <- db.add(db.data,I=(ENGR.ENC >= zi) * 1)

# Truncated variable (z1)
db.data <- db.add(db.data,z1=ifelse(ENGR.ENC>=zi,zi,ENGR.ENC),
                  type.locate=FALSE)

# Mean excess (z2)

db.data <- db.add(db.data,z2=(mi-zi)*I,type.locate=FALSE)

# Residual around mean excess (z3)

db.data <- db.add(db.data,z3=(ENGR.ENC-mi)*I,type.locate=FALSE)
```

The variograms and cross-variograms of $z1$, $z2$, and $z3$ are computed using the function vario.calc(). This shows that $z3$ has no structure (pure nugget effect) and no spatial cross-correlation with $z1$ or $z2$. The analysis thus proceeds with $z1$ and $z2$ only.

```
# Inactivate variable z3
db.data <- db.locate(db.data,names="z3",loctype=NA)
```

The variograms and cross-variogram of $z1$ and $z2$ are estimated using function vario.calc() and modelled using functions model.create() and vario.fit(). Note that as there are multiple locators ("$z1$" and "$z2$") in db.data, these functions perform calculations in the multivariate case automatically. The truncated variable ($z1$) has two nested structures with ranges 8 and 25 nautical miles. The mean excess ($z2$) shows the smallest structure, which is common to both variables and thus present on the cross-variogram.

```
# Projection
projec.define(projection="mean")
# Look for duplicates (points too close)
db.data <- duplicate(db.data)
# Omni-directional variogram
vg <- vario.calc(db.data,lag=2,dirvect=NA, nlag=40)
plot(vg,npairpt=0,npairdw=F,title="",inches=.05)

# Fit variogram model: nugget + 2 spherical models
vg.init <- model.create(vartype="Nugget Effect",ndim=2,nvar=2)
vg.init <- model.create(vartype="Spherical",range=8,model=vg.init)
vg.init <- model.create(vartype="Spherical",range=25,model=vg.init)
# Autimatic fit of sills
vg.mod  <- model.fit(vg, vg.init, niter=100, wmode=3, draw=F)
# Overlay models and variograms
plot(vg,npairdw=F,xlab="Distance (km)",ylab="Variogram")
plot(vg.mod,vario=vg,lwd=2,add=T)
```

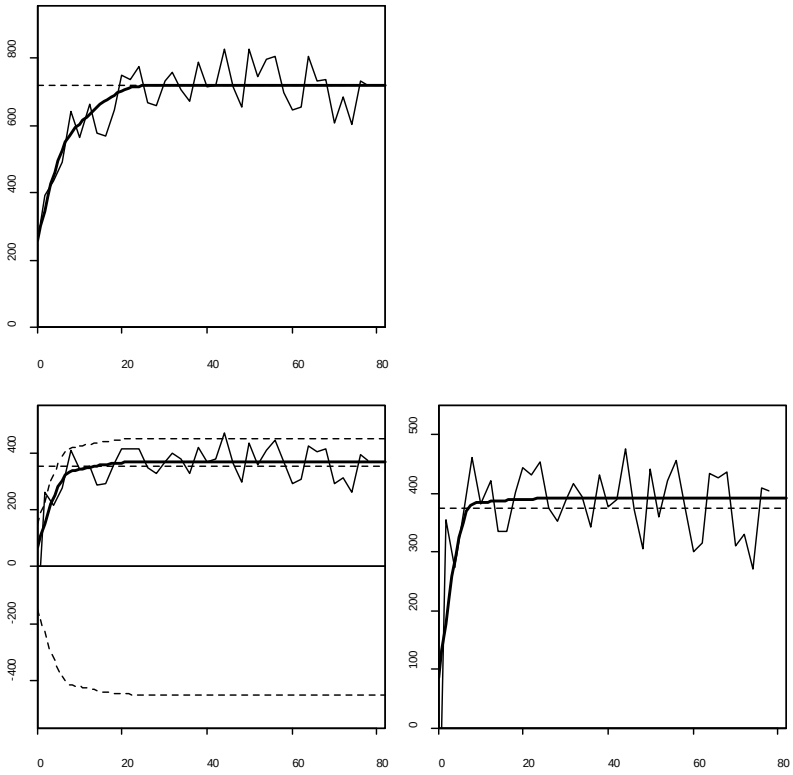


Figure 8.7. Multivariate variogram structure of the topcut model for anchovy in the Bay of Biscay. Top: variogram of the truncated variable. Bottom right: variogram of the mean excess variable. Bottom left: cross-variogram between mean excess and truncated variables.

Kriging with the topcut model now amounts to cokriging variables *z*1 and *z*2 and adding the cokriged values. For that, we use function krige(). Note that as the variogram model is multivariate and as there are multiple "*z*" locators in the data.db, function krige() performs cokriging automatically. Prior to that, a grid and a neighborhood are defined.

```
# Define the Estimation Grid
x0 <- -4; y0 <- 43.4; dx <- 0.1;dy <- 0.1; nx <- 30; ny <- 37
db.grid <- db.create(flag.grid=T,x0=c(x0,y0),dx=c(dx,dy),nx=c(nx,ny))

# Select grid points inside polygon
db.grid <- db.polygon(db.grid,poly.data)

# Define a Moving Neighbourhood
neimov <- neigh.create (ndim=2,type=2,nmini=3,nmaxi=10,radius=25)

# Co-kriging (point)
kres2 <- kriging(dbin=db.data,dbout=db.grid,model=vg.mod,
neigh=neimov,
                 radix="K")

# Add co-kriged z1 and z2 estimates
kres2 <- db.add(kres2,K.topcut.estim=K.z1.estim+K.z2.estim)
```

The map obtained by using the topcut model is compared to that obtained by ordinary kriging on the same grid and with the same neighbourhood. The Rscript for mapping the anchovy by ordinary kriging is provided in Annex 3. The topcut model constrains

the estimation of the high values in areas where the probability is high for these to occur because of cokriging. In contrast, in ordinary kriging, where this constraint is not considered, the rich data values influence the estimate around them. The topcut model result is a lower local estimate in southern areas, where the probability is lower for high values to occur.

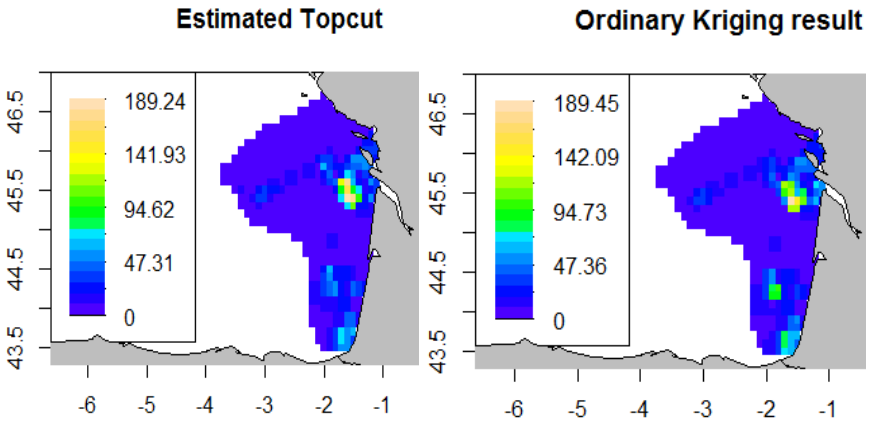


Figure 8.8. Mapping the anchovy in the Bay of Biscay by kriging using a (non-linear) topcut model (left) and ordinary kriging (right).

9 Geostatistical simulations

9.1 General principles

The aim of geostatistical simulations is to reproduce the spatial variability of the regionalized variable. To do so, the variable is represented by an appropriate random function model. Geostatistical simulations are simulations of the random function model, reproducing the variability expected from the model, notably in terms of histogram and variogram. It is possible to build many realizations from the same model; each realization will be different, but will have common features so that they "look" the same (Figure 9.1).

Geostatistical simulations are thus helpful when estimating the uncertainty associated with the combination of different sources of variability. In fishery science, such complex situations occur for acoustic surveys, where different data (acoustic backscatter, fish length, and fish age) must be combined to estimate fish abundance and its associated full uncertainty (Woillez *et al.*, 2009b; 2016). In the following sections, we will see how to simulate herring mean length and acoustic backscatter. These two variables are of different kinds as one shows many zero values (acoustic backscatter) and the other does not. Geostatistical simulations are also particularly helpful in characterizing the uncertainty for decision and risk analysis. They can be used to derive probability maps of exceeding a certain threshold for a variable of interest that could be a pollutant or an anthropic pressure.

Application 9.1. Performing a non-conditional simulation

The following R code performs a non-conditional simulation from a spherical model with range of 0.15 and a sill of 1, visualizing two realizations. To do so, a grid over which values are simulated and a model need to be defined first.

```
# Generate 2 realizations of a non conditional simulation
projec.toggle(0)
data.db <- db.create(data.frame(x1=c(0,0,1,1),x2=c(0,1,1,0)))
grid.db <- db.grid.init(data.db,nodes=c(100,100))
mod <- model.create("Spherical",range=0.15,sill=1)
sim <- simtub(model=mod,dbout=grid.db,nbsim=2,nbtuba=1000)

# Generate figure
plot(sim,name="Simu.V1.S1",pos.legend=1,zlim=c(-4,4))
plot(sim,name="Simu.V1.S2",pos.legend=1,zlim=c(-4,4))
```

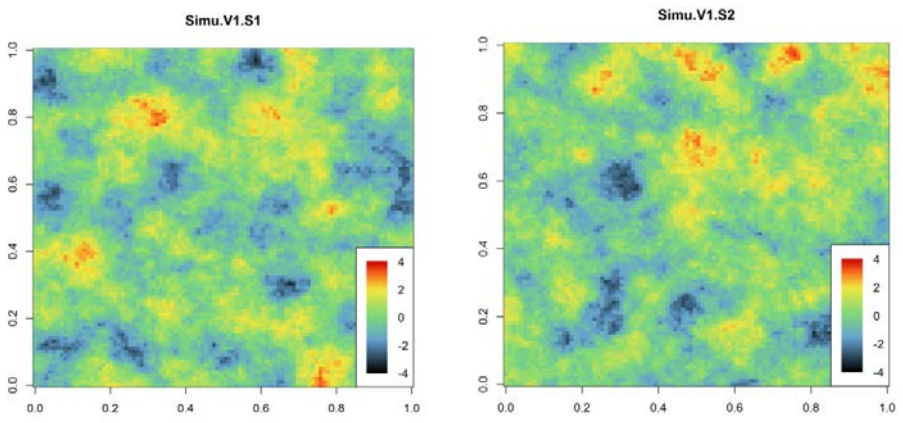


Figure 9.1. Two realizations of a non-conditional simulation with a spherical model of range 0.15 and sill 1. Both maps are different (differences in the location of highs and lows), but they "look" the same as the underlying model is the same.

Geostatistical simulations can be either non-conditional or conditional. Both are simulations of the model. However, non-conditional simulations will ignore the datapoints, while conditional simulations honor the data values at the datapoints; they go through the data. Hence, the highs and lows that can be identified from data will be honoured by conditional simulations. Non-conditional simulations can be helpful to build synthetic examples on which different sampling or exploitation scenarios, for example, can be tested. As it will be seen, non-conditional simulations will also be useful in building conditional simulations. On the other hand, repeated conditional simulations give access, in reality, to the spatial uncertainty of any quantity depending on the simulated variable, e.g. the confidence interval on total abundance in the case of a fish density.

Building simulations requires a full random function model, that is, a model that gives access to the multivariate distribution of the values over any set of points. The Gaussian random function model to be seen now is particularly adapted to simulations. Very often, however, the variable under study cannot be directly modeled by a Gaussian random function, and a transformation (the Gaussian anamorphosis) is necessary to make the link between the variable and its associated Gaussian transformed, as will be seen later.

9.2 Gaussian random functions

The Gaussian model is particularly appropriate for simulations. First, because of the central limit theorem, normality results from the addition of many independent variables, as done when constructing non-conditional simulations (a classical way to make these is the turning bands method, see next section). Secondly, because of its properties, the Gaussian model is easy to condition at datapoints.

The bell-shaped probability density function of a Gaussian random variable t is:

$$g(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$$

This is well known, but a Gaussian random function model is more general.

Theoretically a random function $Y(x)$ is Gaussian if the distribution of any vector $Y(x_1)$, $Y(x_2), \ldots, Y(x_N)$ is multivariate Gaussian (i.e. every linear combination is Gaussian), having the famous bell-shaped probability density function. In the stationary case, this means that the histogram (marginal distribution) of $Y(x)$ is bell-shaped, but also pairs, triplets, and so on are Gaussian. In particular, the bivariate distributions of pairs [scatterplots between $Y(x)$ and $Y(x+h)$ at any distance h] have an elliptical shape corresponding to the bivariate probability distribution function [here, for standard Gaussian with correlation coefficient $= \rho(h)$]:

$$g_\rho(t,u) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{t^2-2\rho tu+u^2}{2(1-\rho^2)}}$$

The full distribution is entirely determined by the mean, the variance, and the covariance function. For a standard Gaussian (mean 0, variance 1) as considered further, the distribution is determined by the covariance $C(h)$ or correlogram $\rho(h)$.

9.3 Non conditional simulation with the turning bands method

The turning bands method is a rapid way to build a non-conditional simulation of a random function model with a desired covariance (Lantuéjoul, 2002; Chilès and Delfiner, 2012). It allows for the simulation of a random function in R^2 or R^3 from independent simulations in R, performed along lines having random directions. For each such line, a 1-D simulation is performed and then expanded to the whole space giving "bands" informing every point in space (Figure 9.2). The final simulation at every point in space is obtained as an average of the values coming from the 1-D simulations in all random directions. Because of the mixing of many independent 1-D simulations, this gives Gaussian random functions.

The 1-D simulations to be made do not obey the same covariance as the desired final covariance, but there is a relationship linking these. For example, to reproduce the covariance $C_3(h)$ in R^3, the 1-D covariance $C_1(h)$ is obtained by:

$$C_1(h) = \frac{d}{dh}[hC_3(h)]dh$$

Ad hoc processes are designed to simulate the 1-D covariances corresponding to the usual desired covariances.

Application 9.2. Non-conditional simulation by turning bands

The following R code performs a non-conditional simulation of an exponential model (range = 0.15 and sill = 1) using the turning bands method. Four realizations are produced with a number of bands corresponding to 1, 10, 100, and 1000.

```
# Generate 4 realizations of non conditional simulation
# with a varying number of bands
projec.toggle(0)
data.db <- db.create(data.frame(x1=c(0,0,1,1),x2=c(0,1,1,0)))
grid.db <- db.grid.init(data.db,nodes=c(100,100))
mod <- model.create("Exponential",range=.15,sill=1)
sim1<- simtub(model=mod,dbout=grid.db,nbsim=1,nbtuba=1)
sim2<- simtub(model=mod,dbout=grid.db,nbsim=1,nbtuba=10)
sim3<- simtub(model=mod,dbout=grid.db,nbsim=1,nbtuba=100)
sim4<- simtub(model=mod,dbout=grid.db,nbsim=1,nbtuba=1000)

# Generate figures
plot(sim1,name="Simu.V1.S1",pos.legend=1,zlim=c(-4,4))
plot(sim2,name="Simu.V1.S1",pos.legend=1,zlim=c(-4,4))
plot(sim3,name="Simu.V1.S1",pos.legend=1,zlim=c(-4,4))
plot(sim4,name="Simu.V1.S1",pos.legend=1,zlim=c(-4,4))
```

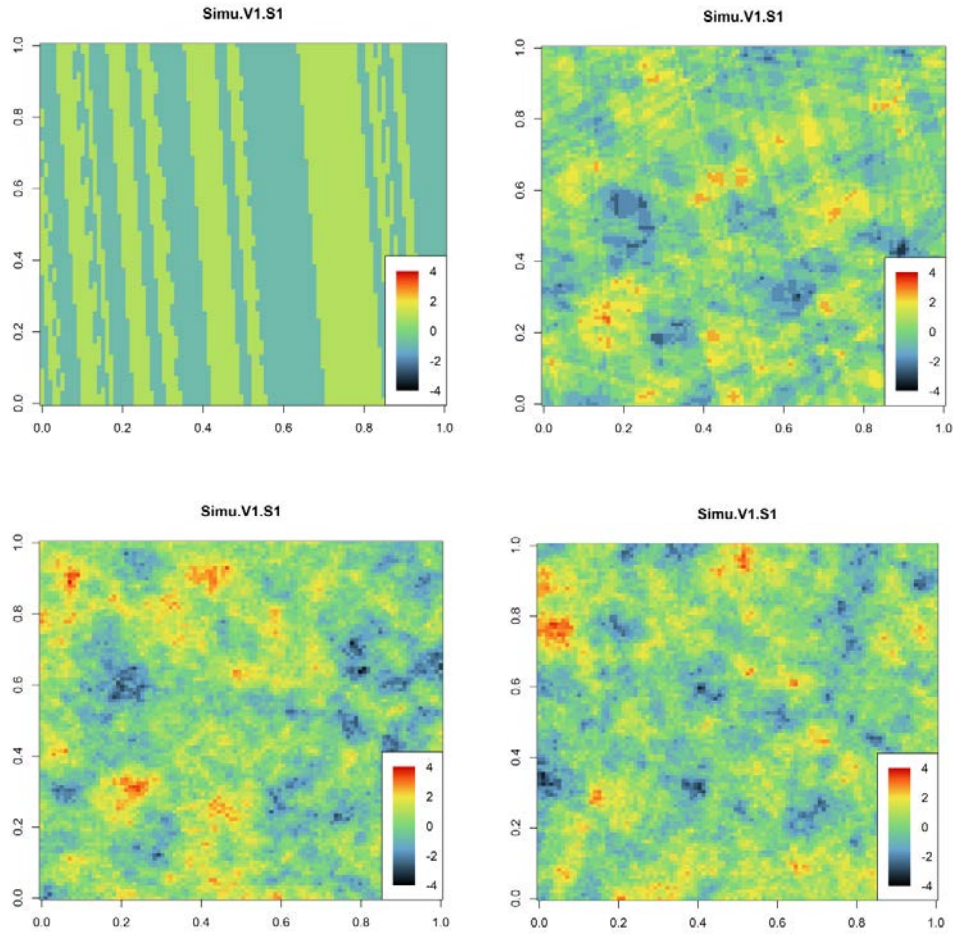


Figure 9.2. Non conditional simulation of an exponential model (range = 0.15 and sill = 1) using turning bands (1, 10, 100, 1000 bands). See how the number of bands impacts the simulated field. At least, 1000 bands are needed to produce a simulated field not impacted by the number of bands.

9.4 Conditioning to the data

Performing conditional simulations is, in general, very difficult, but it is much easier in the Gaussian model. Although a bit technical, this section explains how conditioning to data is treated in this model.

The Gaussian model has useful properties:

- Conditional distributions (e.g. distribution at a target point conditional on data) are still Gaussian.
- The mean of a conditional distribution (the "conditional expectation") is linear (kriging)
- The variance of a conditional distribution is equal to kriging variance and does not depend on the conditioning values (no heterosedasticity).
- No correlation is equivalent to independence.

At a given target point, the distribution of $Y(x)$ conditional on the data is Gaussian, with mean equal to its kriging value and with variance equal to the kriging variance. It can be written as $Y^K(x) + \sigma_K R$ where R is a standard Gaussian residual. This allows

simulating $Y(x)$ at a point by simulating R (but this is not sufficient to simulate Y at all target points).

In non-linear geostatistics, such conditional distributions can be used to estimate any function f of $Y(x)$ at target points:

$$E[f(Y(x))|data] = \int f(Y(x)^K + \sigma_K t)g(t)dt$$

(note that this is not simply the function of the estimate $f(Y(x)^K)$, which would be biased).

In particular, the conditional probability of exceeding some threshold y is:

$$P(Y(x) \geq y|data) = E\left[1_{Y(x)\geq y}|data\right] = 1 - G\left(\frac{y - Y(x)^K}{\sigma_K}\right)$$

where $G(y) = P(Y < y) = \int^y g(t)dt$ is the cumulative density function of a standard Gaussian.

In simulations, the properties of the Gaussian model make conditioning easy, as explained now (Lantuéjoul, 2002; Chilès and Delfiner, 2012).

The difference between the kriged map and the unknown reality is the error map $\varepsilon^Y(x)$:

$$Y(x) = Y^K(x) + \varepsilon^Y(x)$$

In the Gaussian case, this error map is stochastically independent from the kriged map. Then, the idea is to substitute the actual error map by an independently simulated one. First, a non-conditional simulation is performed at all target points and datapoints, then kriged at all target points from datapoints, so that the simulated error is available at all target points:

$$Y_{NCS}(x) = Y^K_{NCS}(x) + \varepsilon^Y_{NCS}(x)$$

Then, this simulated error map is added to the original kriging map, giving the conditional simulation:

$$Y_{CS}(x) = Y^K(x) + \varepsilon^Y_{NCS}(x)$$

Indeed, at datapoints, the kriged value equals the actual value and the error is zero, so that the finally simulated value coincides with the actual value.

In summary, a conditional simulation of a Gaussian random function can be obtained, requiring only a non-conditional simulation and a kriging process (Figure 9.3):

$$Y_{CS}(x) = (Y(x) - Y_{NCS}(x))^K + Y_{NCS}(x)$$

Application 9.3. Principle of a conditional simulation

The following R code performs a conditional simulation from a sampled, simulated field with a nested model with two components; the first component is a nugget with a sill of 0.01, and the second component is spherical with a range of 0.25 and a sill of 0.99. Four figures are produced. The first figure is a realization of a non-conditional simulation, which represents the [unknown] reality. The second figure represents samples taken from this simulated field. The third figure is the kriged map of these samples, which is the best linear unbiased interpolation, and is smoother than the reality. The fourth figure is a realization of a conditional simulation, which honours the values at the sample locations and reproduces the spatial variability of the regionalized variable.

```
# Create the simulation grid
projec.toggle(0)
grid.db <- db.grid.init(db.create(data.frame(x1=c(0,0,1,1),x2=c(0,1,1
,0))),
                        nodes=c(101,101))

# Create a variogram model
a1 <- model.create(vartype=1,sill=0.01)
mod <- model.create(vartype=3,sill=0.99,range=0.25,model=a1)

# Generate the truth
real <- simtub(model=mod,dbout=grid.db,nbsim=1,nbtuba=1000)

# Sample the true field at random
data.df <- data.frame(x1=round(runif(100,0,1),2),x2=round(runif(100,0
,1),2))
data.df <- merge(data.df,real[,2:4],by=c("x1","x2"));names(data.df)[3
]<-"z1"
data.db <- db.create(data.df)

# Perform an ordinary kriging in unique neighbourhood
kri <- kriging(dbin=data.db,dbout=grid.db,model=mod,
               neigh=neigh.create(type=0,ndim=2),uc=NA,mean=0)

# Perform a conditional simulation
sc <- simtub(dbin=data.db,dbout=grid.db,model=mod,
             neigh= neigh.create(type=0,ndim=2),uc=NA,mean=0,
             nbsim=1,nbtuba=1000,seed=232132)

# Generate figures
plot(real,name="Simu.V1.S1",title="Reality",pos.legend=1,zlim=c(-4,4)
)
plot(data.db,title="Sample data")
plot(kri,name="Kriging.z1.estim",title="Kriging",pos.legend=1,zlim=c(
-4,4))
plot(sc,name="Simu.z1.S1",title="Conditional simulation",
     pos.legend=1,zlim=c(-4,4))
```

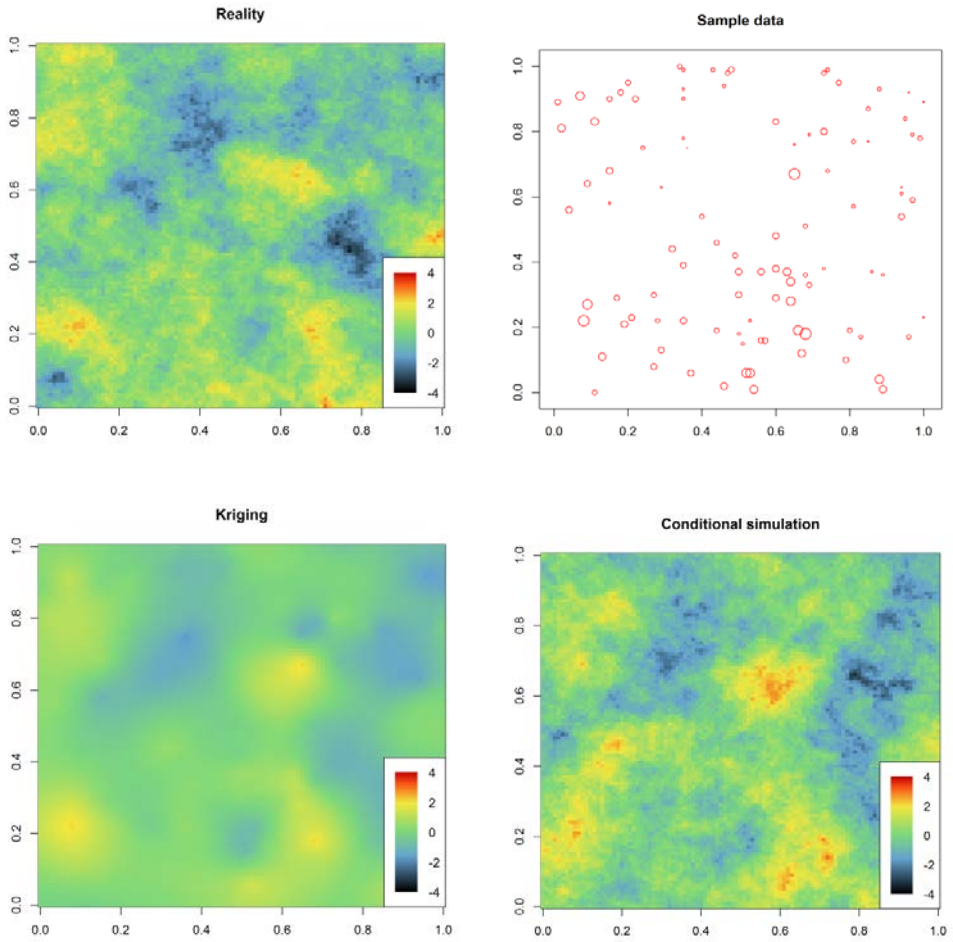


Figure 9.3. Principle of a conditional simulation: non-conditional simulation (top left), data (top right), kriging the data (bottom left), conditional simulation (bottom right). Note how kriging produces a smooth field, while conditional simulation produces a heterogeneous field that "looks" like the reality.

9.5 Gaussian anamorphosis

In many cases, the variable under study cannot be modeled directly with a Gaussian random function. Typical of concentrations like a fish density, it may be stationary, but has a skewed histogram (the case of numerous zeroes will be treated in the next section). Then, a transformation into normal scores must be applied before using a Gaussian random function model. The different steps for simulations are:

- transform the variable into normal scores; data values give Gaussian data values with bell shaped histogram;
- preferably check that the Gaussian random function model is admissible [e.g. the scatterplots $(Y(x), Y(x+h))$ must present an elliptical shape];
- infer the Gaussian random function model (e.g. its covariance);
- perform the simulations of the Gaussian transformed in the Gaussian field;
- transform these simulations back to the original scale.

To go into more details, the transformation between the original variable represented by $Z(x)$ and the Gaussian variable $Y(x)$ is called the Gaussian anamorphosis. It is a

non-decreasing function denoted Φ with $Z = \Phi(Y)$ (not the reverse – we will see why later), so that the variable is seen as deriving from a Gaussian field. When going from Z to Y (supposedly standard), the histogram is reshaped into the bell-shaped Gaussian histogram. Let F be the cumulative distribution function of Z and G the cumulative density function of the standard Gaussian. Each data value z and its normal score y correspond to the same cumulated probability: $F(z) = G(y)$ so that $z = F^{-1}[G(y)]$ and $\Phi = F^{-1}[G]$.

The anamorphosis function Φ represents the distribution (histogram) of values over the domain and is inferred from data. When data are not regularly spaced, they should be declustered (e.g. each datapoint being weighted by its surface of influence). The knowledge of the distribution may, however, be poor when data are not numerous and is generally poor in the tails of the distribution. An anamorphosis model is then used to fit and smooth the empirical anamorphosis of the data.

One classical way to do this consists of fitting a polynomial function. Rather than using monomials, this makes use of so-called Hermite polynomials that are particularly adapted to the Gaussian context (Rivoirard, 1994; Chilès and Delfiner, 2012). The user just has to choose the number of Hermite polynomials, since this will correspond to the degree of the polynomial expansion of the anamorphosis.

Another way to model the anamorphosis consists of dispersing each data value, i.e. replacing each data value z by a distribution with mean z and some variance. Thus, the overall mean is unchanged, and the overall variance is slightly increased (by a quantity which should represent the global estimation variance). Typically for a skewed distribution such as fish density, each data value z is dispersed by a lognormal distribution, having z as mean and a common logarithmic variance, and so a variance proportional to z^2 (the highest z values of the tail are getting more dispersed). The logarithmic variance is adjusted on the desired overall increase of variance.

Application 9.4. Conditional simulation of herring mean length

The following R lines show an example of conditional simulation without the presence of zeros (full demonstration Rscript in Annex 3, data detail in Annex 2). The data used here correspond to herring mean length. They were collected during trawl stations that were performed to assist the scrutinization of the acoustic backscatter.

First, the data are loaded and a projection is set. Then, a RGeostats model of anamorphosis is defined using the function anam.fit(). The "Hermitian anamorphosis" model is used when the type is set to "gaus". It requires defining the number of Hermite polynomials to be used, here 10. The herring mean length data are then transformed into normal scores using the function anam.z2y() (Figure 9.4).

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herring.len.scot.db.data","db.data")
rg.load("Demo.herring.len.scot.poly.data","poly.data")
rg.load("Demo.herring.len.scot.grid.simu","grid.simu")
projec.define(db=db.data)

# Histogram of the Mean Length variable (left figure)
hist(db.data[,"m.length"],breaks=20,col="grey",main="",xlab="m.length
")

# Define the anamorphosis model (right figure)
```

```
model.anam <- anam.fit(db.data,type="gaus",nbpoly=10,draw=T)

# Transform the data into Gaussian
db.data <- anam.z2y(db.data,anam=model.anam)
```

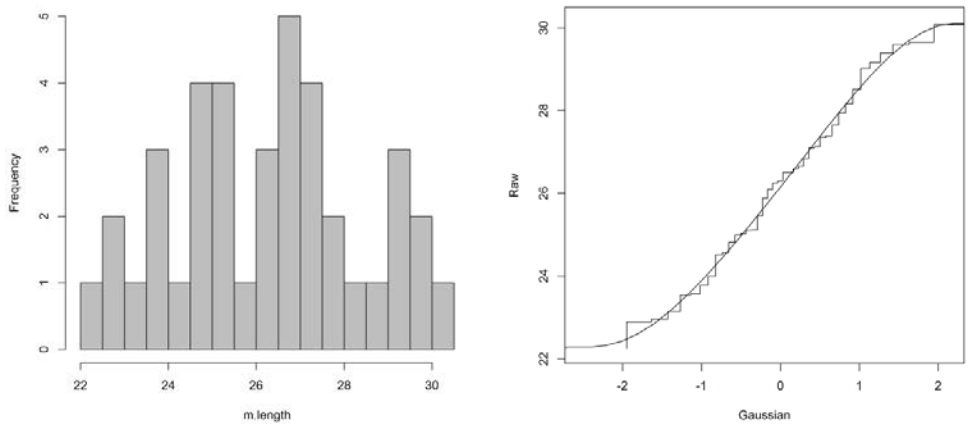


Figure 9.4. Left: histogram of Scottish herring mean length data at stations. Right: Gaussian anamorphosis (the empirical staircase anamorphosis is modeled by a polynomial function).

Before performing the simulations in the Gaussian field, the variogram model is fitted on the Gaussian transformed herring mean length. Then, the conditional simulations are performed using the function simtub(). The model of anamorphosis was stored and is used to convert conditional simulations in the Gaussian space back to the original space using the function anam.y2z() (Figure 9.4).

```
# Transform the data into Gaussian
db.data <- anam.z2y(db.data,anam=model.anam)

# Build the model
vario.data <- vario.calc(db.data)
model.vario <- model.auto(vario=vario.data)

# Cond. simulation of gaussian variable
grid.simu <- simtub(dbin=db.data, dbout=grid.simu, model=model.vario,
                    neigh=neigh.create(type=0,ndim=2), uc = "", mean
= 0,                                seed = 29091978,
                    nbsimu = 1, nbtuba = 1000, radix = "Simu",
                    modify.target = TRUE)

# Transform gaussian cond simulation into raw conditional simulation
grid.simu <- anam.y2z(grid.simu,name="Simu.Gaussian.m.length.S1",anam
=model.anam)

# Display
plot(grid.simu,name="Simu.Gaussian.m.length.S1",
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180),
     title="Conditional Simulation of Gaussian Variable",
     pos.legend=5,flag.proj=F)
plot(poly.data,col=0,flag.proj=F,add=T)
map("worldHires",add=T,fill=T,col=8)

plot(grid.simu,name="Raw.Simu.Gaussian.m.length.S1",
```

```
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180),
     title="Conditional Simulation of Raw Variable",
     pos.legend=5,flag.proj=F)
plot(poly.data,col=0,flag.proj=F,add=T)
map("worldHires",add=T,fill=T,col=8);box()
```

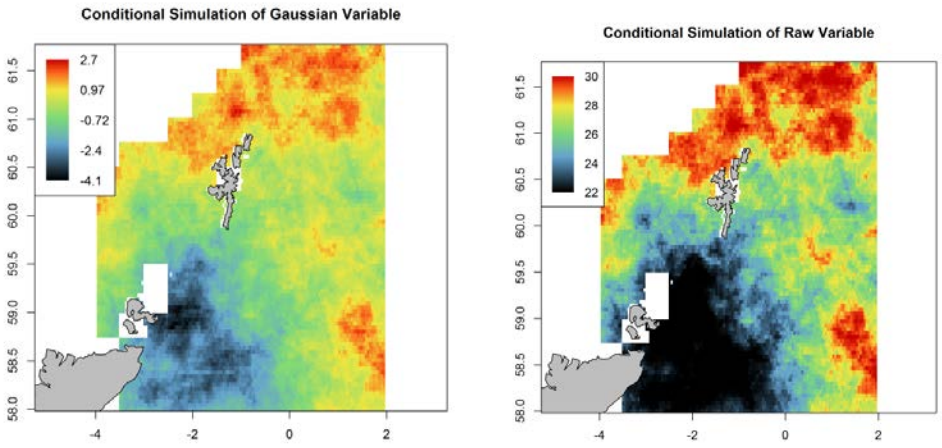


Figure 9.5. Scottish herring mean length: conditional simulation in the Gaussian space (left), back-transformed into mean length arithmetic scale (right).

9.6 Case of zero effects

In the above simulation method, simulations are performed in the Gaussian space. Simulations are conditional on the Gaussian data values. This supposes that original data values have been inverted into Gaussian values. This is generally not a problem when the distribution of Z is continuous, but what if a non-negative variable Z has a continuous distribution except for the presence of 50% of zeroes? There is no problem with inverting the positive values of Z, but what about the zeroes? In this case, the anamorphosis function Φ is identically equal to 0 for all the 50% of the negative values of the Gaussian variable. If the proportion of zeros is p_0, Φ will be 0 for all Gaussian values less than the Gaussian threshold y corresponding to the cumulated probability $p_0 = G(y)$. Z is supposed to derive from a Gaussian field Y by $Z = \Phi(Y)$, but the inverse of Φ does not exist (this is the reason why Φ goes from Y to Z, not the reverse), and we do not know which value of Y corresponds to a 0 value for Z. In addition, since the values of Y are unknown where Z is 0, the variogram or covariance of Y is not directly accessible.

In such a case, two preliminary steps must be performed. The first consists of determining the covariance of Y. In the Gaussian random function model, pairs $[(Y(x), Y(x+h)])$ for a given distance h have a bivariate Gaussian density, depending only on the correlation $\rho = \rho(h)$ at this distance. It follows that the covariance of any function of Y, say $f(Y)$, can be written as:

$$C_f(h) = Cov\big[f\big(Y(x)\big), f(Y(x+h))\big] = E\big[f\big(Y(x)\big)f(Y(x+h))\big] - (E[f(Y)])^2$$
$$= \int f(t)f(u)g_{\rho(h)}(t,u)dt\,du - (E[f(Y)])^2$$

Hence, the covariance of $f(Y)$ at distance h depends on the covariance of Y at this distance (it can be shown to increase when ρ increases from 0 to 1).

In practice, the function $f(Y)$ is to be chosen to be closely linked to Y while determined at all datapoints, for example Y^+ equal to Y when $Z > 0$ and to the threshold y when $Z = 0$ (Y unknown, but smaller than y).

Then, the covariance model $\rho(h)$ of Y can be chosen so that the above transformation fits the observed structure of $f(Y)$. This presents the disadvantage to be indirect. Another solution is to invert each value from the observed $C_f(h)$ according to the above formula in order to exhibit a pseudo-experimental structure $\rho(h)$ for Y to be modeled [some limitations to the inversion may, however, appear with negative values of $\rho(h)$].

Once the structure of Y is determined, the second additional step consists of simulating the values of Y at the datapoints where Z is zero (at the datapoints where Z is positive, Y is directly determined by inversion). This must be conditional on their being less than the threshold y, and conditional on the values of Y where Z is positive, while honouring the Gaussian structure. This is classically done by a Gibbs sampler, iteratively over the $Z = 0$ datapoints as follows:

- Initialize the $Y(x) < y$ at the $Z = 0$ datapoints;
- Iteratively at each such point simulate $Y(x) < y$, conditional on Y values at all other datapoints (the conditional distribution at this point is Gaussian, and we have seen earlier how to simulate it with a residual);
- Repeat until convergence of the variogram of Y at datapoints.

At the end, Y is known or simulated at all datapoints and can be used for the classical conditional simulation seen above.

Application 9.5. Conditional simulation of herring acoustic backscatter in the presence of zeros

The following R lines (full script in Annex 3) show an example of conditional simulation with the presence of zeros. The data used here correspond to herring acoustic backscattering (Annex 2) collected along survey transects.

First, the data are loaded, a projection is set, and areas of influence are computed to compensate for the irregularity of the sampling (variable intertransect distance). Then, a model of anamorphosis is defined using the function anam.fit(). The chosen RGeostats model of anamorphosis is the "empirical anamorphosis" with a logarithmic dispersion. The herring acoustic backscatter data are then transformed into normal scores using the function anam.z2y() (Figure 9.6). However, because of the many zeros, the transformation is not a bijection function, and all the zeros end up receiving the same Gaussian values *y*cut derived from the proportion of zeros.

```
# Pre-requisite
projec.toggle(0)
rg.load("Demo.herring.sa.scot.db.data","db.data")
rg.load("Demo.herring.sa.scot.poly.data","poly.data")
rg.load("Demo.herring.sa.scot.grid.simu","grid.simu")
projec.define(db=db.data)

# Display the histogram
hist(db.data[,"sa"],breaks=100,col="grey",main="",xlab="sa")

# Define the anamorphosis model (Right figure)
model.anam <- anam.fit(db.data,type="emp",ndisc=db.data$nech,
                       sigma2e=800,draw=TRUE,title="")
```

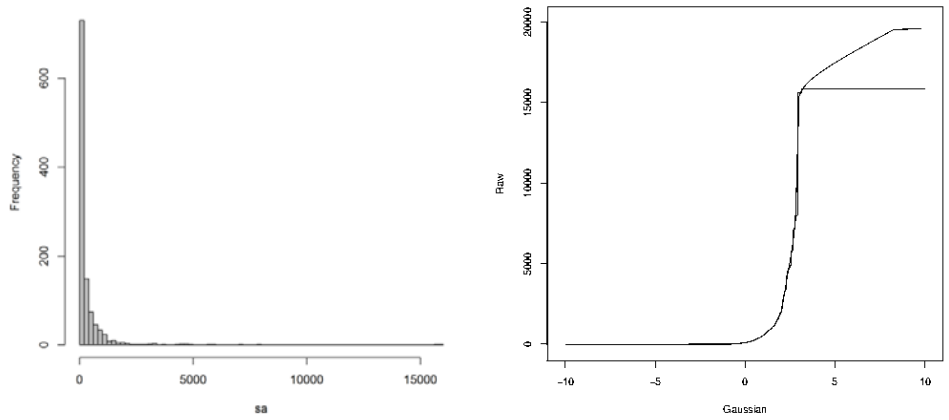


Figure 9.6. Acoustic backscatter attributed to Scottish herring. Left: histogram showing a spike of zero values. Right: model of anamorphosis.

```
# Transform the data into Gaussian
db.data <- anam.z2y(db.data,anam=model.anam)
print(db.data,flag.stats=TRUE,names="Gaussian.sa")
db.data <- db.rename(db.data,name="Gaussian.sa",newname="Yp")
ycut <- round(qnorm(sum(db.extract(db.data,"sa") == 0) / db.data$nech
),5)
Y <- db.extract(db.data,"Yp")
Y[Y < ycut] <- ycut
db.data <- db.replace(db.data,"Yp",Y)
print(db.data,flag.stats=TRUE,names="Yp")
```

Then, we look for the variogram model of the Gaussian. It is modeled from the variogram of the truncated Gaussian Y^+. To do so, the variogram of the truncated Gaussian Y^+ is transformed using the function vario.trans.cut() and the function model.auto() is applied. vario.trans.cut() allows for the inversion of each value from the observed variogram of the truncated Gaussian Y^+ in order to exhibit a pseudo-experimental variogram of the Gaussian Y to be modeled (see formula above). The variogram model of the Gaussian Y exhibits 3 components: a nugget, an exponential and a spherical model (Figure 9.7).

```
# Modeling Gaussian variable Y
n.H <- 50
vario.Yp <- vario.calc(db.data,lag=2.5,nlag=50)
vario.Y  <- vario.trans.cut(vario.Yp,ycut,n.H)
model.vario.Y <- model.auto(vario.Y,struc=melem.name(c(1,2,3)),draw=F
)
plot(vario.Yp,npairdw=T,inches=0.05,col="black",ylim=c(0,1.2))
plot(vario.Y,npairdw=T,inches=0.05,col="red",add=TRUE)
plot(model.vario.Y,add=T,col="red")
legend(x="bottomright",legend=c("Variogram of Yp","Variogram of Y"),
       lty=c(1,1),col=c("black","red"))
```

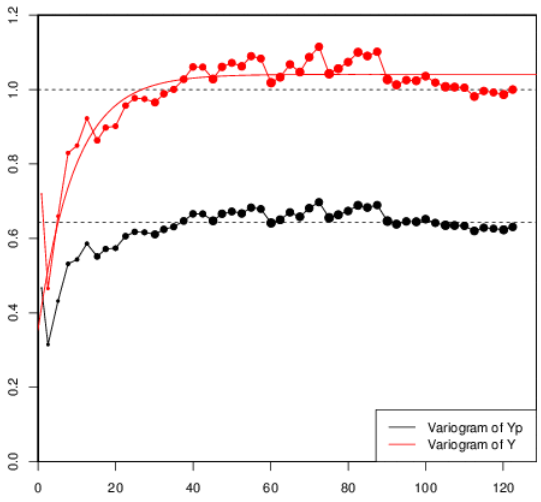


Figure 9.7. Modeling the variogram of the Gaussian *Y* from the variogram of the truncated Gaussian Y^+.

Then, the values of Y at the datapoints where Z is zero are simulated using a Gibbs sampling. The simulation intervals and locator are defined. Then, the Gibbs sampling is performed with the function gibbs(). Figure 9.8 illustrates the histogram before and after the gibbs sampling and compares the experimental variogram and the model of Y after the Gibbs sampling.

```
# Define interval limits for the Gibbs
Ymax <- db.extract(db.data,name="Yp",flag.compress=FALSE)
Ymin <- db.extract(db.data,name="Yp",flag.compress=FALSE)
Ymin[Ymin <= ycut] <- -10
db.data<-db.add(db.data,Ymax)
db.data<-db.locate(db.data,db.data$natt,"upper")
db.data<-db.add(db.data,Ymin)
db.data<-db.locate(db.data,db.data$natt,"lower")

# A Gibbs sampler
db.data <-gibbs(db = db.data, model = model.vario.Y, seed = 232132,
                nboot = 10, niter = 100, flag.norm=FALSE, percent=0,
toleps = 1,
                radix = "Gibbs", modify.target = TRUE)
db.data<-db.rename(db.data,"Gibbs.G1","Y")
print(db.data,flag.stats=TRUE,names="Y")

# Histograms
hist(db.data[,"Yp"],breaks=100,xlim=c(-4,4),ylim=c(0,300),main="",xla
b="Yp")
hist(db.data[,"Y"],breaks=100,xlim=c(-4,4),ylim=c(0,300),main="",xlab
="Y")
vario.Yg <- vario.calc(db.data,lag=2.5,nlag=50)
plot(vario.Yg,npairdw=TRUE,inches=0.05)
plot(model.vario.Y,add=TRUE,col="red")
```

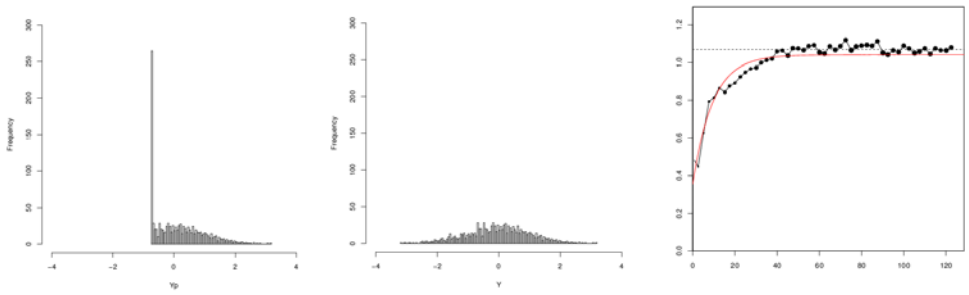


Figure 9.8. Gibbs sampling for a truncated Gaussian variable Y^+. Histogram before (left) and after (middle) the Gibbs sampling. Comparison of the experimental variogram and the model of Y after the Gibbs sampling (right).

The last steps are the conditional simulation on the Gaussian variable using its variogram model, and the back-transformation of the conditional simulation to the original scale (Figure 9.9).

```
# Conditional simulation of the Gaussian variable
grid.simu <- simtub(dbin=db.data, dbout=grid.simu, model=model.vario.
Y,
                    neigh=neigh.create(type=0,ndim=2), uc = "", mean
= 0,                                   seed = 232132,
                    nbsimu = 1, nbtuba = 1000,
                    radix = "Simu", modify.target = TRUE)
grid.simu <- db.rename(grid.simu,"Simu.Y.S1","Simu.Y")
print(grid.simu,flag.stats=TRUE,names="Simu.Y")

# Display
plot(grid.simu,name="Simu.Y",
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180),
     pos.legend=5,flag.proj=F,title="")
map("worldHires",add=T,fill=T,col=8)

# Transform gaussian conditional simulation into raw scale
grid.simu <- anam.y2z(grid.simu,name="Simu.Y",anam=model.anam)

# Display
plot(grid.simu,name="Raw.Simu.Y",
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180),
     pos.legend=5,flag.proj=F,title="",zlim=c(10,18000))
map("worldHires",add=T,fill=T,col=8)
```

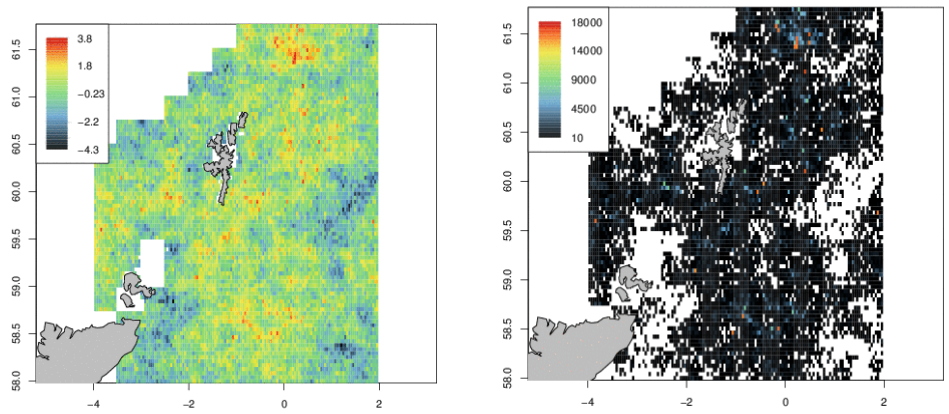


Figure 9.9. Scottish herring acoustic backscatter: conditional simulation in the Gaussian space (left), back-transformed into acoustic backscatter (right).

Note that in acoustic surveys, fish abundance is estimated by combining the acoustic backscatter and the fish length data. Here, conditional simulations offer the unique opportunity to combine uncertainties of each simulated field that we performed in this section, i.e. the simulated mean length and the simulated acoustic backscatter, into a final abundance estimate (see Woillez *et al*., 2009b for more details).

10 Conclusion

In the 1990s, the need to estimate variance for acoustic surveys with regularly spaced transects led to the recognition of geostatistics in fisheries science as a useful framework providing a set of coherent tools to take into account explicitly the autocorrelation in the data (e.g. Rivoirard *et al*., 2000). Now, the need for mapping has broadened with the ecosystem approach. Thus, in addition to variography and kriging, this handbook offers an introduction to a wide range of geostatistical methods, including multivariate, non-linear, and simulation procedures for this wider ecosystem context for which fisheries survey data are now also used. The package RGeostats (Renard *et al*., 2016), freely available for the R environment, will allow the user to apply the many geostatistical tools. The R code provided in this handbook was designed to serve as a teaser for the reader to develop his/her own code for his/her case studies. Spatio-temporal modelling was not considered here (yet, multiyears could be analyzed using a multivariate approach), and this topic is certainly an upcoming challenge for the analysis of multiple surveys.

11 References

Bez, N. 2002. Global fish abundance estimation from regular sampling: the geostatistical transitive method. Canadian Journal of Fisheries and Aquatic Sciences, 59: 1921–1931.

Bez, N., and Braham, C-B. 2014. Indicator variables for a robust estimation of an acoustic index of abundance. Canadian Journal of Fisheries and Aquatic Sciences, 71: 709–718.

Bez, N., and Rivoirard, J. 2000. Indices of collocation between populations. *In* Small Pelagic Fishes and Climate Change Programme: Report of a Workshop on the Use of Continuous Underway Fish Egg Sampler (CUFES) for Mapping Spawning Habitat of Pelagic Fish (9–11 February 2000, San Sebastian, Spain), pp. 48–52. Ed. by D. M. Chekley, J. R., Hunter, L. Motos, and C. D. van der Lingen. GLOBEC Report 14, 1–65.

Bez, N., and Rivoirard, J. 2001. Transitive geostatistics to characterise spatial aggregations with diffuse limits: an application on mackerel ichthyoplankton. Fisheries Research, 50: 41–58.

Chilès, J-P., and Delfiner, P. 2012. Geostatistics: Modeling Spatial Uncertainty, 2nd edn. John Wiley & Sons, New York. 731 pp.

Cochran, W. G. 1977. Sampling Techniques, 3rd edn. John Wiley & Sons, New York. 428 pp.

Doray, M., Massé, J., and Petitgas, P. 2010. Pelagic fish stock assessment by acoustic methods at Ifremer. Internal report Ifremer. http://archimer.ifremer.fr/doc/00003/11446/.

Faraj, A., and Bez, N. 2007. Spatial considerations for the Dakhla stock of *Octopus vulgaris*: indicators, patterns and fisheries interactions. ICES Journal of Marine Science, 64: 1820–1828.

Fernandes, P. G., and Rivoirard, J. 1999. A geostatistical analysis of the spatial distribution and abundance of cod, haddock and whiting in North Scotland. GEOENV II. Quantitative Geology and Geostatistics 10: 201–212.

Foote, K. G., Knudsen, H. P., Vestnes, G., MacLennan, D. N., and Simmonds, E. J. 1987. Calibration of acoustic instruments for fish density estimation: a practical guide. ICES Cooperative Research Report No. 144. 69 pp.

Gini, C. 1921. Measurement of inequality and incomes. The Economic Journal, 31: 124–126.

ICES. 1993. Report of the Workshop on the Applicability of Spatial Statistical Techniques to Acoustic Survey Data. ICES Cooperative Research Report No. 195. 87 pp.

ICES. 2006. Report of the Planning Group on Herring Surveys (PGHERS), 24–27 January 2006, Rostock, Germany. ICES Document CM 2006/LRC: 04. 239 pp.

ICES. 2015. Report of the International Bottom Trawl Survey Working Group (IBTSWG), 23–27 March 2015, Bergen, Norway. ICES Document CM 2015/SSGIEOM: 24. 278 pp.

Journel, A., and Huijbregts, C. 1978. Mining Geostatistics. Academic Press, London. 600 pp.

Knudsen, H. P. 1990. The Bergen Echo Integrator: an introduction. Journal du Conseil International pour l'Exploration de la Mer, 47: 167–174.

Lantuéjoul, C. 2002. Geostatistical Simulation, Models and Algorithms. Springer, Berlin. 256 pp.

MacCall, A. D. 1990. Dynamic Geography of Marine Fish Populations. University of Washington Press, Seattle. 153 pp.

MacLennan, D. N., Fernandes, P. G., and Dalen, J. 2002. A consistent approach to definitions and symbols in fisheries acoustics. ICES Journal of Marine Science, 59: 365–369.

MacLennan, D. N., and Simmonds, E. J. 1992. Fisheries Acoustics. Chapman & Hall, London. 325 pp.

Matheron, G. 1971. The Theory of Regionalized Variables and its Applications. Les Cahiers du Centre de Morphologie Mathématique, Fasc. 5. Ecole Nationale Supérieure des Mines de Paris, Fontainebleau. 212 pp.

Matheron, G. 1989. Estimating and Choosing: An Essay on Probabiliy in Practice. Springer-Verlag, Berlin. 141 pp.

Morfin, M., Fromentin, J-M., Jadaud, A., and Bez, N. 2012. Spatio-temporal patterns of key exploited marine species in the northwestern Mediterranean Sea. PLoS ONE 7(5): e37907.

Petitgas, P. 1993a. Geostatistics for fish stock assessments: a review and an acoustic application. ICES Journal of Marine Science, 50: 285–298.

Petitgas, P. 1993b. Use of disjunctive kriging to model areas of high pelagic fish density in acoustic fisheries surveys. Aquatic Living Resources, 6: 201–209.

Petitgas, P. 1998. Biomass-dependent dynamics of fish spatial distributions characterized by geostatistical aggregation curves. ICES Journal of Marine Science, 55: 443–453.

Petitgas, P. 2001. Geostatistics in fisheries survey design and stock assessment: models, variances and applications. Fish and Fisheries, 2: 231–249.

Petitgas, P., Woillez, M., Doray, M., and Rivoirard, J. 2016. A geostatistical definition of hotspots for fish spatial distributions. Mathematical Geosciences, 48: 65–77.

Reid, D. G. (Ed). 2000. Report on Echo Trace Classification. ICES Cooperative Research Report No. 238. 107 pp.

Renard, D., Bez, N., Desassis, N., Beucher, H., Ors, F., and Laporte, F. 2016. RGeostats: The Geostatistical package [version 11.0.2]. MINES Paris Tech. Free download from: http://cg.ensmp.fr/rgeostats.

Rivoirard, J. 1994, Introduction to Disjunctive Kriging and Non-Linear Geostatistics. Clarendon Press, Oxford. 181 pp.

Rivoirard, J., Demange, C., Freulon, X., Lécureuil, A., and Bellot, N. 2013. A top-cut model for deposits with heavy-tailed grade distribution. Mathematical Geosciences, 45(8): 967–982.

Rivoirard, J., Simmonds, J., Foote, K. G., Fernandes, P., and Bez, N. 2000. Geostatistics for Estimating Fish Abundance. Blackwell Science, Oxford. 206 pp.

Rivoirard, J., and Wieland, K. 2001. Correcting for the effect of daylight in abundance estimation of juvenile haddock (*Melanogrammus aeglefinus*) in the North Sea: an application of kriging with external drift. ICES Journal of Marine Science, 58: 1272–1285.

Simmonds, J., and MacLennan, D. N. 2005. Fisheries Acoustics: Theory and Practice, 2nd edn. Blackwell Science, London. 437 pp.

Woillez, M., Poulard, J-C., Rivoirard, J., Petitgas, P., and Bez, N. 2007. Indices for capturing spatial patterns and their evolution in time with an application on European hake (*Merluccius merluccius*) in the Bay of Biscay. ICES Journal of Marine Science, 64: 537–550.

Woillez, M., Rivoirard, J., and Fernandes, J. 2009b. Evaluating the uncertainty of abundance estimates from acoustic surveys using geostatistical simulations. ICES Journal of Marine Science, 66: 1377–1383.

Woillez, M., Rivoirard, J., and Petitgas, P. 2009a. Notes on survey-based spatial indicators for monitoring fish populations. Aquatic Living Resources, 22(2): 155–164.

Woillez, M., Walline, P. D., Ianelli, J. N., Dorn, M. W., Wilson, C. D., and Punt A. E. 2016. Evaluating total uncertainty for biomass- and abundance-at-age estimates from eastern Bering Sea walleye pollock acoustic-trawl surveys. ICES Journal of Marine Science, 73(9): 2208–2226.

Annex 1: RGeostats package

A1.1 Introduction

Since 2001, the package RGeostats for the R environment has been developed at MINES ParisTech compiling Rscripts which call functions written in C/C++. The main characteristic of the package RGeostats is to perform geostatistical operations with no limitation on the dimension of the space and on the number of variables treated simultaneously (in the case of multivariate geostatistics).

A dedicated website, rgeostats.free.fr, has also been initiated where the user can download the latest version of the package for the relevant operating system, learn some tricks reading the numerous tutorials, get some valuable information in the section for "Frequently Asked Questions", and finally ask on the forum for some help from the RGeostats community for their own issues. It also offers the possibility to download the latest version of RGeostats for one of the following supported operating systems: Windows, Linux (32 or 64), and MacOS. This download is free of charge. The user is supposed to be familiar with R. The scripts shown in this document were developed with RGeostats version 11.0.2. With different versions of RGeostats, the functions described here and used in this document may change. Please see the help on-line and refer to the forum on rgeostats.free.fr in that case.

A1.2 Getting started with RGeostats

The user can download the latest version of RGeostats for their favorite operating system from the site http://cg.ensmp.fr/rgeostats.

When producing results or publishing papers using RGeostats, the following reference should be cited:

Renard, D., Bez, N., Desassis, N., Beucher, H., Ors, F. and Laporte, F. [year of version]. RGeostats, The Geostatistical package [version number]. MINES ParisTech. Free download from: http://cg.ensmp.fr/rgeostats

After RGeostats has been downloaded and installed on your computer, simply open the R session and load the library by typing:

```
library(RGeostats)
```

The previous command will have to be entered each time you enter into a new R session, unless you register it, once for all, in the .First file.

The RGeostats package needs the Rcpp package to be installed beforehand. Moreover, most of the applications described in this manual will suggest the installation projection and map representation packages such as maps, mapproj, and mapdata. It is, therefore, recommended to install them too.

A1.3 Description of the package

A1.3.1 General syntax

RGeostats gives access to a set of more than 400 functions. All functions obey the following standard syntax:

```
a <- function(b, c=1)
```

where *b* and *c* are the arguments and *a* is the returned value (this term is generic and does not characterize its type). The order of the arguments (*b* then *c*) is important as the function can be called without explicitly naming the arguments. Some arguments are optional (such as *c*) and have a default value (here 1) when omitted.

A1.3.2 Documentation

Each method of RGeostats package is described within a help file that can be triggered by typing:

```
?method_name
```

In this help document, each argument of the method is described, together with its default value. The value of the object resulting from this method is also described. Finally, most methods are illustrated through brief and demonstrative examples.

A1.3.3 Classes and methods

When using RGeostats, users will quickly fill their working environment with numerous objects which all belong to classes according to the S4 mechanism (this is described within the method package which is systematically loaded with R). Similarly, the package offers a large variety of functions (or methods) which are specific to a class. A mnemonic convention is used for naming a method using the class name as prefix.

A1.3.3.1 Classes

The information relative to all objects belonging to a given class is obtained in specific documentation that can be obtained by typing:

```
class?class_name
```

The various scripts or examples provided in this manual will manipulate objects belonging to most of the classes available in RGeostats. The different classes are not described in this chapter as they rely on geostatistical concepts which will be explained in the lecture notes. The following is a non-exhaustive list of the main class names and contents:

- *db*: database containing the input data and/or the output results;
- *vario*: experimental spatial characteristics calculated from data, such as experimental variograms, covariances, generalized variograms;
- *model*: model describing the spatial characteristics, such as the variogram, the covariance, or the generalized covariance model;
- *neigh*: set of parameters describing the selection of samples used for processing a target point, called neighbourhood;
- *anam*: set of parameters used to transform a sampled variable from its initial distribution to a standard Gaussian distribution, and vice versa;
- *rule*: the lithotype rule used to convert one or two Gaussian random functions into a categorical variable (facies) through thresholds;
- *thresh*: a set of intervals used to convert a variable into a categorical variable or a set of indicators;
- *polygon*: a set of one or several polysets. Each polyset is a closed broken line defined in 2-D.

The *db* class will be described in details in this section.

A class is an object which stands as a *container*, potentially with a large number of elements. The user may wish to question or to set the value of one of these elements. Some *assessors* are defined to access these elements quickly. Their syntax is as follows:

```
object$assessor
```

where *object* belongs to a given class and *assessor* is the name provided to the element of interest. The assessors of the *db* class will be described in the specific section.

Some classes contain a (single) list of elements or subclasses (e.g. the model contains a list of basic structures; the variogram contains a list of calculation directions). A specific assessor has been designed in order to reach directly one of the items of the list (without having to know the contents of the list explicitely):

```
object[item]$assessor
```

where item is the (integer) rank of the element of the list (starting from 1). An example will be provided in the chapter concerning the *db* class. If the name of the assessor is misspelled, the list of all available assessor names is listed. This can also be obtained by typing:

```
object$all
```

Finally, to discover the class to which an object belongs, it suffices to use the following command:

```
class(object)
```

A1.3.3.2 Methods

Any object belonging to a class has a set of generic methods attached according to the S4 mechanism. To get more information on these generic methods, use the command:

```
method?method_name
```

where *method_name* corresponds to the name of the generic function.

Some generic functions are available in the RGeostats package. These functions use their first argument as a signature in order to decide on the exact function that will be triggered. This signature consists of an object of a given RGeostats class. These functions are:

- *show*: prints the contents of an object belonging to a class;
- *print*: prints the contents of an object belonging to a class;
- *summary*: same as *print*, but with a shorter format;
- *plot*: displays graphically the contents of an object belonging to a class;
- *ascii.write*: dumps the contents of an object belonging to a class in a text file according to a specific format (ASCII refers to the coding of information which makes it readable);
- *digitize*: digitize an object from a graphic plot.

The advantage of such a generic function is its syntax. As a manner of fact, if *db.data* designates an object belonging to the *db* class, the following commands will give the same result:

```
plot(db.data)
db.plot(db.data)
```

The drawback is that the documentation that can be reached on the generic function *plot* is useless; only the documentation on *.plot* will provide relevant information.

A1.3.4 Mnemonic techniques

In order to navigate among more than 400 functions, the user can rely on the mnemonic habit where the name of a function is built using:

- the class of the object to which it applies as a prefix;
- the verb describing the action performed by this method (if generic) as a suffix.

The following is a non-exhaustive list of the possible suffixes:

- digit: to digitize the information from a graphic window;
- input: to define the contents of the object using a dialogue (rather than reading the values from the arguments);
- create: to initialize an object using the information provided by the arguments;
- read: to read the contents of an object from an ASCII file (organized in a manner specific to the type of the object);
- write: to write an object in an ASCII file.

A1.4 First steps in RGeostats

This paragraph describes the batch of few command lines which are necessary to load your data set inside RGeostats and to carry out a small study aiming at performing a quick estimation (using inverse squared distance interpolator) on the nodes of a regular grid, restrained within a polygon.

In this sectionr, the highlight will be placed on objects belonging to the *db* class which contain the database where the information is stored; such an object will be called a *db* (for short). The information can consist of the measurement data that the user wishes to use. Note that the results of an estimation procedure will also be stored in another database which belongs to the same class. A special case is when the data are organized as a regular grid. This refers to a grid database. Otherwise, we consider that the information is provided on a set of isolated points.

The database corresponds to a set of columns (also called fields) defined on a set of samples. The variables are numeric only and stored as real values (even if they can be printed in integer format).

A1.5 Loading data in R

Before describing the database manipulation within the RGeostats package, let us review how to load the users's data. The next paragraph is not specific to RGeostats, but it has been written in order to help the user getting started from the most general format, a text file.

A1.5.1 Loading data from a text file

The most general application is to consider that the data are coded in a text file. The easier representation consists of values contained in a file according to the representation as in the spreadsheet of the Excel Microsoft package: data is presented as a table

with one sample per row and one variable per column. The table is always a full table, which means that a variable which would not be measured must still be present and coded using a specific pattern meaning "absence of data". Rather than organizing information in a set of fixed offset columns, a more flexible format consists of writing all variables of a sample consecutively and separating them with a specific character (such as the comma, referred to the comma-separated variable or *.CSV format; or tab, usually in a text or *.txt format . Finally, it is convenient to dedicate the first line of this text file to define the names of all the variables.

Let us imagine that the data are provided as a structured table of values (46 lines and 5 columns) contained in the file called *data.ascii*, where the first line contains the names assigned to each column.

year	long	lat	depth	m.length
2003	–2.0164	59.0528	76	22.2530
2003	–0.1361	59.0513	132	24.8226
2003	–0.0875	59.0511	144	24.5158
2003	0.5348	59.3009	132	25.4600
2003	–0.5817	59.3008	126	23.7842
2003	–1.4711	59.2977	99	22.8952
2003	–1.2037	59.5516	105	24.0028
…/ …				
2003	–3.0689	60.0504	130	23.5723
2003	–1.6758	59.8011	115	25.1120
2003	–3.6691	59.9163	146	27.1296
2003	–3.4115	59.6800	145	26.5100
2003	–3.9003	59.5503	164	27.9387

The next command is used to read the data from the text file into R, inside an object which belongs to the *data.frame* class. The user must specify the character used as a column separator (here the tabulation character or "\t"). A *.CSV file would have sep=",".

```
daf <- read.csv("data.ascii", header=TRUE, sep="\t")
```

A1.5.2 Loading data from a demonstration set

To facilitate the illustrations of the methods or to illustrate the concepts described in this manual, numerous demonstration datasets are embedded in the RGeostats package. The specific command *rg.load()* enables the user to load a demonstration set (in R format directly) and to choose its name in the user's working environment. The term "demonstration dataset" has been used as its contents can be any type of RGeostats object.

The user can replace the load from a text file described in the previous paragraph by downloading the embedded demonstration set called *Demo_CRR.data.frame* and storing it in a *data.frame* called *daf*.

```
rg.load("Demo.CRR.data.frame","daf")
```

Note that the names must be specified within quotes. If you misspell the first argument, the function will return the list of all demonstration sets available within RGeostats.

A1.5.3 Data frame object

The resulting object (here called *daf*) belongs to a specific class of R called a *data.frame*. An object of this class can be considered as a matrix where each column can be addressed using its name. This particular *data.frame* contains 5 columns and 45 rows.

The following command gives the list of names of the data frame columns:

```
names(daf)
```

```
[1] "year"     "long"     "lat"      "depth"    "m.length"
```

A1.6 Creating the db from a data frame

The contents of the data frame *daf* must now be converted into an object of the *db* class called *db.data*:

```
db.data = db.create(daf)
```

Note that, as we mentioned beforehand, the name of the previous function indicates that this is a method dedicated to the *db* class.

We can check the contents of the *db.data* by typing any of the following commands:

```
db.print(db.data)
```

```
print(db.data)
```

```
db.data
```

The first syntax uses the method dedicated to the *db* class for printing its contents. The second syntax uses the generic *print* method (which triggers the use of *db.plot* for an object belonging to the *db* class). Finally, the third syntax uses the implicit generic method which launches the print (or actually summary generic method) when a command simply refers to the object name. Note that this function has no returning value.

```
Data Base Characteristics
=========================

Data Base Summary
-----------------
File is organized as a set of isolated points
Space dimension              = 0
Number of fields             = 6
Maximum Number of attributes = 6
Total number of samples      = 45

Variables
---------
Field =  1 - Name    =  rank - Locator =  rank
Field =  2 - Name    =  year - Locator =  NA
Field =  3 - Name    =  long - Locator =  NA
Field =  4 - Name    =  lat - Locator =  NA
Field =  5 - Name    =  depth - Locator =  NA
Field =  6 - Name    =  m.length - Locator =  NA
```

Note that at this stage, *db.data* refers to a database, organized as a set of isolated points, containing 45 samples for a space of dimension 0. It contains 6 fields (or attributes); their names have been inherited from the names of the columns of the data frame *daf*. Note that the new field *rank* has been added automatically; this corresponds to the rank of the sample.

A large set of functions dedicated to objects of the *db* class is available. As an example, let us mention the one which provides basic statistics on a (set of) variable(s) (ranks 6).

```
db.print(db.data,flag.stats=T,names=6)
```

The result demonstrates that the attribute called *m.length* varies from 18.5 cm to 30.5 cm:

```
Data Base Characteristics
=========================


Data Base Summary
-----------------
File is organized as a set of isolated points
Space dimension              = 2
Number of fields             = 6
Maximum Number of attributes = 6
Total number of samples      = 45

Data Base Statistics
--------------------
6 - Locator Variable z1 (Name=m.length)
 Nb of data          =         45
 Nb of active values =         45
 Minimum value       =     18.448
 Maximum value       =     30.544
 Mean value          =     26.763
 Standard Deviation  =      2.084
 Variance            =      4.344
```

Another essential function consists of adding one attribute to an already existing database, say for calculating the log of the (positive) attribute called *m.length* and storing it into the new attribute called *log.m.length*:

```
db.data = db.add(db.data,log.m.length=log(m.length))
```

The method that can be demonstrated now provides some statistics graphically. This is the correlation feature which represents the scatterplot between two variables (the average length and the depth), calculates and draws the regression line, and produces the correlation coefficient (equal to 0.54):

```
correlation(db.data,5,6,ndisc=50,flag.regr=T)
[1] 0.5424279
```

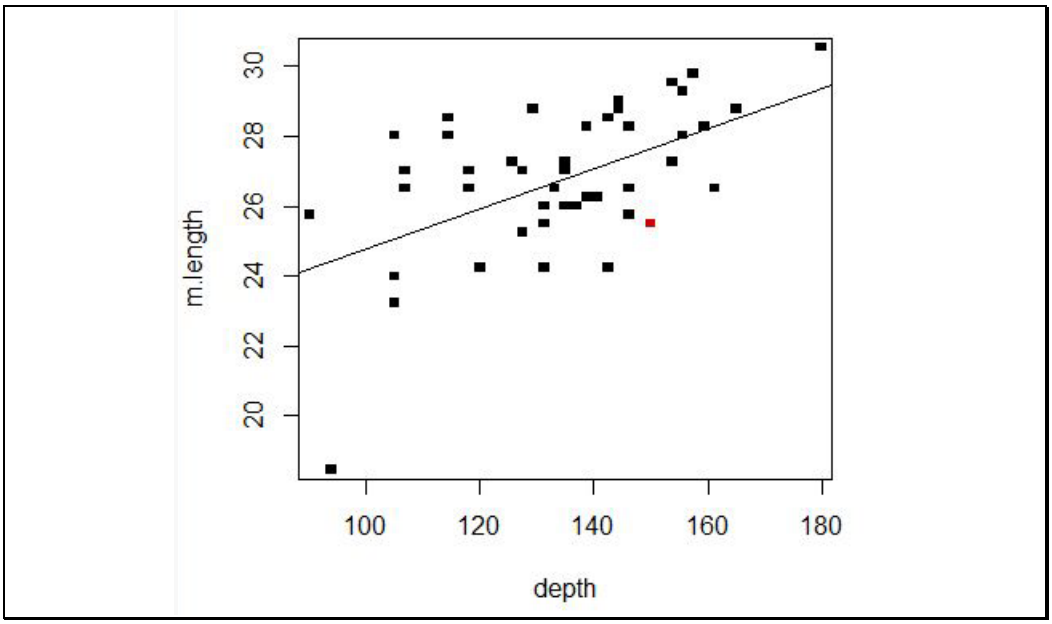


Figure A1.1. Correlation between average fish length and bottom depth.

A1.7 Locators

The concept of locator is specific to RGeostats; it allows the user to define the role attributed to each attribute (we use the term *attribute* rather than *variable* in this paragraph to avoid confusion). Note that the RGeostats concept of locator is not to be confused with the R function locator(). For example, it is necessary to define the attributes which will serve as coordinates; this is the case of the attribute called *long* which will correspond to the first (x1) coordinate and the attribute called *lat* to the second (x2) coordinate.

Note that, as RGeostats is not limited in the space dimension, coordinates are referred to as *x1,x2,x3...* This is obtained by defining the locator attached to these attributes, by typing:

```
db.data = db.locate(db.data,"long","x",1)
db.data = db.locate(db.data,"lat","x",2)
```

Note that the attribute is referenced by using its name (between quotes), but you could also use the following command, where the attribute(s) are specified by their rank(s) (or range of ranks in the next example):

```
db.data = db.locate(db.data,3:4,"x")
```

In the same manner, the user wishes to designate the attribute called *m.length* (field 6) as the variable of interest. This corresponds to the locator *z*. Once more, there is no limitation in the number of attributes that can be attached to this locator:

```
db.data = db.locate(db.data,6,"z")
```

The resulting contents of *db.data* is now as follows:

```
Data Base Characteristics
=========================
```

```
Data Base Summary
-----------------
File is organized as a set of isolated points
Space dimension              = 2
Number of fields             = 7
Maximum Number of attributes = 7
Total number of samples      = 45

Variables
---------
Field =  1 - Name    =  rank - Locator =  rank
Field =  2 - Name    =  year - Locator =  NA
Field =  3 - Name    =  long - Locator =  x1
Field =  4 - Name    =  lat - Locator =  x2
Field =  5 - Name    =  depth - Locator =  NA
Field =  6 - Name    =  m.length - Locator =  z1
Field =  7 - Name    =  log.m.length - Locator = NA
```

Note the crucial importance of the locators. We can now check that the space dimension is now equal to 2, due to the presence of the locators *x1* and *x2*. This will be important in the next operations, e.g. to doublecheck that a geostatistical model is consistent with the dimension of the space of the *db*. The operations that will be performed from now on will concern one target variable (called *m.length*) due to the presence of the locator *z1*. If we want to switch to a bivariate procedure involving two variables (namely *depth* and *m.length*), we must set the locator *z1* to the attribute *depth* and the locator *z2* to the variable *m.length*. Some attributes are left with no locator assigned; they have no specific role.

An important point is that, for a given locator type, their numbers are always consecutive and start from 1. This is the reason why, if one wishes to set the attribute *depth* as the new variable of interest (with locator *z*), it suffices to set:

```
db.data = db.locate(db.data,"depth","z")
```

The previous command automatically set the locator of the attribute *m.length* back to NA.

Finally, it is worth noticing that we can cancel the locator of a given attribute (say *depth*) by typing:

```
db.data = db.locate(db.data,"depth")
```

and even more cancel all the locators of a given type (say *z*) by typing:

```
db.data = db.erase(db.data,"z")
```

Finally, let us set the attribute *m.length* as the target variable for the rest of the paper:

```
db.data = db.locate(db.data,"m.length","z")
```

A1.8 Slots

The *db* contains a series of information such as:

- the dimension of the space where the information is defined;
- the number of samples;
- the number of variables.

In the case of a grid *db*, it also contains:

- a vector giving the number of grid cells;
- a vector giving the cell dimensions;
- a vector giving the origin of the grid;
- the rotation angles or rotation matrix.

All the previous vectors have a dimension equal to the space dimension.

It is worth adding that the origin of the grid is the node which has the lowest coordinate along each space dimension (before rotation). If a rotation is defined, the grid origin is left invariant by this rotation.

To illustrate the use of the assessors, let us mention the following possibilities for inquiring the *db* about the number of samples or the dimension of the space. This possibility is often used when writing your own scripts:

```
db.data$nsamples
```

```
[1] 45
```

```
db.data$ndim
```

```
[1] 2
```

Finally, an additional slot corresponds to the *data frame* which contains the numerical data. Its syntax mimics the one of a *matrix*. Hence, the following command which gives the value of a variable (column 6) for a given sample (row 4):

```
db.data[4,6]
```

```
[1] 23.9058
```

Another usage is to ask for the whole set of variables for a given sample (row 12):

```
db.data[12,]
```

```
   rank year    long     lat depth m.length
12   12 2002 -0.4738 59.1686   142   24.224
```

It can also be used to give the whole set of values registered for a given variable (say *m.length*):

```
db.data[,"m.length"]
```

```
[1] 25.5298 26.3873 25.4096 23.9058 25.6667 26.6263 27.0818 25.3521
18.4484
```

```
[10] 23.1362 24.1552 24.2240 26.1699 25.4391 25.9813 24.1502 25.9334
26.9289
```

```
[19] 26.4592 26.2269 25.8260 26.1136 26.3938 27.2800 27.2773 27.0710
26.5305
```

```
[28] 27.2862 28.1919 28.4917 27.9436 28.2380 28.8767 28.6661 28.3694
29.4091
```

```
[37] 29.7603 30.5435 29.3181 28.0311 28.0211 27.1142 28.7624 28.4525
29.1500
```

Note that, although the command asks for printing the contents of a column, the result is provided as a series of values printed in line.

A1.9 Graphic representation

The *db* can now be considered as a geographical database where the 45 samples are located using their longitude–latitude coordinates. The samples are represented graphically (using a specific pattern with a blue color); the area of the pattern is proportional to the (absolute) value of the target variable (the one which is associated with the locator *z1*). We can also overlay the coastline using the *mapdata* library; note the use of flag *add=T* to indicate the overlay.

```
plot(db.data,pch=19,col="blue")
map("worldHires",add=T,fill=T,col=8)
```

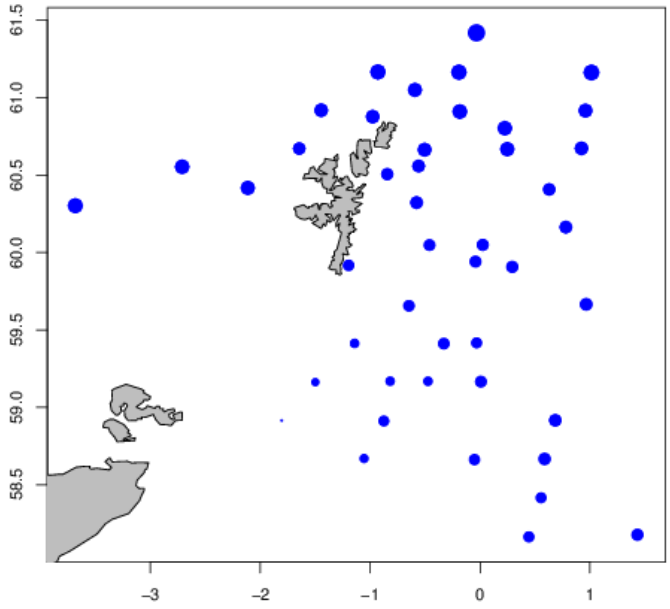


Figure A1.2. Dataset and coastline overlaid

A1.10 Projections

The samples are specified using decimalized longitude–latitude coordinates. In some cases, it is essential to convert these spherical coordinates into orthonormal ones. This is obtained through a projection system. RGeostats is connected with the projection package called *mapproj* which handles a complete set of complex projections. However, a basic projection (called *mean*) is provided which is valid for low or medium latitudes and short distances; the parameters of this projection are calibrated on the mean coordinates of the dataset. The projection definition only requires the name of the *db*. Then, this projection will apply to all graphic representation of spatial objects (belonging to the class *db* or *polygon* for example).

```
projec.define(projection="mean",db=db.data)
plot(db.data,pch=19,col="blue")
```

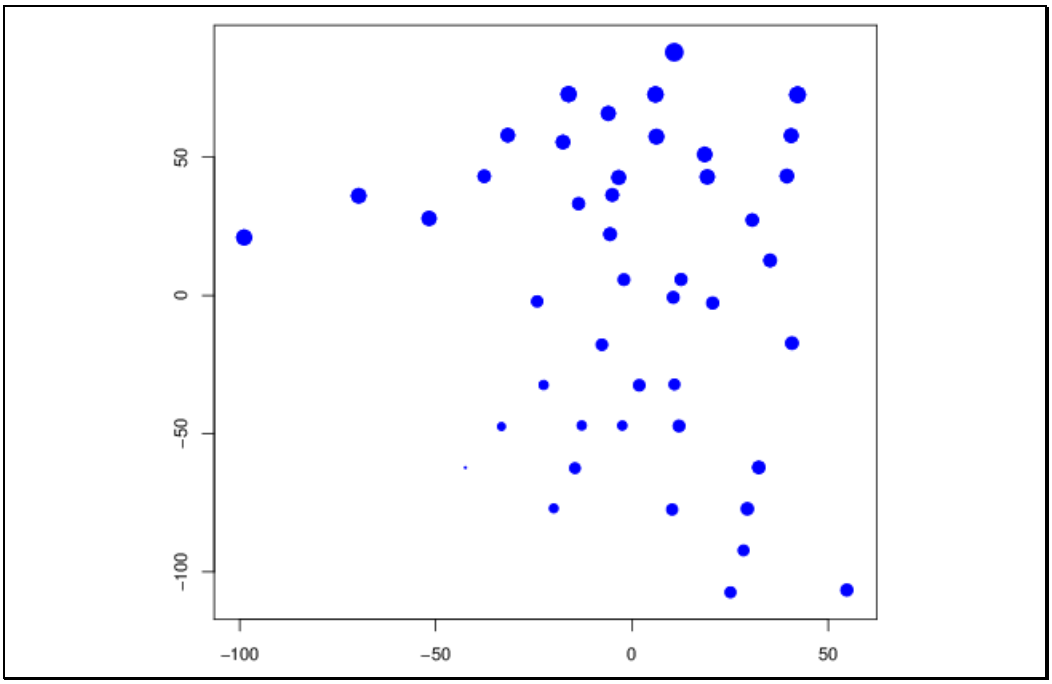


Figure A1.3. Dataset displayed in the projected coordinates

The projection will impact all the subsequent operations such as the calculation of the variogram or the estimation using kriging, where the distances will be established in the projected system.

The projection system remains valid until it is cancelled. This is even true when leaving and re-entering a session. A projection is never deleted, but it can be turned off (we are then back to the natural system) by typing:

```
projec.toggle(0)
```

If the previous command is entered again, the projection is active again, with the same parameters as the ones defined initially.

It is important to note that some packages such as *maps* do not cope with the projections defined in RGeostats. This is the reason why the overlay of figures produced with *RGeostats* and *maps* should be performed only in the natural system.

A1.11 Selections

We wish to discard temporarily (but not remove from the database) some samples, e.g. the ones where the mean length (attribute *m.length*) is smaller than 25. The corresponding command is:

```
db.data = db.sel(db.data,m.length > 25)
```

In the previous command, note that the attribute is specified by its name (without quotes for better efficiency); using it by number would be confusing.

The modified *db* is printed as follows:

```
Data Base Characteristics
=========================
```

```
Data Base Summary
-----------------
File is organized as a set of isolated points
Space dimension              = 2
Number of fields             = 8
Maximum Number of attributes = 8
Total number of samples      = 45
Number of active samples     = 39

Variables
---------
Field =  1 - Name    =  rank - Locator =  rank
Field =  2 - Name    =  year - Locator =  NA
Field =  3 - Name    =  long - Locator =  x1
Field =  4 - Name    =  lat - Locator =  x2
Field =  5 - Name    =  depth - Locator =  NA
Field =  6 - Name    =  m.length - Locator =  z1
Field =  7 - Name    =  log.m.length - Locator =  NA
Field =  8 - Name    =  sel - Locator =  sel
```

Note that the selection corresponds to the new variable (Field 8) which is assigned to the locator called *sel*. From the total number of samples (45), 39 are considered as active.

The principle is that all subsequent calculations will be performed based on the active samples as soon as a variable is attached to the *sel* locator or on the whole dataset when no variable is assigned the *sel* locator. There can only be one *sel* locator defined at a time.

A1.12 Defining a polygon

The next operation consists of defining a polygon which will restrain the area of interest. The polygon is comprised of a number of polysets; each polyset is a 2-D polygonal closed shape. This polygon belongs to the *polygon* class. It can either be imported from a text file, loaded from an embedded dataset, or directly digitized from the previous plot.

A1.12.1 Digitizing a polygon from a graphic plot

When digitized from an already existing plot, the corresponding command is:

```
poly.data = polygon.digit()
```

where the user must pick the vertices of the polyset, close it, and possibly resume with a second polyset. The operation is repeated until the whole polygon is defined.

The result file (called *poly.data*) is directly an object which belongs to the *polygon* class.

A1.12.2 Loading a polygon from a text file

When reading it from the text file, the polygon is limited to a single polyset. The file format contains two columns as described below:

```
"lon" "lat"
-3.5 58
-3.5 58.20265
```

```
-3.48146 58.2115
-3.46421 58.2262
-3.45236 58.2498
-3.41261 58.2643
-3.37841 58.2818
-3.34151 58.2948
…/…
-1 61.75
2 61.75
2 58
-3.5 58
```

The text file is loaded in R as a data.frame using the same command as before:

```
polydaf <- read.csv("poly.dat",header=T,sep=" ")
```

The next command creates an object belonging to the polygon class of RGeostats:

```
poly.data <- polygon.create(polydaf)
```

A1.12.3 Loading a polygon from a demonstration set

For convenience, a dedicated polygon can be loaded directly from the embedded demonstration sets using the command:

```
rg.load("CRR.demo.poly.data","poly.data")
```

A1.13 Selection using the polygon

The polygon created previously can be represented on top of the data. The active data are represented by a point and the value of the variable *m.length*:

```
plot(db.data,col="blue",name.literal="m.length",name.post=1)
plot(poly.data,add=T,col="red",pch=4)
```

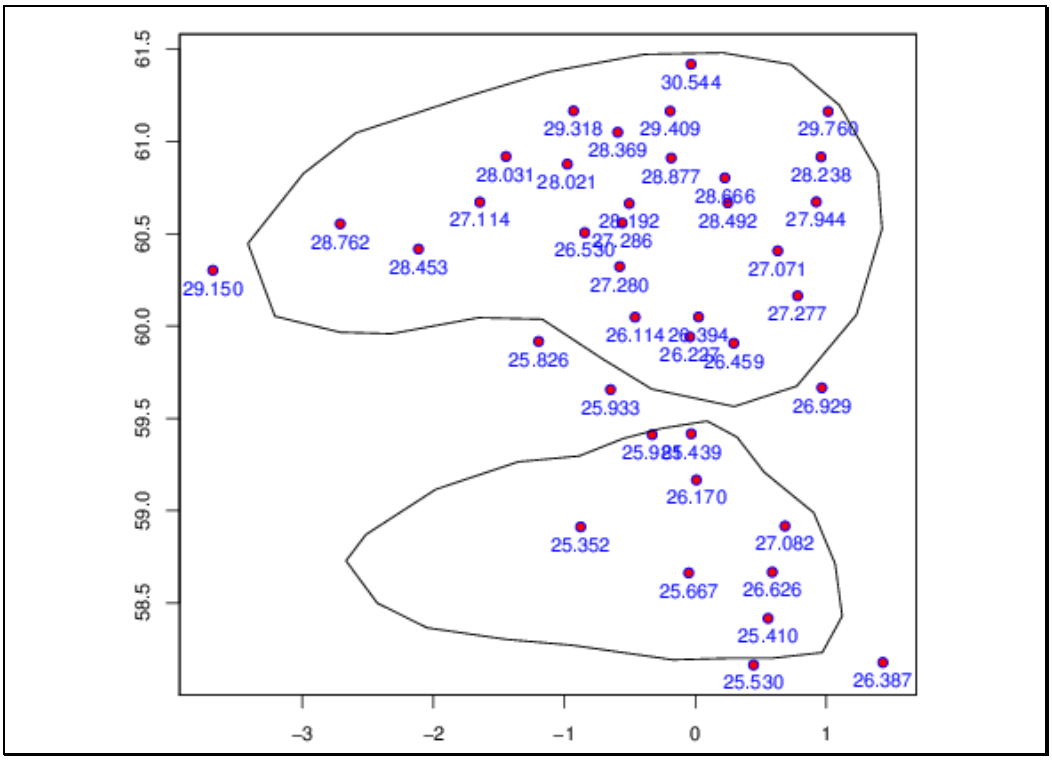


Figure A1.4. Dataset and polygons overlaid.

At this point, the samples which are represented correspond to the selection currently active (i.e. the ones where the attribute *m.length* > 25).

We can now use the polygon in order to select the only samples which are included within the polygon.

```
db.data = db.polygon(db.data,poly.data)
```

This new selection does not consider the previous active selection (unless asked through the arguments). It simply creates another selection called *polygon* which is now active; the previous selection has been deactivated. When plotting the data again (masked by the previous polygon) together with the polygon, we now obtain the following figure where we can check the presence of samples where *m.length* is smaller than 25.

```
plot(db.data,col="blue",name.literal=”m.length”,name.post=1)
plot(poly.data,add=T,col=”red”,pch=4)
```

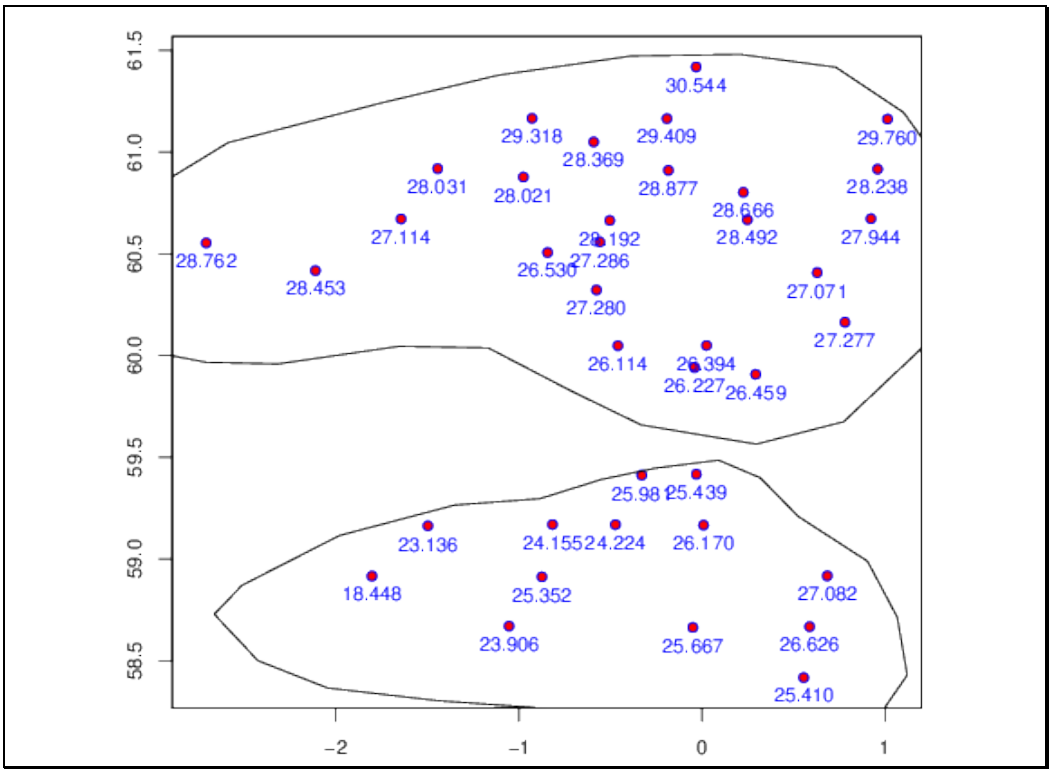


Figure A1.5. Dataset contained inside the polygon.

A1.14 Creating the interpolation grid

The estimation grid (called *grid.data*) must now be defined. This can be done automatically for a grid covering the dataset with 100 × 100 nodes:

```
db.grid = db.grid.init(db.data,nodes=100)
```

The polygon *poly.data* is used again, this time to mask off the nodes located outside the polygon. This selection will be taken into account in any subsequent calculation, which avoids performing calculations over the discarded cells. When printing the resulting *db.grid*, we can read that the number of active grid nodes is 7471 (out of the initial 10 000).

```
db.grid = db.polygon(db.grid,poly.data)
```

It is now time to perform the estimation of the target variable, starting from the 39 samples where it has been measured, down to the nodes of the grid (the ones included in the polygon).

```
db.grid <- invdist(db.data, db.grid)
```

The final plot represents the interpolated variable, together with the initial information and the polygon.

```
plot(db.grid, pos.legend=3)
plot(db.data,pch=21,col="yellow",bg="black",add=T)
plot(poly.dat,add=T)
```

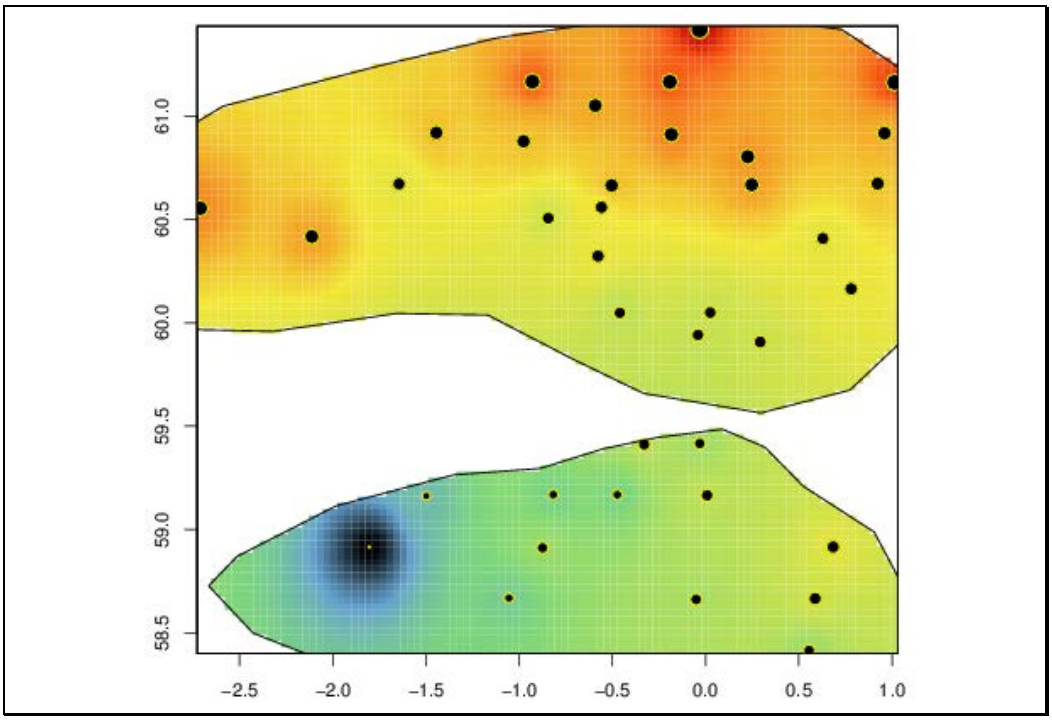


Figure A1.6. Estimation by inverse squared distance method and dataset.

A1.15 Main functions

In this paragraph, we describe in detail a small set of essential functions which will be used throughout almost any of the applications provided in this manual:

- *vario.calc*: Calculation of the experimental spatial structure (i.e. covariance, variogram, or transitive covariogram) from a set of isolated datapoints. This operation requires a *db* in input and produces a *vario* in output.
- *model.auto*: Procedure used to automatically fit a geostatistical model on the experimental variogram calculated for one or several variables in one or several directions. This operation requires a *vario* in input and produces a *model* in output.
- *global*: Procedure used to calculate the global variance of estimation over an area, starting from a dataset and a geostatistical model. This procedure requires a *db* (for the conditioning information), a *model* (for the geostatistical model), and a *polygon* (to delineate the domain) in input. It produces the global estimate and variance of estimation in output.
- *kriging*: Procedure used to perform the estimation of one or several variables using kriging at a set of target points (usually located on the nodes of a grid). This procedure requires an input *db* (for conditioning information), an output *db* (for the set of target sites), a *model* (for the geostatistical model), and a *neigh* (for the conditioning neighbourhood) in input. It produces the output *db* containing the new attributes: the estimation and the standard deviation of the estimation in output.
- *krigtest*: Procedure used to perform the kriging procedure on a single target and produce all the intermediate calculations. It uses the same input as the *kriging* procedure.

- *anam.fit*: Procedure used to calculate the gaussian anamorphosis fitted on a set of samples. This procedure requires a *db* (for the data information) in input and produces an *anam* (for the modeled anamorphosis) in output.
- *simtub*: Procedure used to perform conditional or non-conditional simulations of one or several variables at a set of target points. This procedure requires an input *db* (only in the case of conditional simulations), an output *db* (for the set of target sites), a *model* (for the geostatistical model), and a *neigh* (for the conditioning neighbourhood) in input. It produces the output *db* containing the simulated outcomes as new attributes.

Annex 2: Data

The data used throughout the document in the demonstration Rscripts and applications are provided below in detail, together with their corresponding survey design:

- Hake in the Bay of Biscay, demersal trawl survey
- Hake in the Gulf of Lion, demersal trawl survey
- Octopus in Marocco, demersal trawl survey
- Herring eggs in a spawning bed in Scotland, dredge survey
- Anchovy in the Bay of Biscay, acoustic survey
- Herring in the northern North Sea, acoustic and trawl survey
- Acoustic backscatter of pelagic fish in Mauritania, acoustic survey

A2.1 Bay of Biscay hake (trawl survey)

The case study comes from the French groundfish survey EVHOE series (1987–2015) carried out by Ifremer on the eastern continental shelf of the Bay of Biscay and the Celtic Sea (ICES, 2015). Sampling was randomly stratified according to latitude and depth, with strata depth ranging from 15 to 600 m. The number of hauls per survey varied depending mostly on weather conditions. A 36/47 Grande Ouverture Verticale (GOV) trawl was used, with a codend liner of 20-mm mesh. Haul duration was 30 min at a towing speed of 4 knots, mainly in daylight. Catch weights and numbers were recorded for all species, mostly demersal. For some species, such as the European hake (*Merlucius merlucius* L.), sex and total length were recorded, and otoliths were extracted and examined in the laboratory to build age–length keys (ALKs) by sex. These keys were used to transform the length frequencies observed at each trawl station into age frequencies. Here, we considered the survey data for hake in 1987 over a study area between 48°30′N and 43°30′N. Age 0 and age 1 hake densities were considered for the case study. They were converted from numbers of fish caught per hour trawled to numbers of fish caught per nautical mile^2, assuming that the area swept in 30 min of trawling was 0.02 nautical mile^2. These data are used to illustrate the chapter on spatial distribution indices. A more extensive analysis of this dat set using spatial indices can be found in Woillez *et al*. (2007).

Within the RGeostats library, two RGeostats objects are available for this case study and can be loaded using the function rg.load(). First, there is a two-dimensional RGeostats database named “Demo.hake.bob.db.data”, which contains the following six fields:

- "rank" is the rank of the sample;
- "long" is the longitude of the trawl sampling location in decimal degrees;
- "lat" is the latitude of the trawl sampling location in decimal degrees;
- "A0" is the density of age 0 hake in the trawl sampling location (no. nautical mile^{-2});
- "A1" is the density of age 1 hake in the trawl sampling location (no. nautical mile^{-2}).

Then, there is a RGeostats polygon named "Demo.hake.bob.poly.data", which contains the coordinates of the polygon vertices in decimal degree for the EVHOE survey series. This polygon has been defined to encompass the survey area according to depth isobaths, coastline, and latitude. It is used to limit the extension of the area of influence computation for the samples that are at the edge of the survey area.

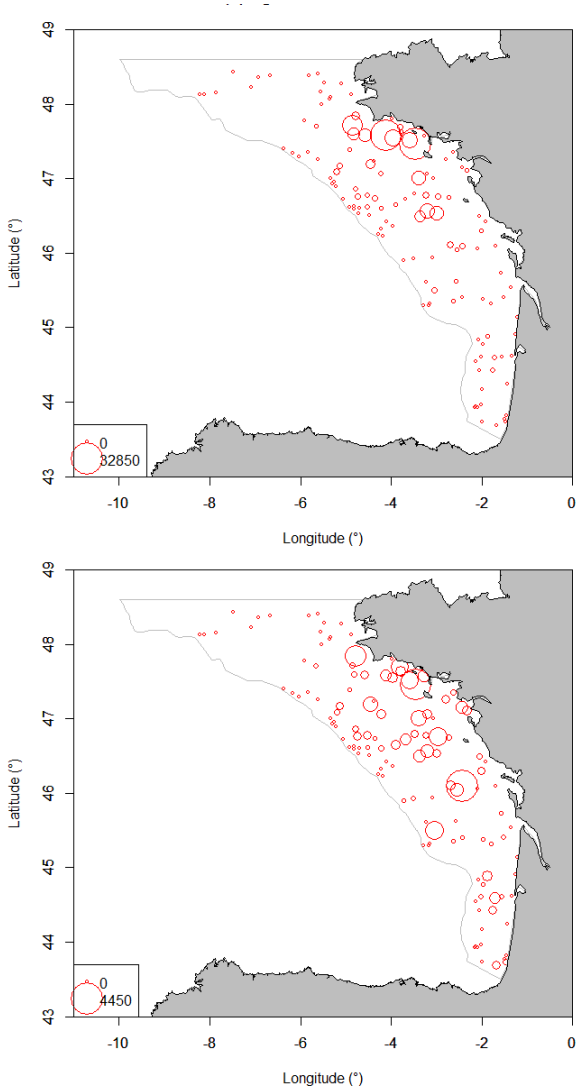


Figure A2.1. Proportional representation of (a) the age 0 and (b) age 1 hake 1987 data. Number of data: n = 127; mean: m_{A0} = 2191, and m_{A1} = 491; coefficient of variation: CV_{A0} = 2.39 and CV_{A1} = 1.62; frequency of zeroes: $f0_{A0}$ = 0.338 and $f0_{A1}$ = 0.378; maximum value: max_{A0} = 32 850 and max_{A1} = 4450. The polygon used to delineate the survey area is represented in light grey on both figures

A2.2 Gulf of Lion hake (trawl survey)

This dataset was provided by Angélique Jadaud at Ifremer, Sète, France. The data come from the "International Bottom Trawl Survey in the Mediterranean Sea" (MEDITS) project, conducted every year since 1994 in May–June. In the Gulf of Lion, 66 stations are defined according to a stratified random sampling desgn based on five depth strata (10–50 m, 50–100 m, 100–200 m, 200–500 m, and 500–800 m) divided into east–west substrata by 4°E longitude. Hauls have been performed according to the same protocol since 1994. The hauls are 30 min for shelf stations (10–200 m) and 60 min on the upper slope (>200 m, to compensate for a lower catchability on irregular grounds). Georeferenced position, speed, and distance covered by the trawl are systematically recorded. The experimental net (GOC 73) used for sampling has a 20-mm-diamond-stretched mesh in the codend. An underwater Scanmar system is used to control the trawl ge-

ometry and eliminate the analysis for tows not properly sampled. Horizontal and vertical openings of the gear are ca. 18 and 2 m, respectively. The catch is sorted, counted, and weighed by species. The survey provides the density of individuals for each species by dividing the observed counts by the trawled surface. It is an indicator of local abundance relative to trawl catchability, which is assumed to be constant.

Since the onset of the MEDITS survey, 300 different species have been identified in the Gulf of Lion, but many display low abundance or are rare. Here, we decided to use European hake (*Merluccius merluccius*) densities as an example to compute varirogams.

Within the RGeostats library, one RGeostats object is available for this case study and can be loaded using the function rg.load(). This is a two-dimensional RGeostats database named "Demo.hake.med.db.data", which contains the following fields:

- "rank" is the rank of the sample;
- "STATION_NAME" is the name of the trawl station;
- "YEAR" is the survey year;
- "Lat" is the latitude of the trawl station in decimal degrees;
- "Long" is the longitude of the trawl station in decimal degrees;
- "Prof" is the bottom depth at the trawl station;
- "DISTANCE" is the distance over which the trawl has been towed;
- "WING_OPENING" is the wing opening of the trawl;
- "AREA" is the number of the geographical subarea (GSA);
- "HAUL_DURATION" is the haul duration;
- "MERLMER" is the hake density in number of individuals km^{-2}.

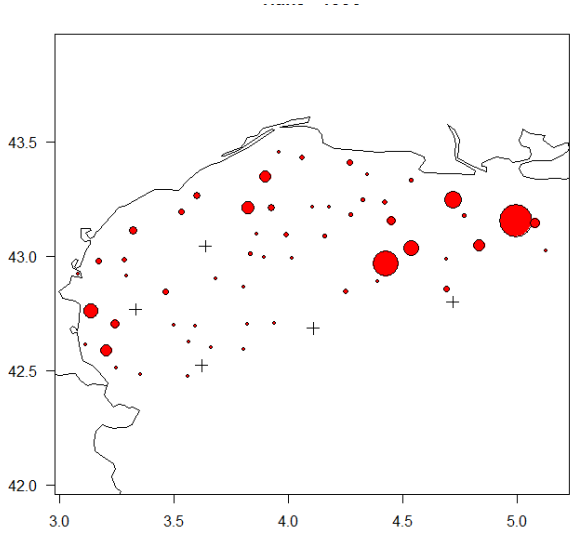


Figure A2.2. European hake (*Merluccius merluccius*) densities recorded during the 1996 MEDITS survey. Crosses represent zero densities. Circle sizes are proportional to the values of the positive densities. Number of data: n = 61; mean: m = 2120; coefficient of variation: CV = 1.65; frequency of zeroes: f_0 = 0.08; maximum value: max = 19 353.89.

A2.3 Octopus off Morocco (trawl survey)

This dataset was provided by Abdelmalek Faraj at the Institut National de Recherches Halieutiques (INRH), Casablanca, Morocco. The data are derived from the Moroccan monitoring trawl surveys carried out since 1980 by the INRH. Since 1998, two surveys have been carried out each year covering the continental shelf between 20°50'N and

26°00N from the coast to a depth of 100 m. The autumn surveys allow estimating recruitment and depicting its geographical patterns and extension. A stratified random sampling design is performed, with each sample being located randomly inside a cell of 11 × 11 nautical miles and independently from the other cells. In 1999, the survey comprised 107 sampling stations. Swept areas were computed on the basis of horizontal trawl opening, which is monitored continuously, and towing speed. Haul duration is standardized. In 1999, haul duration was 12 min, leading to an average swept area of 30 000 m². Octopus (*Octopus vulgaris*) yields were divided by swept area and expressed in terms of density, i.e. number of octopus per nautical mile².

The catch from each tow is weighed and measured, and the sex and stage of maturity are noted. The study variable is thus the density of juveniles expressed as the number of juveniles per nautical mile². Juveniles correspond to the small commercial size categories, i.e. categories Tako 8 and Tako 9, according to Japanese classification. Landing these categories is prohibited by Moroccan fishery legislation.

Within the RGeostats library, two RGeostats objects are available for this case study and can be loaded using the function rg.load(). First, there is a two-dimensional RGeostats data base named "Demo.octopus.morocco.db.data", which contains the following fields:

- "rank" is the sample rank;
- "SURVEYS" is the survey code (CI1099CF);
- "lat" is the latitude of the trawl sampling location in decimal degrees;
- "long" is the longitude of the trawl sampling location in decimal degrees;
- "DEPTH" is the bottom depth at the trawl station;
- "AIRE" is the area trawled;
- "JUV" is the density of juvenile in no. nautical mile^{-2};
- "NBTOT" is the number of individuals.

Then, there is a RGeostats polygon named "Demo.octopus.morocco.poly.data", which has been defined to encompass the survey area. It is used to limit the extension of the surface of influence computation for the samples that are at the edge of the survey area and to select relevant grid nodes for the kriging map.

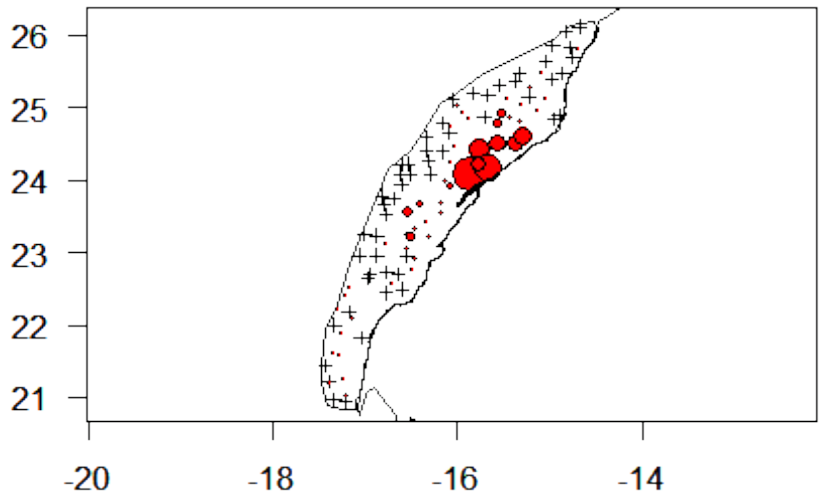


Figure A2.3. Octopus densities (1999) in number of juveniles per nautical mile². Black crosses represent null densities. Red circle areas are proportional to positive values. The polygon delineates the area to be mapped. Number of data: n = 107; mean: m = 107 910; coefficient of variation: CV = 2.39; frequency of zeroes: f_0 = 0.48; maximum value: max = 1 617 791.42.

A2.4 Herring eggs west of Scotland (dredge survey)

This case study data was provided by Marine Scotland Science at the Marine Laboratory, Aberdeen, UK. Herring eggs are benthic and are laid on well-identified gravel beds. Thus, the precise survey of a spawning bed can be undertaken using a dredge. The dataset results from a survey conducted over one herring spawning bed in the Firth of Clyde, west of Scotland. The survey design is pseudoregular. Egg counts are located at stations where dredge tows have been made. This case study is used to illustrate kriging (mapping) and global estimation in 2 D and particularly to understand the benefits of kriging, i.e. weighting the sample points optimally depending on the variogram structure.

Within the RGeostats library, two RGeostats objects are available for this case study and can be loaded using the function rg.load(). First, there is a two-dimensional RGeostats data base named "Demo.herreggs.scot.db.data", which contains the following five fields:

- "rank" is the sample rank;
- "x" is the x-coordinate of the dredge station in projected space;
- "y" is the y-coordinate of the dredge station in projected space;
- "eggs" is the egg count at the dredge station;
- "sel" is the relevant dredge station contained in the survey area (i.e. the polygon).

Then, there is a RGeostats polygon named "Demo.herreggs.scot.poly.data", which defines the geographical limits of the study areas.

Note that the coordinates from the database and the polygon are already transformed in km allowing the user to compute distances directly (no projection needed).

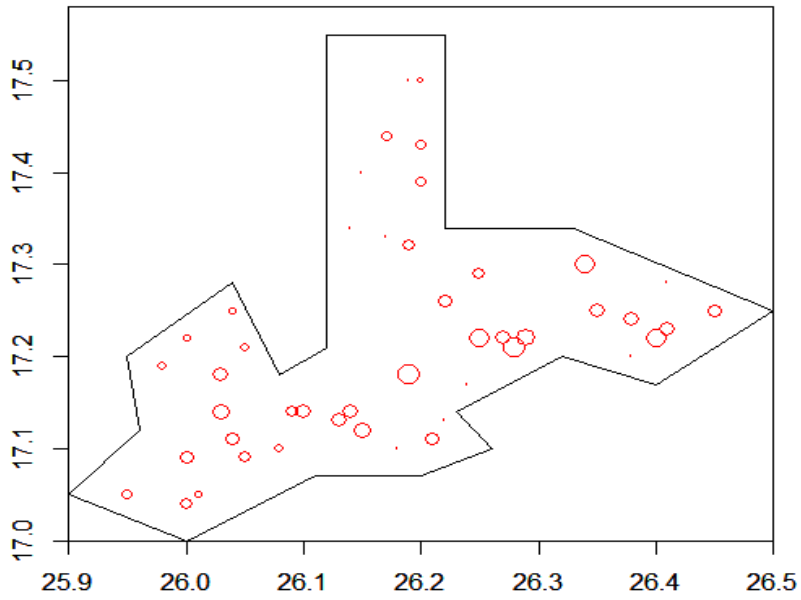


Figure A2.4. Proportional representation of the Marine Laboratory egg data. Number of data: n = 46; mean: m = 963.26; coefficient of variation: CV = 0.62; frequency of zeroes: f_0 = 0.20; maximum value: max = 2064.

A2.5 Bay of Biscay anchovy (acoustic survey) in 2D and 1D

These case study data were provided by Ifremer. Small pelagic fish resources, i.e. anchovy, sardine (*Sardina pilchardus*), sprat (*Sprattus sprattus*), mackerel (*Scomber scombrus*), and horse mackerel (*Trachurus trachurus*), are assessed yearly over the French shelf of the Bay of Biscay by the acoustic survey PELGAS undertaken by Ifremer. This survey is coordinated with Spanish and Portuguese surveys by the ICES Working Group on Acoustic and Egg Surveys (WGACEGG). The survey consists of 28 regularly spaced transects 12 nautical miles apart (perpendicular to the isobaths) from the coast (20 m bottom depth) to the shelfbreak. Fish acoustic backscatter at 38 kHz (and also at other frequencies) are recorded continuously along the transects, and values are integrated vertically and horizontally (sA in m^2 nautical $mile^{-2}$) in bins or ESDUs (elementary sampling distance units) of 1 nautical mile along the ship's sailing track. Opportunistic trawl hauls are undertaken to determine echo-traces to species as well as estimate fish length and age. Trawl data and acoustic sA data are combined using standard methodology to derive an estimate of biomass (t nautical $mile^{-2}$) by species every nautical mile along the survey track (Simmonds and McLennan, 2005; Doray *et al*., 2010). Here, we considered the survey data for anchovy in 2002, which provide a typical example of acoustic data with a high proportion of zeroes and a heavy-tailed histogram skewed to the right and a short correlation range (5–10 nautical miles) along the transects.

Within the RGeostats library, two datasets are provided. The first dataset illustrates the chapter involving indicators (the border effects, the topcut model). There is a two-dimensional RGeostats database named "Demo.anchovy.bob.2d.db.data", which contains the 16 southern-most transects corresponding to the main area of anchovy presence (south of the Isle of Yeu) and the following four fields:

- "rank" is the sample rank;
- "LAT" is the latitude of the ESDUs in decimal degrees;
- "Long" is the longitude of the ESDUs in decimal degrees;
- "ENGR.ENC" is the biomass of anchovy in the ESDUs (t nautical $mile^{-2}$).

A RGeostats polygon named "Demo.anchovy.bob.2d.poly.data" is associated with this 2D dataset and contains the coordinates of the polygon vertices in decimal degree.

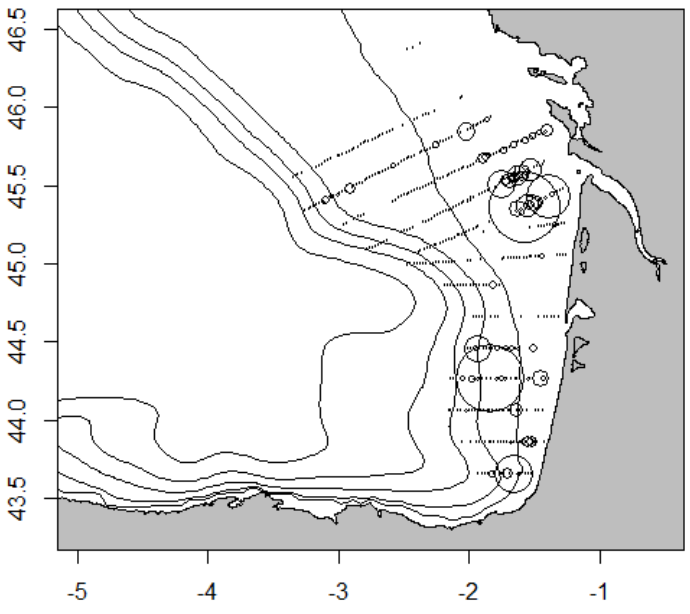


Figure A2.5. Proportional representation of the 2D anchovy 2002 data. Number of data: n = 822; mean: m = 12.83; coefficient of variation: CV = 3.75; frequency of zeroes: f_0 = 0.434; maximum value: max = 701.

The second dataset illustrates the chapter about global estimation variance using the transitive approach. It uses a one-dimensional dataset formed by summing values along the continuously sampled 28 transects over the entire Bay of Biscay (Petitgas, 1993a). For this dataset, there is one RGeostats object available within the RGeostats library that can be loaded using the function rg.load(). This is a one-dimensional RGeostats database named "Demo.anchovy.bob.1d.db.data", which contains the following fields:

- "rank" is the sample rank;
- "x1" is the sample rank starting from 0 instead of 1;
- "Transect" is the transect code;
- "Tr.length" is the transect length (in nautical miles);
- "Tr.biomass" is the summed anchovy biomass along each transect (t nautical mile^{-2}).

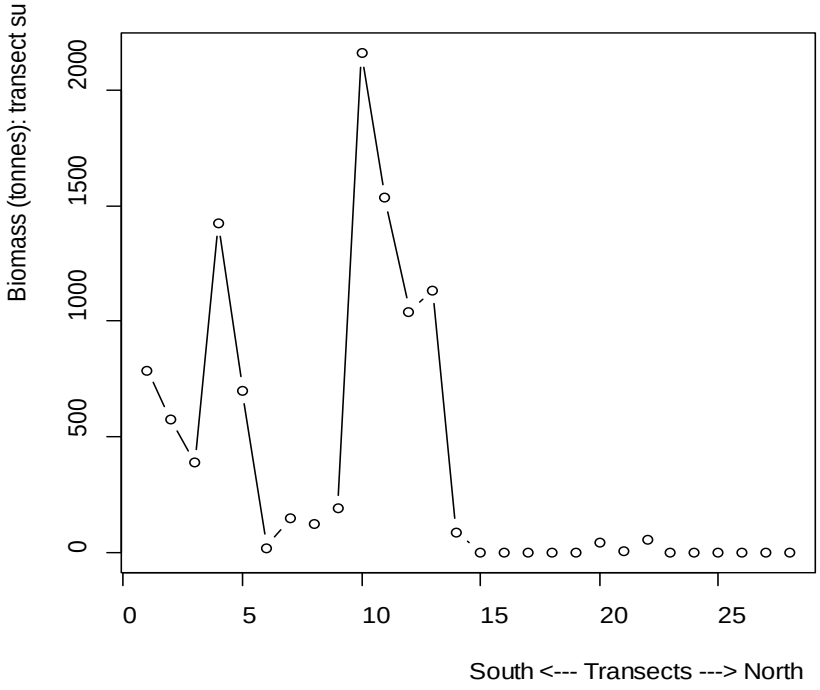


Figure A2.6. One-dimensional representation of anchovy in 2002; biomass summed along the transects. Number of transects: n = 28; mean: m = 371.74; coefficient of variation: CV = 1.57; inter-transect distance: 12 nautical miles.

A2.6 Scottish North Sea herring (acoustic–trawl survey)

Thess case study data were provided by Marine Scotland Science at the Marine Laboratory, Aberdeen, UK. Acoustic–trawl surveys have been conducted in the northern North Sea (northwestern half of ICES Division IVa) in midsummer of each year since 1979 on the prespawning concentration of autumn-spawning Atlantic herring (*Clupea harengus*). These surveys are part of the larger international survey for North Sea herring. The result of the larger survey is used to tune the assessment, which ultimately aims to determine biomass estimates of the North Sea herring stock (e.g. ICES, 2006).

The Scottish survey design consists of longitudinal transect lines covering a domain surveyed by the research vessel RV "Scotia" around Orkney and Shetland. Transects are laid down in a systematic manner with a random start point, and the transect spacing is chosen according to historical levels of abundance at 30, 15, or 7.5 nautical miles. The surveyed domains are defined by the ICES Planning Group for Herring Surveys (ICES, 2006). Calibrated acoustic-backscatter data were recorded using a Simrad EK500 echosounder operating at 38 kHz and scrutinized to estimate nautical-area-scattering coefficients attributed to herring for 15 min (2.5 nautical miles) equivalent distance sampling units (EDSU). Trawl hauls were taken regularly to assist in the scrutiny process and to collect biological data, such as fish length and fish age, following Simmonds and MacLennan (2005). Here, we considered the acoustic backscatter (sA in m^2 nautical $mile^{-2}$) and the mean length (cm) data recorded in 2003. The use of the mean length was justified because of a small variation in length at each trawl station. These data are used to illustrate the chapter about the multivariate geostatistics and the geostatistical simulation (more details in Woillez *et al*., 2009b).

Two datasets are available for this case study. The first dataset concerns the acoustic backscatter attributed to herring recorded along transects. Within the RGeostats library, there is a two-dimensional database named "Demo.herring.sa.scot.db.data", which can be loaded using the function rg.load(). It contains the following five fields:

- "rank" is the sample rank;
- "year" is the survey year;
- "lon" is the longitude of the ESDUs in decimal degrees;
- "lat" is the latitude of the ESDUs in decimal degrees;
- "sa" is the acoustic backscatter (in m^2 nautical $mile^{-2}$) attributed to herring.

The second dataset concerns the mean length data of herring computed from the trawl locations. A two-dimensional database named "Demo.herring.len.scot.db.data" can be loaded using the function rg.load() from the RGeostats library. It contains the following six fields:

- "rank" is the sample rank;
- "year" is the survey year;
- "lon" is the longitude of the trawl station in decimal degrees;
- "lat" is the latitude of the trawl station in decimal degrees;
- "depth" is the bottom depth measured at the trawl station;
- "m.length" is the mean length (in cm) of herring.

Then, for both datasets, the following two RGeostats polygons are available: "Demo.herring.sa.scot.poly.data" and "Demo.herring.len.scot.poly.data". Both contain the coordinates of the polygon vertices in decimal degree for the survey in 2003. They have been defined to encompass the survey area according to the ICES Planning Group for Herring Survey (PGHERS) (ICES, 2006). They are used to limit the extension of the areas of influence computation for the samples that are at the edge of the survey area and to select relevant grid nodes for kriging or simulated map.

A depth grid is also provided to complement the mean length dataset when used as a auxiliary variable in the chapter of the multivariate geostatistics. The RGeostats grid is named "Demo.herring.len.scot.grid.kriging" and contains the following four fields:

- "lon" is the longitude of the grid nodes in decimal degrees;
- "lat" is the latitude of the grid nodes in decimal degrees;
- "depth" is the bottom depth (in m) estimated at the grid nodes;
- "sel" is the relevant grid nodes contained within the survey area (i.e. the polygon).

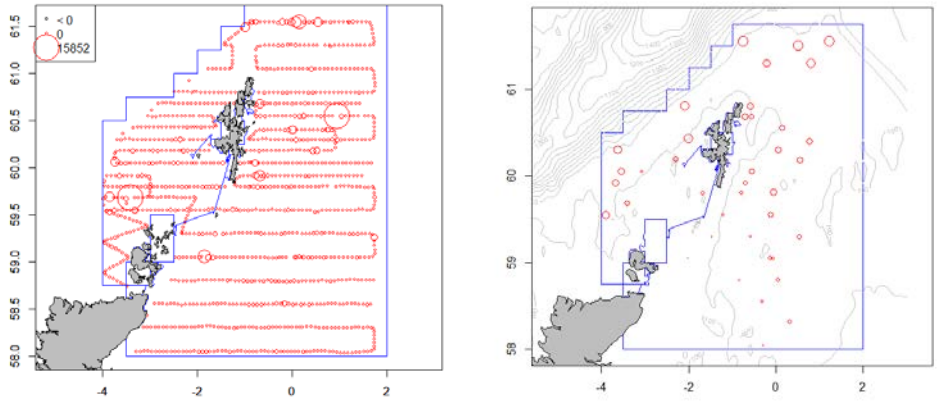


Figure A2.7. Left: proportional representation of the acoustic backscatter data attributed to herring. Number of data: n = 1108; mean: m = 334.75; coefficient of variation: CV = 2.79; frequency of zeroes: f_0 = 0.239; maximum value: max = 15 852. Right: proportional representation of the mean length of herring at stations. Number of data: n = 39; mean: m = 26.195; coefficient of variation: CV = 0.079; maximum value: max = 30.07. The polygon used to delineate the area to be mapped is represented in blue on both figures, while the grid depth, which will be used as auxiliary variable, is represented with contour lines on the right figure only.

A2.7 Mauritanian pelagic fish (acoustic survey)

This dataset was provided by Cheikh-Baye Braham at the Institut Mauritanien de Recherches Océanographiques et des Pêches (IMROP), Nouadhibou, Mauritania. The acoustic data were collected day and night during four Mauritanian national surveys carried out by the RV "Al-Awam" (2007–2010) during –December of each year. The sampling scheme followed transects oriented perpendicular to the coast from depths greater than 10 m and up to 500 m. Radials were 10 nautical miles apart. Details on the "Al-Awam" surveys can be found in the IMROP reports (webmaster@imrop.mr). Surveys were conducted using the Simrad EK-500, dual-frequency 38 and 120 kHz, threshold for filtering the echoes of −70 dB, calibrated by the standard sphere method (Foote *et al.*, 1987) and the same elementary sampling distance unit (ESDU) was equal to 5 nautical miles. During these surveys, the objective was to identify acoustic echoes at the species level whenever possible (MacLennan and Simmonds, 1992; Reid, 2000) using the Bergen integrator (Knudsen, 1990). Where such detail was not achievable, the energy was allocated to a wider group based on a combination of a visual scrutiny of the behavior pattern, as deduced from echo diagrams, and the catch compositions. The present study was focused on the entire pelagic community. Following MacLennan *et al*. (2002), we used the acoustic energy (i.e., the nautical area scattering coefficient (NASC)), usually denoted (sA).

Within the RGeostats library, two RGeostats objects are available for this case study and can be loaded using the function rg.load(). First, there is a two-dimensional RGeostats database named "Demo.acoustic.maur.db.data", which contains the following five fields:

- "rank" is the sample rank;
- "an" is the surveyed year;
- "lat" is the latitude of the ESDUs in decimal degrees;
- "long" is the longitude of the ESDUs in decimal degrees;
- "Total" is the total acoustic backscatter attributed to pelagic fish (sA in m^2 nautical $mile^{-2}$).

Then, there is a RGeostats polygon named "Demo.acoustic.maur.poly.data", which contains the coordinates of the polygon vertices in decimal degree. It has been defined to encompass the survey area and is used to limit the extension of the areas of influence computation for the samples that are at the edge of the survey area and to select relevant grid nodes for kriging map.

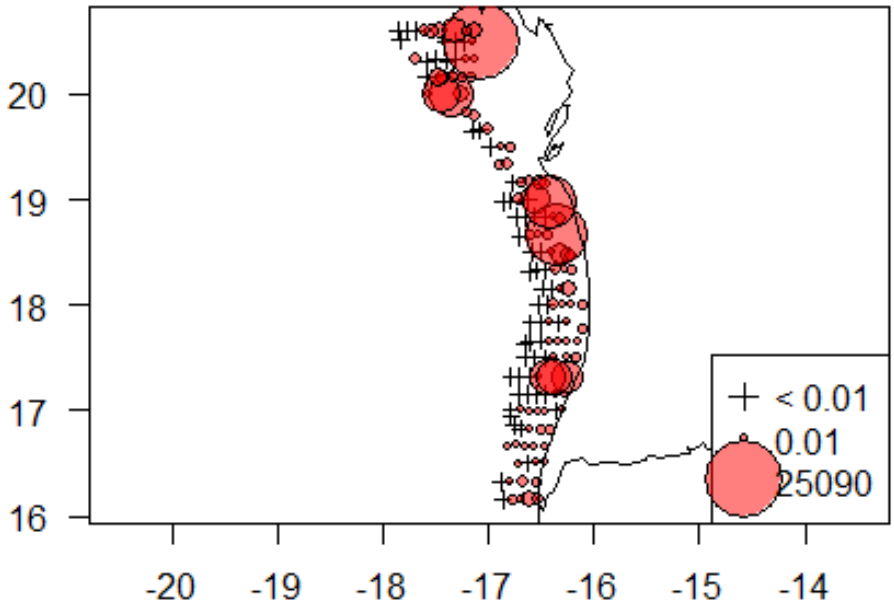


Figure A2.8. Proportional representation of the acoustic backscatter data attributed to pelagic fish along the Mauritian coast. Number of data: n = 651; mean: m = 1034.47; coefficient of variation: CV = 2.53; frequency of zeroes: f_0 = 0.437; maximum value: max = 25 090.

Annex 3: Demonstration Rscripts

Demonstration Rscripts that provide the capacity to undertake a geostatistical computation are presented here. They use case studies described in Annex 2, can be copy pasted in the R environment, and are ready to use. The reader is encouraged to understand the different steps of the computations and to customize the R scripts to his/her own data case studies. The table below contains a list of what can be achieved with the demonstration Rscripts available in the present annex.

Table A3.1. List of the demonstration Rscripts and related case studies available in this annex.

Demonstration Rscripts	Case studies
Computing spatial indices from survey data	Bay of Biscay hake (trawl data)
Computing variograms for a series of surveys	Gulf of Lion hake (trawl data)
Mapping by kriging with a variogram	Bay of Biscay anchovy (acoustic 2D data)
Global estimation and mapping by kriging with a transitive covariogram	Moroccan octopus (trawl data)
Global estimation with a variogram	Scottish herring egg (dredge data)
Global estimation in 1D for acoustic surveys	Bay of Biscay anchovy (acoustic 1D data)
Mapping by ordinary kriging, cokriging, colocated cokriging, and by kriging with an external drift	Scottish herring mean length (acoustic–trawl data)
Mapping by cokriging indicators with a linear model of coregionalization	Mauritian pelagic (acoustic data)
Exploring border effects among spatial sets of multiple indicators	Bay of Biscay anchovy (acoustic 2D data)
Mapping with the topcut (non-linear) model	Bay of Biscay anchovy (acoustic 2D data)
Conditional simulations	Scottish herring mean length (acoustic–trawl data)
Conditional simulations with the presence of zeros	Scottish herring acoustic backscatter (acoustic–trawl data)

A3.1 Computing spatial indices from survey data

```
#################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code computes spatial indices on hake densities from trawl su
rvey data
## in the bay of Biscay, France
## The data were supplied by Ifremer
##
## Author: M.Woillez, Ifremer
#################

# Load libraries
library(RGeostats)
```

```
library(mapdata)

# Load data
rg.load(filename="Demo.hake.bob.db.data",objname="db.data")
rg.load(filename="Demo.hake.bob.poly.data",objname="poly.data")

# Data management
db.data <- db.locate(db=db.data,names=1,loctype="rank")
db.data <- db.locate(db=db.data,names=2:3,loctype="x")
db.data <- db.locate(db=db.data,names=4,loctype="z")
db.data

# Visualizing the data set (proportional representation)
plot(db.data,title="Fish density samples",xlim=c(-11,0),ylim=c(43,49)
,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180),inches
=4,
     pos.legend=3,zmin=0,zmax=c(db.stat(db.data,"maxi")),include.boun
ds=FALSE)
plot(poly.data,col=8,add=T)
map("worldHires",add=T,fill=T,col=8); box()

# Use the following lines to draw and save a new polygon
#poly.data <- digitize(x="polygon")
#n
#polygon.write(polygon=poly.data,filename="poly.data.txt")
#poly.data <- polygon.read(filename="poly.data.txt")

# Define a simple projection based on the cosine of the mean latitude
projec.define(projection="mean", db=db.data)

# Compute areas of influence of survey samples
db.data <- db.delete(db=db.data,names=6)
db.data <- infl(db.data,nodes=c(400,400),origin=c(-11,43),extend=c(11
,6),
                dmax=100,polygon=poly.data,plot=T,asp=1)
db.data

# Compute and plot the inertia, the total abundance, the isotropy,
# the center of gravity and the coordinates of the axes of inertia
# and the isotropy.
# Note that intermediate results of the PCA decomposition are provide
d
# (the eigen values and the eigen vectors).
plot(db.data,title="Center of gravity and inertia of densities and sa
mples",asp=1,
     xlim=c(-300,150),ylim=c(-200,150),inches=5)
plot(poly.data,col=8,add=T)
SI.cgi(db.data,flag.plot=T,flag.inertia=T,col=2)

# Get the coordinates of the center of gravity in degrees
projec.invert(SI.cgi(db.data,flag.plot=F)$center[1],
              SI.cgi(db.data,flag.plot=F)$center[2])

# Compute and plot the inertia, the total abundance, the isotropy,
# the center of gravity and the coordinates of the axes of inertia
# and the isotropy of the samples
plot(db.add(db.data,S=1),add=TRUE,col=1,inches=5,pch="+")
SI.cgi(db.add(db.data,S=A0>=0),flag.plot=T,flag.inertia=T,col=1)
```

```
# Compute the global index of collocation between age 0 and age 1
SI.gic(db1=db.data,db2=db.data,name1="A0",name2="A1", col1="red",col2
="blue",
       flag.plot=T,flag.inertia=T,asp=1,inches=5,title="A0 and A1")
plot(poly.data,col=8,add=T)

# Compute the local index of collocation between age 0 and age 1
SI.lic(db.data,name1="A0",name2="A1")

# compute the microstructure index
SI.micro(db.data,h0=10,pol=poly.data,dlim=50,ndisc=400)

# compute abundance, positive area, equivalent area and spreading are
a
SI.stats(db.data,flag.plot=T)

# compute the number of spatial patches
SI.patches(db.data, D.min = 100, A.min = 10)
projec.toggle(0)
plot(poly.data,col=8,add=T)
title("Spatial patches")
```

A3.2 Computing variograms for a series of surveys

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code computes variograms on hake densities for a series of tr
awl surveys
## in the gulf of Lion, France. The data were supplied by Ifremer
##
## Author: N.Bez, IRD
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load libraries
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load(filename="Demo.hake.med.db.data", objname="db.data")

# Data presentation
plot(db.sel(db.data,YEAR==1996),zmin=0.001,pch.low=3,cex.low=0.25,las
=1,pch=21,col=1,     inches=5,title="Hake - 1996",asp=1)
map("worldHires",add=T)

# Define a simple projection based on the cosine of the mean latitude
projec.define(projection="mean",db=db.data)
```

```
# Evaluate the distance lag (in projected units i.e. n.mi.)
# and the number of lags by clicking on points
plot(db.sel(db.data,YEAR==1996),zmin=0.001,pch.low=3,cex.low=0.25,las
=1,pch=21,col=1,inches=5,title="Hake - 1996",asp=1)
worldHires <- map("worldHires",plot=F,xlim=c(3,5),ylim=c(42,44))
lines(projec.operate(worldHires $x,worldHires $y))

# distance lag
lag <- dist.digit()
lag <- signif(lag,2)
lag

# number of lag
diagonal <- dist.digit()
nlag <- ceiling((diagonal/2)/lag)
nlag

# Compute and represent omnidirectional variogram
vg.data <- vario.calc(db.sel(db.data,YEAR==1996), lag=lag,nlag=nlag)

# Edit the results
vg.data

# Plot the results
plot(vg.data,las=1,xlab="Distance (n.mi.)")
plot(vg.data,npairdw=T,inches=0.1,las=1,xlab="Distance (n.mi.)")

# Compute annual variograms and superimpose them
for(i in unique(db.data[,"YEAR"])){
  vg.data <- vario.calc(db.sel(db.data,YEAR==i), lag=lag,nlag=nlag)
  plot(vg.data,npairdw=T,inches=0.05,col=rgb(0,0,0,0.25),add=!(i==199
6),
       las=1,xlab="Distance (n.mi.)",ylim=c(0,1e+08))
}

# superimpose several omnidirectional standardized variograms
for(i in unique(db.data[,"YEAR"])){
  vg.data <- vario.calc(db.sel(db.data,YEAR==i), lag=lag,nlag=nlag)
  plot(vg.data,npairdw=T,inches=0.1,col=rgb(0,0,0,0.25),add=!(i==1996
),
       flag.norm=T,las=1,xlab="Distance (n.mi.)",ylim=c(0,2))
}

# Same computations but for the log-transformation of the hake densit
y
# This transformation can reduce the fluctuations and facilitate the
capture of
# a structure, however it does not allow to go from the structure of
the log to
# the structure of the raw variable.
for(i in unique(db.data[,"YEAR"])){
  vg.data <- vario.calc(db.sel(db.add(db.data,z1=log(1+MERLMER)),YEAR
==i),
                        lag=lag,nlag=nlag)
  plot(vg.data,npairdw=T,inches=0.1,col=rgb(0,0,0,0.25),add=!(i==1996
),
       flag.norm=T,las=1,xlab="Distance (n.mi.)",ylim=c(0,2))
}
```

```
# Standardized the density by the annual standard deviation and creat
e a new file.
# This should not be done with R function var() or sd() which compute
s
# the variance in (n-1)
# The YEAR is then attributed the locator "code" for selecting pairs
of points of
# similar years from the same year
db.data.std <- db.data
for(i in unique(db.data[,"YEAR"])){
  sel <- db.data.std[,3]==i
  sd.year <- sqrt(mean(db.data.std[,11][sel]^2) - mean(db.data.std[,1
1][sel])^2)
  db.data.std[,11][sel] <- db.data.std[,11][sel]/sd.year
}
db.data.std <- db.locate(db.data.std,3,"code")

# Compute annual variograms (which are normalized because of the stan
dardization of
# the density values)
for(i in unique(db.data[,"YEAR"])){
  vg.data <- vario.calc(db.sel(db.data.std,YEAR==i),lag=lag,nlag=nlag
)
  plot(vg.data,npairdw=T,inches=0.1,col=rgb(0,0,0,0.25),add=!(i==1996
),
       las=1,xlab="Distance (n.mi.)",ylim=c(0,2))
}

# Compute the mean annual variogram
# Pairs are retained if their codes are the same i.e. if their differ
ence is smaller # or equal than 0
vg.data.std <- vario.calc(db.data.std, lag=5,nlag=15,opt.code=1,tolco
de=0)
plot(vg.data.std,npairdw=T,inches=0.1,las=1,add=T,col=2,lwd=2)
```

A3.3 Mapping by kriging with a variogram

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code performs variography and mapping by kriging for a fisher
ies
## acoustic survey on anchovy in the bay of Biscay, France
## The data were supplied by Ifremer
##
## Author: P.Petitgas, Ifremer
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
```

```
projec.toggle(0)

# Load data
rg.load(filename="Demo.anchovy.bob.2d.db.data",objname="db.data")
rg.load(filename="Demo.anchovy.bob.2d.poly.data",objname="poly.data")

# Area limits
y1lim <- 43.3; y2lim <- 47; x1lim <- -4.5; x2lim <- -1

# Plot data
plot(db.data,name="ENGR.ENC",pch=1,asp=1.2,inches=5,col="black",
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim))
plot(poly.data, add=T, lty=1, density=0)
map("worldHires",add=T,fill=T,col=8)

#######
## Variography
#######

# Define projection
projec.define(projection="mean")

# Mask duplicates (points too close)
db.data <- duplicate(db.data)

# Calculate directional variogram
vg2 <- vario.calc(db.data,lag=c(2,15),dirvect=c(35,145), nlag=c(40,7)
)
plot(vg2,npairpt=0,npairdw=TRUE,title="",inches=.05)

# omni-directional variogram
vg <- vario.calc(db.data,lag=2,dirvect=NA, nlag=40)
plot(vg,npairpt=0,npairdw=TRUE,title="",inches=.05)

# fit isotropic variogram
vg.mod <- model.auto(vario=vg,struct=melem.name(c(1,3,3)))

#######
## Kriging
#######

# Grid for Kriging
x0 <- -4; y0 <- 43.4; dx <- 0.1;dy <- 0.1; nx <- 30; ny <- 37
db.grid <- db.create(x0=c(x0,y0),dx=c(dx,dy),nx=c(nx,ny))

# Select grid points inside polygon
db.grid <- db.polygon(db.grid,poly.data)

# plot data, grid and polygon
plot(db.grid, xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),pch=3, col="red
",asp=1.2,
     flag.proj=FALSE)
plot(db.data,pch=20,add=T,col="black",inches=3, flag.proj=FALSE)
map("worldHires",add=T,fill=T,col=8); box()

# neighbourhood
neimov <- neigh.create(ndim=2,type=2,nmini=3,nmaxi=10,radius=25)

# Kriging
```

```
kres <- kriging(dbin=db.data,dbout=db.grid, model=vg.mod, neigh=neimo
v)

# plot kriging results: K.estim
plot(kres,name.image="z1",title="K.estim",col=topo.colors(20), asp=1.
2,
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),pos.legend=5,flag.proj=F
ALSE)
plot(db.data,pch=20,add=T,col="red",inches=3,flag.proj=FALSE)
map("worldHires",add=T,fill=T,col=8)

# plot kriging results: K.std
plot(kres,name.image="z2",title="K.std",col=topo.colors(20), asp=1.2,
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),pos.legend=5,flag.proj=F
ALSE)
plot(db.data,pch=20,add=T,col="black",inches=1.5,flag.proj=FALSE)
map("worldHires",add=T,fill=T,col=8)

# ratio of means kriged.map/data
mean(kres[db.grid[,"sel"],"z1"])/mean(db.data[,"z1"])
```

A3.4 Global estimation and mapping by kriging with a transitive covariogram

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code performs global and local estimation using transitive ge
ostatistics
## on octopus data from trawl survey off Morocco. The data are suppli
ed
## by Abdelmalek Faraj, Institut National de Recherche Halieutique
## (http://www.inrh.ma/), Casablanca, Morocco.
##
## Author: N. Bez, IRD
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load("Demo.octopus.morocco.db.data","db.data")
rg.load("Demo.octopus.morocco.poly.data","poly.data")

# Dimension of a regular strata in n.mi.
strata = 11

# Defining a projection
projec.define("mean",db=db.data)
```

```
# Surface of influence for inner data
# in square nautical miles
db.data <- infl(db.data,nodes=400,extend=c(6,6),origin=c(-18,20.5),
                polygon=poly.data,plot=T,asp=0.8,
                title = "Influence Polygons")

# Covariogram computation
# Lag = Regular strata
lag     <- strata
nlag    <- 20
dirvect <- 50 + c(0,90)
covario.data <- vario.calc(db.data,breaks=seq(strata/2,nlag*strata,by
=strata),calcul="covg",tolang=45,dirvect=dirvect)

# Calculate relative g(h) scaled by
# Q (total abundance)
Q <- sum(db.data[,"JUV"]*db.data[,"Influence.Surface"])
covario.data = vario.transfo("v1/Q^2",covario.data)
variance = covario.data$vars

# Ajust the empirical covariogram with
# variance constraints
model.covario <- model.auto(covario.data,struct=c(1,3,3),constraints=
variance,
                            npairdw=1,las=1,inches=0.05,lwd=2,xlim=c(
0,250),
                            title = "Relative Geometrical Covariogram
")

# Estimation variance
projec.toggle(0,verbose=FALSE)
CVtrue <- sqrt(strata^2*(variance -
                           model.cvv(v.mesh=strata,model=model.covari
o,
                                     seed=110366,ndisc=20)))
cat(paste("Abundance estimate = ",round(Q/10^6,0)," e+06",sep=""),"\n
")
cat(paste("Estimation Coefficient of Variation = ",
          round(100*CVtrue,1),"%",sep=""),"\n")

# Grid definition in geographical space
projec.toggle(0,verbose=FALSE)
grid.kri <- db.grid.init(db.data,nodes=400,extend=c(6,6),origin=c(-18
,20.5))
grid.kri <- db.polygon(grid.kri,poly.data)
projec.toggle(1,verbose=FALSE)

# Neighborhood definition
neigh.kri <- neigh.create(ndim=2,type=2,flag.aniso=TRUE,flag.rotation
=TRUE,
                          nmini=10,nmaxi=30,radius=c(150,50),
                          rotmat=util.ang2mat(ndim=2,angles=50))

# Transitive kriging
res <- kriging(db.data,grid.kri,model.covario,neigh.kri)

# Threshold negative estimates
res[,"Kriging.JUV.estim"][res[,"Kriging.JUV.estim"]<0] <- 0
```

```
# Map of the result
plot(res,col=rainbow(6,start=0.2,end=1),las=1,
     title="Octopus density - 1999",asp=0.8,flag.proj=FALSE)
plot(poly.data,las=1,add=T,flag.proj=FALSE)
map("worldHires", fill=T,col=grey(0.8),add=T)
plot(db.data,las=1,add=T,col=1)
legend.image(range(db.extract(res,"Kriging.JUV.estim"),na.rm=T),
             position="bottomright",col=rainbow(6,start=0.2,end=1),
             ntdec=0,cex=0.75)
```

A3.5 Global estimation with a variogram and precision of alternative survey designs

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code performs Global estimation over a domain (polygon)

## It computes also the survey precision for alternative survey desig
ns
## It uses herring eggs densitites from a dredge survey.
## The data were supplied by Marine Scotland Science
## at the Marine Laboratory, Aberdeen, UK.
##
## Author: P.Petitgas, Ifremer
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load("Demo.herreggs.scot.db.data","db.data")
rg.load("Demo.herreggs.scot.poly.data","poly.data")

# limits of study area : limits of plots
x1lim<-25.9; x2lim<-26.5; y1lim<-17.0; y2lim<-17.58

# select points inside polygon
db.data <- db.polygon(db.data,poly.data,verbose=TRUE)

# plot data and polygon
plot(db.data,name.prop="z1",xlab="",ylab="",
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim))
plot(poly.data,add=T,lty=1,density=0)

# Data mean and variance inside polygon
zm <- mean(db.data[,4][db.data[,5]])
zv <- var(db.data[,4][db.data[,5]])*(sum(db.data[,5])-1)/sum(db.data[
,5])
cat("Data\n")
```

```
cat("mean: ",zm,"    std: ",sqrt(zv),"   cv: ",sqrt(zv)/zm,"\n")

######
### Calculate experimental Variogram
######

cat("Defining the model...\n")

# Calculate experimental variogram
Lag <- 0.05; Nlag <- 9
vg <- vario.calc(db.data,dirvect=0,lag=Lag,toldis=0.5, nlag=Nlag, bre
aks = NA,
                 calcul="vg",by.sample=FALSE,opt.code=0, tolcode=0, m
eans=NA)

# Plot experimental variogram
vario.plot(vg,npairdw=F,xlab="Distance (km)",ylab="Variogram")

# Fit a variogram model
vg.init <- model.create("Nugget Effect",sill=50000,ndim=2)
vg.init <- model.create("Exponential",range=0.15,sill=370000,model=vg
.init)
vg.fit  <- model.fit(vg, vg.init, niter=100, wmode=3, draw=T,
                     npairdw=F,xlab="Distance (km)",ylab="Variogram")

#######
### Global estimation variance 2D
### Estimate = zone mean (over polygon)
#######

# Define the discretization grid and select those points inside the p
olygon
gnx <- 100; gny <- 100;
gd.disc <- db.grid.init(obj=poly.data,nodes=c(gnx,gny))
gd.disc <- db.polygon(gd.disc,poly.data)
plot(gd.disc,pch=3,col=1);plot(db.data,add=T,pch=21)
plot(poly.data,add=T)

# Global estimate = arithmetic mean
cat("Estimating the Global estimate by arithmetic mean...\n")
global.ma <- global(dbin=db.data, dbout=gd.disc, model = vg.fit, uc=c
("1"),
                    polygon = poly.data, calcul = "arith", verbose=0)

# Global estimate = kriged mean
cat("Estimating the Global Estimate by Kriging...\n")
global.mk <- global(dbin=db.data, dbout=gd.disc, model = vg.fit, uc=c
("1"),
                    polygon = poly.data, calcul = "krige", flag.wgt=T
RUE,
                    verbose=0)

# Display kriging weights
db.data <- db.add(db.data,global.mk$wgt,loctype="w")
plot(db.data,name.prop="w");
plot(poly.data,add=T)

# Theoretical process mean
cat("Estimating the Process Mean...\n")
```

```
global.mt <- global(dbin=db.data, dbout=gd.disc, model = vg.fit, uc=c
("1"),
                    polygon = poly.data, calcul = "mean", flag.wgt=TR
UE,
                    verbose=0)

# Summary of results
cat("\nGlobal estimation over the Polygon\n")
tab1 <- rbind(c(global.ma$zest,global.ma$cv),
              c(global.mk$zest,global.mk$cv),
              c(global.mt$zest,global.mt$cv) )
dimnames(tab1) <- list(c("arith mean","kriged zone mean","kriged proc
ess mean"),
                       c("Estimate","CV"))
print(round(tab1,3))

#######
###  Testing alternative sampling designs: regular and purely random
#######

# Regular grid design : create
x0 <- 25.9; y0 <- 17.0
dx <- 0.06; dy <- 0.06
nx <- 13;   ny <- 12
db.nw <- db.create(x0=c(x0,y0),dx=c(dx,dy),nx=c(nx,ny))
db.nw <- db.add(db.nw,loctype="z")
db.nw <- db.locate(db.nw,2:3,loctype="x")
db.nw <- db.add(db.nw,z1=rep(zm,db.nw$nech))
db.nw <- db.locate(db.nw,4,loctype="z");
db.nw <- db.polygon(db.nw,poly.data)

# Regular grid design : display
plot(db.nw,pch=3,xlab="km",ylab="km",flag.aspoint=TRUE,name.post=1,
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim))
plot(poly.data,add=T,lty=1,density=0)

# Regular grid design : global estimation
cat("Testing the Regular Design...\n")
global.syst <-global(dbin=db.nw, dbout=gd.disc, model = vg.fit, uc=c(
"1"),
                     polygon = poly.data, calcul = "arith", verbose=0
)

# Summary of results
cat("\nTesting alternative sampling designs:\n")
tab2 <- rbind(c(global.ma$zest,global.ma$cv,sum(db.data[,5])),
              c(zm,global.syst$sse/zm,sum(db.nw[,5])),
              c(zm,sqrt(zv/sum(db.data[,5]))/zm,sum(db.data[,5])) )
dimnames(tab2) <- list(c("Data","Regular","Random"),c("Mean","CV","NB
"))
print(round(tab2,3))
```

A3.6 Global estimation in 1D for acoustic surveys and precision for different sampling efforts

```
#################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
```

```
cology
##
## This code performs global estimation in 1D using acoustic transect
sums
## The approach applies to acoustic surveys with regularly spaced par
allel
## transects
## The code uses the 1-d data files where fish biomass per transect
## was obtained by summing the densities along the transects. This fi
le was
## constructed from the 2-d data set corresponding to an acoustic sur
vey in the bay
## of Biscay. Data are supplied by Ifremer.
##
## Author: P.Petitgas, Ifremer
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load("Demo.anchovy.bob.1d.db.data","db.data")
nrad <-db.data$nsamples                  # Nb of transects
aa <- 1                                  # Inter-transect (arbitrary)
distance

# Transform db.data into regular grid
db.datagrid <- db.grid.init(db.data,nodes=nrad,flag.regular=T)
db.datagrid <- migrate(db.data,db.datagrid,flag.fill=2,name="Tr.bioma
ss")

# Display information
plot(db.data,pch=20,type="b",title="Biomass",
     xlab="S <---- Transects ----> N", ylab="Biomass per transect")
plot(db.datagrid,add=TRUE,col="red")

#######
## Estimation globale 1d : intrinsic method + geometric error
#######

# Experimental variogram 1D
vg <- vario.calc(db.data,calcul="vg")

# Semi-automatic fit (sills only)
vg.init <- model.create(vartype="Nugget Effect",sill=1.5E+05,ndim=1)
vg.init <- model.create(vartype="Spherical",range=4.5,
                        model=vg.init)
vg.fit  <- model.fit(vg, vg.init, niter=100, wmode=3,draw=FALSE)

# Automatic fit
vg.auto <- model.auto(vg,struc=c("Nugget Effect","Spherical"),
                      xlab="Distance", ylab="variogram",draw=FALSE)
```

```
# Variogram and model representations
plot(vg,npairpt=0,npairdw=T,xlab="Distance", ylab="variogram",inches=
.025)
plot(vg.fit ,add=TRUE,col="blue")
plot(vg.auto,add=TRUE,col="red")

# Choose model
vgmod <- vg.auto
perc  <- vgmod[1]$sill/(vgmod[1]$sill + vgmod[2]$sill)
cat("Percent of nugget in total sill =",perc,"\n")

# Estimation variance 1D
gloa    <- global(db.datagrid,calcul="arith",model=vgmod,ndisc=100,ve
rbose=0)
s2      <- var(db.data[,"Tr.biomass"],na.rm=T)*(nrad-1)/nrad
d2geom  <- s2*(aa^2/6)/(aa*nrad)^2
cv.geom <- sqrt(d2geom)/gloa$zest            # Geometric error (limit
s of 1D)
cv.tot  <- sqrt(gloa$sse^2+d2geom)/gloa$zest # Total error CV

cat("Mean=",round(gloa$zest,3), "CVest=",round(gloa$sse/gloa$zest,3),
"\n")
cat("CV.est=",round(gloa$cv,3)," CV.geom=",round(cv.geom,3),"\n")
cat("Qtot=",gloa$zest*nrad*aa,"   CV=",round(cv.tot,3),"\n")

#######
## Alternative sampling effort:
## Estimation variance for other ('nk') inter-transect distances
## (geometric error neglected)
#######

nk <- 9

# Loop on alternative inter-transect distances
sse <- numeric(nk)
for (k in 1:nk) {
  ak  <- k*0.25*aa                      # new inter-transect distance
  nrk <- round(nrad*aa/ak,0)            # new nb of transects
  d2geom <- s2*(ak^2/6)/(ak*nrk)^2      # variance geometric error

  # variance estimation error
  db.datagrid <- db.grid.init(db.data,nodes=nrk,flag.regular=T)
  db.datagrid <- migrate(db.data,db.datagrid,flag.fill=2,name="Tr.bio
mass")
  d2estim     <- global(db.datagrid,calcul="arith",model=vgmod,ndisc=
100,
                        verbose=0)$sse^2
  sse[k] <- sqrt(d2estim+d2geom)
}

plot(0.25*aa*(1:nk),sse/gloa$zest,type="b",
     xlab="Multiplyer of Inter-transect Distance",ylab="Estimation CV
")
```

A3.7 Mapping by ordinary kriging, by cokriging, by collocated cokriging, and by kriging with an external drift

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code performs ordinary kriging, co-kriging, co-located co-kri
ging
## and kriging with external drift using herring mean length and bott
om
## depth data from an acoustic-trawl survey around the Shetland.
## The data were supplied by Marine Scotland Science
## at the Marine Laboratory, Aberdeen, UK.
##
## Author: M.Woillez, Ifremer
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load(filename="Demo.herring.len.scot.db.data",objname="db.data")
rg.load(filename="Demo.herring.len.scot.poly.data",objname="poly.data
")
rg.load(filename="Demo.herring.len.scot.grid.kriging",objname="grid.k
riging")

# Select points inside polygon
db.data <- db.polygon(db.data,poly.data,verbose=TRUE)

# Visualizing the data set
plot(poly.data,col=4,asp=1/cos(mean(db.extract(db=db.data,names="x2")
)*pi/180))
plot(grid.kriging,name.contour="depth",levels=seq(100,1500,100),col=8
,add=T)
plot(db.data,inches=4,pos.legend=5,add=T)
title(main="mean length")
map("worldHires",add=T,fill=T,col=8); box()

# Define the projection
projec.define(projection="mean", db=db.data)

######
### Ordinary kriging
######

# Basic statistics
print(db.data,flag.stats=TRUE,names="m.length")
hist(db.extract(db.data,"z1"),xlab="m.length",main="",col=8);box()

# Calculate experimental omni-directional variogram
```

```
vario.data <- vario.calc(db.data)
plot(vario.data,npairpt=F,npairdw=T,inches=0.08)

# Model it with model.auto(): automatic fitting procedure for range a
nd sill
model.vario <- model.auto(vario=vario.data,struc=c("Nugget Effect","G
aussian"),
                          wmode=2)

# Define the neighborhood
neigh.kriging <- neigh.create(ndim=2,type=0)
neigh.kriging

# View grid
plot(grid.kriging,col=1,title="",pch="+",asp=1)
plot(db.data,col=2,pch=19,add=T)
plot(poly.data,col=4,add=T)

# Perform ordinary kriging
grid.kriging <- kriging(dbin=db.data,dbout=grid.kriging,model=model.v
ario,
                        neigh=neigh.kriging,uc=c("1"),mean=NA,
                        calcul="point",radix="OK")

# View ordinary kriging
# local estimation
plot(grid.kriging,name.image="z1",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
plot(db.data,name.prop="z1",col=1,pch=20,add=T,flag.proj=F)
map("worldHires",add=T,fill=T,col=8); box()

# Kriging variance
plot(grid.kriging,name.image="z2",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
map("worldHires",add=T,fill=T,col=8); box()

######
### Co-kriging
######

# Data management
db.data <- db.locate(db.data,5:6,loctype="z")
db.data

# Basic statistics
print(db.data,flag.stats=TRUE,names=c("depth","m.length"))
hist(db.extract(db.data,"z1"),xlab="depth",main="",col=8);box()
hist(db.extract(db.data,"z2"),xlab="m.length",main="",col=8);box()
correlation(db.data,name1="depth",name2="m.length")

# Calculate experimental omni-directional variogram
vario.data <- vario.calc(db.data)
plot(vario.data,npairpt=F,npairdw=T,inches=0.08)

# Model it with model.auto(): automatic fitting procedure for range,
and sill
model.vario <- model.auto(vario=vario.data,wmode=2,flag.goulard=TRUE,
                          struc=c("Nugget Effect","Gaussian","Gaussia
n"))
```

```
model.vario

# Define grid
grid.kriging <- db.locate(grid.kriging,6:7,loctype=NA)
grid.kriging

# View grid
plot(grid.kriging,col=1,title="",pch="+",asp=1)
plot(db.data,col=2,pch=19,add=T)
plot(poly.data,col=4,add=T)

# Perform ordinary cokriging
grid.kriging <- kriging(dbin=db.data,dbout=grid.kriging,model=model.v
ario,
                        neigh=neigh.kriging,uc=c("1"),mean=NA,calcul=
"point",
                        radix="CK")

# View cokriging
# Local estimation
plot(grid.kriging,name.image="z1",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
plot(db.data,name.prop="z1",col=1,pch=20,add=T,flag.proj=F)
map("worldHires",add=T,fill=T,col=8); box()

# Local estimation
plot(grid.kriging,name.image="z2",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
plot(db.data,name.prop="z2",col=1,pch=20,add=T,flag.proj=F)
map("worldHires",add=T,fill=T,col=8); box()

# Kriging variance
plot(grid.kriging,name.image="z3",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
map("worldHires",add=T,fill=T,col=8); box()

# Kriging variance
plot(grid.kriging,name.image="z4",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
map("worldHires",add=T,fill=T,col=8); box()

######
### Collocated co-kriging
######

# Define grid
grid.kriging <- db.locate(grid.kriging,8:11,loctype=NA)
grid.kriging <- db.locate(grid.kriging,4,loctype="z")
grid.kriging

# View grid
plot(grid.kriging,name.prop="z1",col=1,title="",pch="+",asp=1)
plot(db.data,col=2,pch=19,add=T)
plot(poly.data,col=4,add=T)

# Perform colocated cokriging
grid.kriging <- kriging(dbin=db.data,dbout=grid.kriging,model=model.v
ario,
                        neigh=neigh.kriging,uc=c("1"),mean=NA,calcul=
```

```
"point",
                          radix="CCK",rank.colcok=c(4,NA))

# View colocated cokriging
# Local estimation
plot(grid.kriging,name.image="z2",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
plot(db.data,name.prop="z2",col=1,pch=20,add=T,flag.proj=F)
map("worldHires",add=T,fill=T,col=8); box()

# Kriging variance
plot(grid.kriging,name.image="z4",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
map("worldHires",add=T,fill=T,col=8); box()

######
### Kriging with external drift
######

# Regression
plot(db.extract(db.data,"depth"),db.extract(db.data,"m.length"),main=
"regression",
     pch=19,xlim=c(50,250),ylim=c(15,35),xlab="depth",ylab="mean leng
th")
db.data <- regression(db.data,names="depth",name1=6,flag.draw=T)

# Data management
db.data <- db.locate(db.data,6,loctype=NA)
db.data

# Calculate multivariate experimental omni-directional variogram
vario.data <- vario.calc(db.data)
plot(vario.data,npairpt=F,npairdw=T,inches=0.08)

# Model it with model.auto(): automatic fitting procedure for range a
nd sill
model.vario <- model.auto(vario=vario.data,struc=c("Nugget Effect","G
aussian"),
                          wmode=2)
model.vario

# Data management
db.data <- db.locate(db.data,8,loctype=NA)
db.data <- db.locate(db.data,6,loctype="z")
db.data <- db.locate(db.data,5,loctype="f")
db.data

# Define grid
grid.kriging <- db.locate(grid.kriging,12:15,NA)
grid.kriging <- db.locate(grid.kriging,"depth","f")
grid.kriging

# View grid
plot(grid.kriging,col=1,title="",pch="+",asp=1)
plot(db.data,col=2,pch=19,add=T)
plot(poly.data,col=4,add=T)

# View drift
plot(grid.kriging,name.image="depth",title="",asp=1)
```

```
plot(poly.data,col=4,add=T)

# Perform kriging with external drift
grid.kriging <- kriging(dbin=db.data,dbout=grid.kriging,model=model.v
ario,
                        neigh=neigh.kriging,uc=c("1","f1"),mean=NA,
                        calcul="point",radix="KED")
grid.kriging

# View kriging with external drift
# Local estimation
plot(grid.kriging,name.image="z1",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
plot(db.data,name.prop="z1",col=1,pch=20,add=T,flag.proj=F)
map("worldHires",add=T,fill=T,col=8); box()

# Kriging variance
plot(grid.kriging,name.image="z2",pos.legend=5,flag.proj=F,
     asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180))
map("worldHires",add=T,fill=T,col=8); box()
```

A3.8 Mapping by cokriging indicators with a linear model of coregionalization

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code performs indicator co-kriging using acoustic data
## The data are supplied by Cheikh-Baye Braham,
## Institut Mauritanien de Recherche Oceanographique et des Peches (I
MROP),
## Nouadhibou, Mauritania
## http://www.imrop.mr/
##
## Author: N.Bez, Ifremer
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load("Demo.acoustic.maur.db.data","db.data")
rg.load("Demo.acoustic.maur.poly.data","poly.data")

# Create grid
grid.kri <- db.grid.init(poly.data,margin=10,nodes=150)
grid.kri <- db.polygon(grid.kri,poly.data)

# Define a projection
projec.define("mean",db=db.data)
```

```
# Display the data
plot(db.data,asp=1,flag.proj=FALSE)
plot(poly.data,add=T,flag.proj=FALSE)
map("worldHires",add=T)

# Define cutoffs and build indicators
zcut <- as.numeric(quantile(db.data [,5][db.data[,5] >0]))
my.limits <- limits.create(zcut=zcut[-5],flag.zcut.int = F)
db.data <- db.indicator(db.data ,my.limits)

# Variograhy
lag     <- c(5,10)
nlag    <- 15
dirvect <- c(0,90)

# Annual variograms
for(i in 1:4){
  plot(vario.calc(db.sel(db.data,an==i),lag=lag,
                  nlag=nlag,dirvect=dirvect),
       flag.norm=T,add=(i!= 1))
}

# Mean annual variogram
vario.data <- vario.calc(db.data,lag=lag,nlag=nlag,dirvect=dirvect,
                         opt.code=1,tolcode=0)

# Model using structures that are linear
# at origin (spherical or exponential)
model.vario <- model.auto(vario.data,struct=c(1,3,3,2,2),
                          wmode=2,npairdw=1)

# Co-Kriging
neigh.kri <- neigh.create(type=2,ndim=2,nmini=10,nmaxi=50,radius=60)
kri.1 <- kriging(db.sel(db.data,an==1),grid.kri,model.vario,neigh.kri
)

# Truncating estimations within [0,1]
ranks = db.ident(kri.1,names="Kriging.Indicator*")
for(i in ranks){
  kri.1[,i][kri.1[,i] < 0] <- 0
  kri.1[,i][kri.1[,i] > 1] <- 1
}

# Mapping the results
plot(kri.1,asp=1,zlim=c(0,1),col=rainbow(4,start=0.2,end=1),flag.proj
=FALSE,
     name="Kriging.Indicator.Total.1.estim",title="First Indicator")
legend.image(c(0,1),position="bottomleft",col=rainbow(4,start=0.2,end
=1),
             ntdec=2,cex=0.75)
map("worldHires",add=T)
plot(db.data,col="black",cex1=0.1,add=T,flag.proj=FALSE)

plot(kri.1,asp=1,zlim=c(0,1),col=rainbow(4,start=0.2,end=1),flag.proj
=FALSE,
     name="Kriging.Indicator.Total.4.estim",title="Fourth Indicator")
legend.image(c(0,1),position="bottomleft",col=rainbow(4,start=0.2,end
=1),
```

```
             ntdec=2,cex=0.75)
map("worldHires",add=T)
plot(db.data,col="black",cex1=0.1,add=T,flag.proj=FALSE)

# Building indicators for intervals
res <- kri.1
natt.first = res$natt + 1
res <- db.add(res, zero   = (1-res[,ranks[1]]))
res <- db.add(res, low    = (res[,ranks[1]]-res[,ranks[2]]))
res <- db.add(res, medium = (res[,ranks[2]]-res[,ranks[3]]))
res <- db.add(res, large  = (res[,ranks[3]]-res[,ranks[4]]))
res <- db.add(res, extrem = (res[,ranks[4]]))

# Getting rank of most probable interval
res <- db.compare(res,fun="maxi",names=natt.first:res$natt)
res <- db.add(res, class=rep(NA,res@nx[1]*res@nx[2]))
for(i in 1:5)
  res[,"class"][res[,natt.first+i-1]==res[,"maxi"]] <- i

plot(res,asp=1,col=rainbow(5,start=0.2,end=1),flag.proj=FALSE)
legend.image(c(1,5),position="bottomleft",col=rainbow(5,start=0.2,end
=1),
             ntdec=0,cex=0.75)
map("worldHires",add=T)
plot(db.data,col="black",cex1=0.1,add=T,flag.proj=FALSE)
```

A3.9 Exploring border effects among spatial sets of multiple indicators

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code explores border effects among spatial nested sets define
d by indicators
## The methodology is applied to a fisheries acoustic survey on ancho
vy in
## the bay of Biscay, France
## The data were supplied by Ifremer
##
## Author: M.Woillez, Ifremer
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load("Demo.anchovy.bob.2d.db.data","db.data")
rg.load("Demo.anchovy.bob.2d.poly.data","poly.data")

# Select points inside polygon
```

```
db.data <- db.polygon(db.data,poly.data,verbose=TRUE)

# Check if duplicates (points too close)
db.data <- duplicate(db.data)

# Basic statistics
print(db.data,flag.stats=TRUE,name="ENGR.ENC")

# Histogram
hist(db.extract(db.data,"ENGR.ENC"),col=8,main="",nclass=100,
     xlab="biomass of anchovy (tonnes/n.mi.²)");box()

# Visualization
plot(poly.data,asp=1/cos(mean(db.extract(db.data,"x2"))*pi/180),col=4
)
plot(db.data,inches=5,add=T,pos.legend=5,zmin=0,zmax=c(db.stat(db.dat
a,"maxi")),
     include.bounds=FALSE)
plot(poly.data,col=4,add=T)
map("worldHires",add=T,fill=T,col=8);box()

# Define a simple projection
projec.define(projection="mean",db=db.data)

# Create indicator variables into the RGeostats database
zi <- 150 # Topcut value
zcut <- c(quantile(db.data[,"ENGR.ENC"][db.data[,"ENGR.ENC"]!=0],seq(
0,0.8,0.2)),zi)
my.limits <- limits.create(zcut=zcut,flag.zcut.int = F)
db.data <- db.indicator(db.data ,my.limits)

# Visualization
for(i in 1:length(zcut)){
  plot(db.locate(db.data,paste("Indicator.ENGR.ENC.",i,sep=""),"z"),
       asp=1/cos(mean(db.extract(db=db.data,names="x2"))*pi/180),
       inches=1,bg=2,pch=21,flag.proj=F)
  plot(poly.data,col=4,flag.proj=F,add=T)
  map("worldHires",add=T,fill=T,col=8);box()
}

# Statistics
rbind("zcut"=zcut,"P(Z>=zcut)"=round(apply(db.data[,7:db.data$natt],2
,mean),3))

# Compute simple and cross variograms of indicator variables
lag <- 5; nlag <- 20; dirvect <- 0
vg <- vario.calc(db.data,lag=lag,nlag=nlag,dirvect=dirvect)

# Visualization of simple variograms
flag <- F
for(i in 1:length(zcut)){
  plot(vg,varcols=i,inches=.05,flag.norm=T,npairdw=T,npairpt=F,
       col=grey(seq(0,1,.1))[i],ylim=c(0,2),
       main="normed indicator variograms",
       xlab="Distance (n.mi.)",add=flag)
  flag <- T
}

# Visualization of simple and cross variograms
```

```
plot(vg,maxnvar=length(zcut),inches=.03,flag.norm=T,npairdw=T,
     npairpt=F,ylim=c(0,2))

# Compute variogram ratio: cross variogram / first simple variogram
vgr <- vario.transfo("v1",vario1=vg,oper="g12/g1")

# Visualization of normed variogram ratios
plot(vgr,maxnvar=length(zcut),inches=0.03,flag.norm=T,
     npairdw=T,npairpt=F,ylim=c(0,2))

# Visualization of normed variogram ratios
# grouped relatively to the first simple variograms used in the ratio
for(i in 1:(length(zcut)-1)){
  flag <- F
  for(j in i+1:length(zcut)){
    plot(vgr,varcols=i,varcols2=j,inches=0.05,flag.norm=T,
         npairdw=T,npairpt=F,col=grey(seq(0,1,.1))[j],ylim=c(0,2),
         main=paste("normed variograms ratio relative to I",i,sep="")
,
         add=flag)
    flag <- T
  }
}
```

A3.10 Mapping with the topcut (non-linear) model

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code fits a topcut model and performs kriging with it
## Here the Topcut model reduces to co-kriging the truncated variable
with
## the indicator of the topcut cutoff
## The interest in the methodology is twofold:
##  1/ because of the indicator approach, variography is more robust
##     with respect to high values
##  2/ because of co-kriging, high values are estimated where the pro
bability
##     is high for them to occur
## The methodology is applied to a fisheries acoustic survey on ancho
vy in
## the bay of Biscay, France
## The data were supplied by Ifremer
##
## Author: P.Petitgas, Ifremer
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)
```

```
# Load data
rg.load("Demo.anchovy.bob.2d.db.data","db.data")
rg.load("Demo.anchovy.bob.2d.poly.data","poly.data")

# Area limits
y1lim <- 43.3; y2lim <- 47; x1lim <- -4.5; x2lim <- -1

# Plot data
plot(db.data,name="ENGR.ENC",pch=1,asp=1.2,inches=5,col="black",
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim))
plot(poly.data, add=T, lty=1, density=0)
map("worldHires",add=T,fill=T,col=8)

# Topcut value: it has been chosen after structural analysis of bor-
der effects
zi=150

# new variables in db.data: topcut indicator (I)
#                           truncated variable (z1), excess variables
(z2)
#                           residuals around mean excess (z3)
# note: Z3 shows : no spatial structure (nugget variogram) ;
#                  no spatial cross-correlation with z1 nor z2

# Topcut indicator (I)
db.data <- db.add(db.data,I=(ENGR.ENC >= zi) * 1)
p <- mean(db.data[,"I"])
v <- p * (1-p)

# Truncated variable (z1)
db.data <- db.add(db.data,z1=ifelse(ENGR.ENC>=zi,zi,ENGR.ENC),
                  type.locate=FALSE)
mi <- mean(db.data[db.data[,"ENGR.ENC"]>=zi,"ENGR.ENC"])

# Mean excess (z2)
db.data <- db.add(db.data,z2=(mi-zi)*I,type.locate=FALSE)

# Residual around mean excess (z3)
db.data <- db.add(db.data,z3=(ENGR.ENC-mi)*I,type.locate=FALSE)

# Note: Residual is not considered in the structural analysis
# as its structure is a pure nugget effect
db.data <- db.locate(db.data,db.data$natt)

cat("Truncation at",zi,"\n")
cat("Indicator: mean=",round(p,4),"variance=",round(v,4),"\n")
cat("Mean above truncation=",mi,"\n")

#######
## Variography
#######
```

```
# Projection
projec.define(projection="mean")

# Look for duplicates (points too close)
db.data <- duplicate(db.data)

# Omni-directional variogram
vg <- vario.calc(db.data,lag=2,dirvect=NA, nlag=40)
plot(vg,npairpt=0,npairdw=F,title="",inches=.05)

# Fit variogram model
vg.init <- model.create(vartype="Nugget Effect",ndim=2,nvar=2)
vg.init <- model.create(vartype="Spherical",range=8,model=vg.init)
vg.init <- model.create(vartype="Spherical",range=25,model=vg.init)

# Autimatic fit of sills only
vg.fit  <- model.fit(vg, vg.init, niter=100, wmode=3, draw=F)

# Automatic fit of ranges and sills
vg.auto <- model.auto(vario=vg,struct=melem.name(c(1,3,3)),draw=F)

# Overlay models and variogram
plot(vg,npairdw=F,npairpt=F)
plot(vg.fit,vario=vg,lwd=2,add=T)
plot(vg.auto,vario=vg,lwd=2,add=T,col="blue")

# Choose a model
vg.mod <- vg.fit

#######
## Co-Kriging z1,z2
#######

# Define the Estimation Grid
x0 <- -4; y0 <- 43.4; dx <- 0.1;dy <- 0.1; nx <- 30; ny <- 37
db.grid <- db.create(flag.grid=T,x0=c(x0,y0),dx=c(dx,dy),nx=c(nx,ny))

# Select grid points inside polygon
db.grid <- db.polygon(db.grid,poly.data)

# Define a Moving Neighbourhood
neimov <- neigh.create(ndim=2,type=2,nmini=3,nmaxi=10,radius=25)

# Co-kriging (point)
kres2 <- kriging(dbin=db.data,dbout=db.grid,model=vg.mod,
neigh=neimov,
                 radix="K")
```

```
# Add co-kriged z1 and z2 estimates
kres2 <- db.add(kres2,K.topcut.estim=K.z1.estim+K.z2.estim)

# Display estimated Topcut
plot(kres2,name="K.topcut.estim",title="Estimated Topcut",
     col=topo.colors(20), asp=1.2,
xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),
     pos.legend=5,flag.proj=FALSE)
plot(db.data,name.prop="ENGR.ENC",pch=20,add=T,col="red",inches=3)
map("worldHires",add=T,fill=T,col=8)

#######
## Ordinary kriging of original variable (for comparison)
#######

# Change locators in db.data
db.data <- db.locerase(db.data,"z")
db.data <- db.locate(db.data,names="ENGR.ENC",loctype="z")

# Omni-directional variogram
vg1 <- vario.calc(db.data,lag=2,dirvect=NA, nlag=40)
vg1.mod <-
model.auto(vg1,struct=melem.name(c(1,3)),npairdw=TRUE,inches=0.05)

# Kriging (point)
kres1 <-
kriging(dbin=db.data,dbout=db.grid,model=vg1.mod,neigh=neimov,
                  radix="K")

# Display Ordinary Kriging results
plot(kres1,name.image="K.ENGR.ENC.estim",title="Ordinary Kriging re-
sult",
     col=topo.colors(20), asp=1.2,
     xlim=c(x1lim,x2lim),ylim=c(y1lim,y2lim),pos.leg-
end=5,flag.proj=FALSE)
plot(db.data,name.prop="ENGR.ENC",pch=20,add=T,col="red",inches=3)
map("worldHires",add=T,fill=T,col=8)

# Compare topcut & ok; circles (z2)
cat("Topcut: Mean(estimated) / Mean(data)\n")
ratio = mean(db.extract(kres2,"K.topcut.estim")) /
mean(db.data[,"ENGR.ENC"])
cat("Topcut  = ",ratio,"\n")
ratio = mean(db.extract(kres1,"K.ENGR.ENC.estim")) /
mean(db.data[,"ENGR.ENC"])
cat("Kriging = ",ratio,"\n")

correlation(kres2,"K.topcut.estim","K.ENGR.ENC.estim",kres1,
            name.size="K.z2.estim", flag.aspoint=TRUE, inches=0.15,
            xlab="Ordinary kriging",ylab="Topcut co-
```

```
kriging",mini1=0,mini2=0,
            flag.same=TRUE, flag.iso=TRUE, flag.diag=TRUE)

# conclusion:
# - with Ordinary Kriging high values are spread around the data
# - with the Topcut model high values are estimated only where the
indicator is high
#   because of co-kriging
```

A3.11 Conditional simulations

```
##################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code performs conditional simulation on herring mean length
## collected at trawl stations from an acoustic-trawl survey around t
he Shetland.
## The data were supplied by Marine Scotland Science
## at the Marine Laboratory, Aberdeen, UK.
##
## Author: M.Woillez, Ifremer
##################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load(filename="Demo.herring.len.scot.db.data",objname="db.data")
rg.load(filename="Demo.herring.len.scot.poly.data",objname="poly.data
")

# Select points inside polygon
db.data <- db.polygon(db.data,poly.data,verbose=TRUE)

# Print statistics and histogram
print(db.data,flag.stats=TRUE,names="m.length")
hist(db.extract(db.data,"m.length"),col=8,xlab="m.length",main="")

# Display data
plot(db.data,inches=5,asp=1/cos(mean(db.extract(db.data,"x2"))*pi/180
),
     pos.legend=5,zmax=c(db.stat(db.data,"maxi")),include.bounds=FALS
E)
plot(poly.data,col=4,add=T)
map("worldHires",add=T,fill=T,col=8)

# Define the projection
projec.define(projection="mean", db=db.data)
```

```
# Define the anamorphosis model
model.anam <- anam.fit(db.data,type="gaus",nbpoly=10,draw=T)

# Transform the data into Gaussian
db.data <- anam.z2y(db.data,anam=model.anam)

# Modeling the Gaussian variable Y
vario.data <- vario.calc(db.data)
model.vario <- model.auto(vario.data,struc=melem.name(c(1,3,12)),wmod
e=2,draw=F)
plot(vario.data ,npairdw=T,npairpt=F,inches=0.08,col="black")
plot(model.vario,add=T,col="black")

# Define simulation grid
gnx <- 144
gny <- 90
grid.simu <- db.grid.init(poly.data,nodes=c(gnx,gny))
grid.simu <- db.polygon(grid.simu,poly.data)

# Display of the grid
plot(grid.simu,col=1,title="",pch="+",asp=1)
plot(db.data,inches=3,col=2,pch=19,add=T)
plot(poly.data,col=4,add=T)

# Define the neighborhood
neigh.simu <- neigh.create(ndim=2,type=0)

# Conditional simulation of Y
grid.simu <- simtub(dbin=db.data, dbout=grid.simu, model=model.vario,
                    neigh=neigh.simu, uc = "", mean = 0, seed = 29091
978,
                    nbsimu = 1, nbtuba = 1000, radix = "Simu",modify.
target = TRUE)
print(grid.simu,flag.stats=TRUE,names="Simu.Gaussian.m.length.S1")

# Transform gaussian conditional simulation into raw conditional simu
lation
grid.simu <- anam.y2z(grid.simu,name="Simu.Gaussian.m.length.S1",anam
=model.anam)
print(grid.simu,flag.stats=TRUE,names="Raw.Simu.Gaussian.m.length.S1"
)

# Display conditional simulation of Y
plot(poly.data,col=0,asp=1/cos(mean(db.extract(db=db.data,names="x2")
)*pi/180),
     flag.proj=F)
plot(grid.simu,name="Simu.Gaussian.m.length.S1",pos.legend=5,flag.pro
j=F,add=T)
map("worldHires",add=T,fill=T,col=8);box()

# Display conditional simulation of Z
plot(poly.data,col=0,asp=1/cos(mean(db.extract(db=db.data,names="x2")
)*pi/180),
     flag.proj=F)
plot(grid.simu,name="Raw.Simu.Gaussian.m.length.S1",pos.legend=5,flag
.proj=F,add=T)
map("worldHires",add=T,fill=T,col=8);box()
```

A3.12 Conditional simulations with the presence of zeros

```
#################
## ICES CRR Handbook of geostatistics in R for fisheries and marine e
cology
##
## This code performs conditional simulation on herring acoustic back
scatter
## collected along transects from an acoustic-trawl survey around the
Shetland.
## Acoustic data are characterized by a high proportion of zeros. Con
dition
## simulation are based on transformed simulation and gibbs sampler t
o
## handle the zeros. The data were supplied by Marine Scotland Scienc
e
## at the Marine Laboratory, Aberdeen, UK.
##
## Author: M.Woillez, Ifremer
#################

# Clean workspace
rm(list=ls(all=TRUE))

# Load geostatistical package and others
library(RGeostats)
library(mapdata)

# Inactivate any previous projection
projec.toggle(0)

# Load data
rg.load(filename="Demo.herring.sa.scot.db.data",objname="db.data")
rg.load(filename="Demo.herring.sa.scot.poly.data",objname="poly.data"
)

# Select points inside polygon
db.data <- db.polygon(db.data,poly.data,verbose=TRUE)

# Print statistics and histogram
print(db.data,flag.stats=TRUE,names="sa")
hist(db.extract(db.data,"sa"),col=8,xlab="sa",main="",nclass=100);box
()

# Visualization
plot(db.data,inches=5,asp=1/cos(mean(db.extract(db.data,"x2"))*pi/180
),
     pos.legend=5,zmin=0,zmax=c(db.stat(db.data,"maxi")),include.boun
ds=FALSE)
plot(poly.data,col=4,add=T)
map("worldHires",add=T,fill=T,col=8)

# Define a simple projection
projec.define(projection="mean", db=db.data)

# Compute areas of influence (weights)
db.data <- infl(db.data,nodes=c(300,300),origin=c(-4,57.5),
                extend=c(6.5,4.5),dmax=12.5,
                polygon=poly.data,plot=T,asp=1)
```

```
plot(poly.data,col=4,add=T)

# Define the anamorphosis model
model.anam <- anam.fit(db.data,type="emp",ndisc=db.data$nech,sigma2e=
800,draw=T)

# Transform the data into Gaussian
db.data <- anam.z2y(db.data,anam=model.anam)
db.data <- db.rename(db.data,name="Gaussian.sa",newname="Yp")

# check minimum values
print(db.data,flag.stats=TRUE,names="Yp")
ycut <- qnorm(sum(db.extract(db.data,"sa") == 0) / db.data$nech)
Y <- db.extract(db.data,"Yp")
Y[Y == -10] <- ycut
db.data <- db.replace(db.data,"Yp",Y)
print(db.data,flag.stats=TRUE,names="Yp")

# Modeling Gaussian variable Y
n.H <- 50
vario.Yp <- vario.calc(db.data,lag=2.5,nlag=50)
vario.Y  <- vario.trans.cut(vario.Yp,ycut,n.H)
model.vario.Y <- model.auto(vario.Y,struc=melem.name(c(1,2,3)),draw=F
)
plot(vario.Y ,npairdw=T,npairpt=F,inches=0.08,col="red",xlab="Distanc
e (n.mi.)")
plot(vario.Yp,npairdw=T,npairpt=F,inches=0.08,col="black",add=T)
plot(model.vario.Y,add=T,col="red")

# Define simulation grid
gnx <- 144
gny <- 90
grid.simu <- db.grid.init(poly.data,nodes=c(gnx,gny))
grid.simu <- db.polygon(grid.simu,poly.data)

# Display of the grid
plot(grid.simu,col=1,title="",pch="+",asp=1)
plot(db.data,inches=3,col=2,pch=19,add=T)
plot(poly.data,col=4,add=T)

# Define the neighborhood
neigh.simu <- neigh.create(ndim=2,type=2,nmini=5,nmaxi=100,radius=60)

# Define interval limits for the gibbs
Ymax <- db.extract(db.data,name="Yp",flag.compress=F)
Ymin <- db.extract(db.data,name="Yp",flag.compress=F)
Ymin[Ymin <= ycut] <- -10

# Add those limits into database
db.data<-db.add(db.data,Ymax)
db.data<-db.locate(db.data,db.data$natt,"upper")
db.data<-db.add(db.data,Ymin)
db.data<-db.locate(db.data,db.data$natt,"lower")

# Simulating gaussian values below ycut at datapoints where raw data
value is 0
# while honouring the gaussian variable model and conditional on the
other
# gaussian data values:
```

```
# A Gibbs sampler simulates a gaussian value at each point between it
s
# boundaries ymin and ymax (here -10 and ycut where raw data is 0;
# ymin = ymax = y where y is known)
db.data <-gibbs(db = db.data, model = model.vario.Y, seed = 232132,
                nboot = 10, niter = 100, flag.norm=FALSE, percent=0,
toleps = 1,
                radix = "Gibbs", modify.target = TRUE)
db.data<-db.rename(db.data,"Gibbs.G1","Y")
print(db.data,flag.stats=TRUE,names="Y")

# For each simulation, gaussian values are now defined at all datapoi
nts
# They will be used for classical gaussian simulation

# Conditional simulation of gaussian variable
grid.simu <- simtub(dbin=db.data, dbout=grid.simu, model=model.vario.
Y,
                    neigh=neigh.simu, uc = "", mean = 0, seed = 23213
2, nbsimu = 1,
                    nbtuba = 1000, radix = "Simu",modify.target = TRU
E)
grid.simu <- db.rename(grid.simu,"Simu.Y.S1","Simu.Y")
print(grid.simu,flag.stats=TRUE,names="Simu.Y")

# Transform gaussian conditional simulation
# into raw conditional simulation
grid.simu <- anam.y2z(grid.simu,name="Simu.Y",anam=model.anam)
print(grid.simu,flag.stats=TRUE,names="Raw.Simu.Y")

# Visualization Gibbs sampling step
histYp<-hist(db.extract(db.data,name="Yp"),plot=F,breaks=seq(-4,4,.1)
)
hist(db.extract(db.data,name="Yp"),proba=T,breaks=seq(-4,4,.1),
     xlab="Y+",col=8,main="",
     xlim=c(-4,4),ylim=c(0,ceiling(max(histYp$density))))

hist(db.extract(db.data,name="Y"),proba=T,breaks=seq(-4,4,.1),
     xlab="Y",main="",col=8,
     xlim=c(-4,4),ylim=c(0,ceiling(max(histYp$density))))
lines(seq(-4,4,0.1),dnorm(seq(-4,4,0.1),0,1),col=2)

plot(vario.calc(db.data,lag=2.5,nlag=50),npairdw=T,npairpt=F,
     ylab=expression(gamma),main="",xlab="Distance (n.mi.)",inches=0.
08)
plot(model.vario.Y,add=T,col=2)

# Display conditional simulation of Y
plot(poly.data,col=4,asp=1/cos(mean(db.extract(db.data,"x2"))*pi/180)
,flag.proj=F)
plot(grid.simu,name="Simu.Y",pos.legend=5,flag.proj=F,add=T)
map("worldHires",add=T,fill=T,col=8);box()

# Display conditional simulation of Z
pal2 <- colorRampPalette(c("cyan", "yellow", "red", "black"), bias=4)
plot(poly.data,col=4,asp=1/cos(mean(db.extract(db.data,"x2"))*pi/180)
,flag.proj=F)
plot(grid.simu,name="Raw.Simu.Y",col=pal2(100),pos.legend=5,flag.proj
```

```
=F,add=T)
map("worldHires",add=T,fill=T,col=8);box()

# Display conditional simulation of Z>0
Raw.Simu.Y.sup0 <- grid.simu@items$Raw.Simu.Y
Raw.Simu.Y.sup0[round(Raw.Simu.Y.sup0,2)==0.00] <- NA
grid.simu <- db.add(grid.simu,Raw.Simu.Y.sup0)
plot(poly.data,col=4,asp=1/cos(mean(db.extract(db.data,"x2"))*pi/180)
,flag.proj=F)
plot(grid.simu,name="Raw.Simu.Y.sup0",col=pal2(100),pos.legend=5,flag
.proj=F,add=T)
map("worldHires",add=T,fill=T,col=8);box()
```

Annex 4: List of applications illustrating the theory

The applications presented in the document that illustrate the theory of geostatistics in fisheries and marine ecology are listed below according to the chapter in which they are presented.

Chapter 2 Basic notions

Application 2.1. Change of reference system

Application 2.2. Change of reference system

Chapter 3 Indices of spatial distributions

Application 3.1. Center of gravity, inertia, and isotropy of hake

Application 3.2. Global index of collocation of hake

Application 3.3. Local index of collocation of hake ages 0 and 1

Application 3.4. Microstructure index of hake

Application 3.5. Area indices of hake

Application 3.6. Number of spatial patches of hake

Chapter 4 Structural analysis and variography

Application 4.1. Omnidirectional variogram on demersal survey data

Application 4.2. Comparing and averaging omnidirectional variograms in demersal survey

Application 4.3. Directional variograms on acoustic data

Application 4.4. Transitive covariogram of cephalopod concentrations

Chapter 5 Dispersion and estimation variances

Application 5.1. Global estimation with a variogram

Application 5.2. Global estimation of cephalopod with transitive method

Application 5.3. Global estimation in 1D for acoustic surveys

Chapter 6 Kriging

Application 6.1. Global estimation with a variogram, kriging the global mean over a polygon

Application 6.2. Kriging herring eggs on a spawning bed, neighbourhood, cross-validation, and mapping

Application 6.3. Mapping cephalopod concentrations by transitive kriging

12 Author contact information

Pierre Petitgas
Ifremer
Centre Atlantique
Rue de l'île d'Yeu
BP 21105, 44311 cedex 03 Nantes
France
Pierre.Petitgas@ifremer.fr

Mathieu Woillez
Ifremer
Centre de Bretagne
ZI de la Pointe du Diable, CS-10070
29280 Plouzané
France
Mathieu.Woillez@ifremer.fr

Didier Renard
Mines ParisTech
Centre de Géosciences
35 rue Saint Honoré, 77305
Fontainebleau
France
didier.renard@mines-paristech.fr

Nicolas Bez
Institut de Recherche pour le Développement (IRD)
UMR Marbec Avenue Jean Monnet, CS-30171
34203, Sète
France
nicolas.bez@ird.fr

Jacques Rivoirard
Mines ParisTech
Centre de Géosciences
35 rue Saint Honoré, 77305
Fontainebleau
France
jacques.rivoirard@mines-paristech.fr