Spatial sampling with SamplingStrata

Marco Ballin, Giulio Barcaroli

2020-02-13

Optimization with the *spatial* method

Let us suppose we want to design a sample survey with k Z target variables, each one of them correlated to one or more of the available Y frame variables.

When frame units are georeferenced or geocoded, the presence of spatial auto-correlation can be investigated. This can be done by executing for instance the Moran test on the target variables: if the null hypothesis is rejected (i.e. the hypothesis of the presence of spatial auto-correlation is accepted) then we should take into account also this variance component.

As indicated by deGruijter et al. (2016) and deGruijter, Wheeler, and Malone (2019), in case Z is the target variable, omitting as negligible the *fpc* factor, the sampling variance of its estimated mean is:

$$V(\hat{\bar{Z}}) = \sum_{h=1}^{H} (N_h/N)^2 S_h^2/n_h$$

We can write the variance in each stratum h as:

$$S_h^2 = \frac{1}{N_h^2} \sum_{i=1}^{N_{h-1}} \sum_{j=i+1}^{N_h} (z_i - z_j)^2$$

The optimal determination of strata is obtained by minimizing the quantity O:

$$O = \sum_{h=1}^{H} \frac{1}{N_h^2} \{ \sum_{i=1}^{N_{h-1}} \sum_{j=i+1}^{N_h} (z_i - z_j)^2 \}^{1/2}$$

Obviously, values z are not known, but only their predictions, obtained by means of a regression model. So, in Equation we can substitute $(z_i - z_j)^2$ with

$$D_{ij}^2 = \frac{(\tilde{z}_i - \tilde{z}_j)^2}{R^2} + V(e_i) + V(e_j) - 2Cov(e_i, e_j)$$

where R^2 is the squared correlation coefficient indicating the fitting of the regression model, and $V(e_i)$, $V(e_j)$ are the model variances of the residuals. The spatial auto-correlation component is contained in the term $Cov(e_i, e_j)$.

In particular, the quantity D_{ij} is calculated in this way:

$$D_{ij}^2 = \frac{(\tilde{z}_i - \tilde{z}_j)^2}{R^2} + (s_i^2 + s_j^2) - 2 s_i s_j e^{-k(d_{ij}/range)}$$

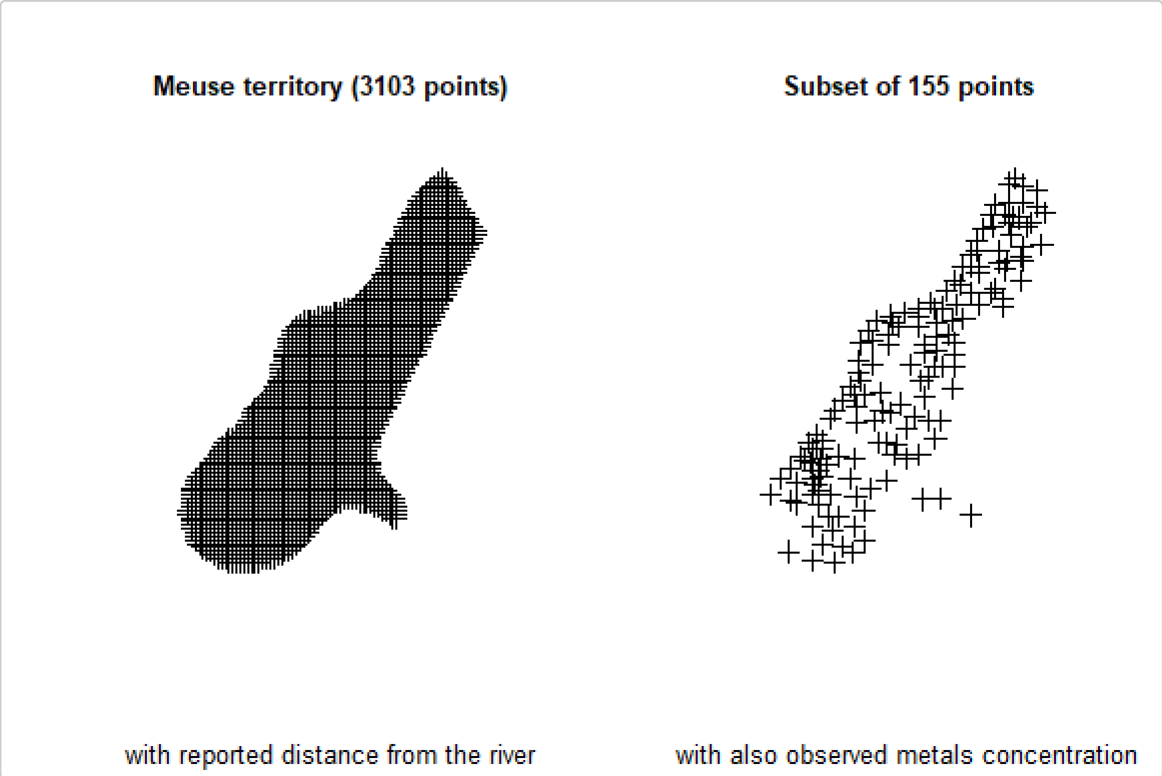
where d_{ij} is the Euclidean distance between two units i and j in the frame (calculated using their geographical coordinates, that must be expressed in meters), the s_i and s_j are estimates of the prediction errors in the single points and *range* is the maximum distance below which spatial auto-correlation can be observed among points. The value of *range* can be determined by an analysis of the spatial *variogram*.

To summarize, when frame units can be geo-referenced, the proposed procedure is the following:

- acquire coordinates of the geographic location of the units in the population of interest;
- fit a *kriging* model (or any other spatial model) on data for each Z;
- obtain predicted values together with prediction errors for each Z and associate them to each unit in the frame;
- perform the optimization step.

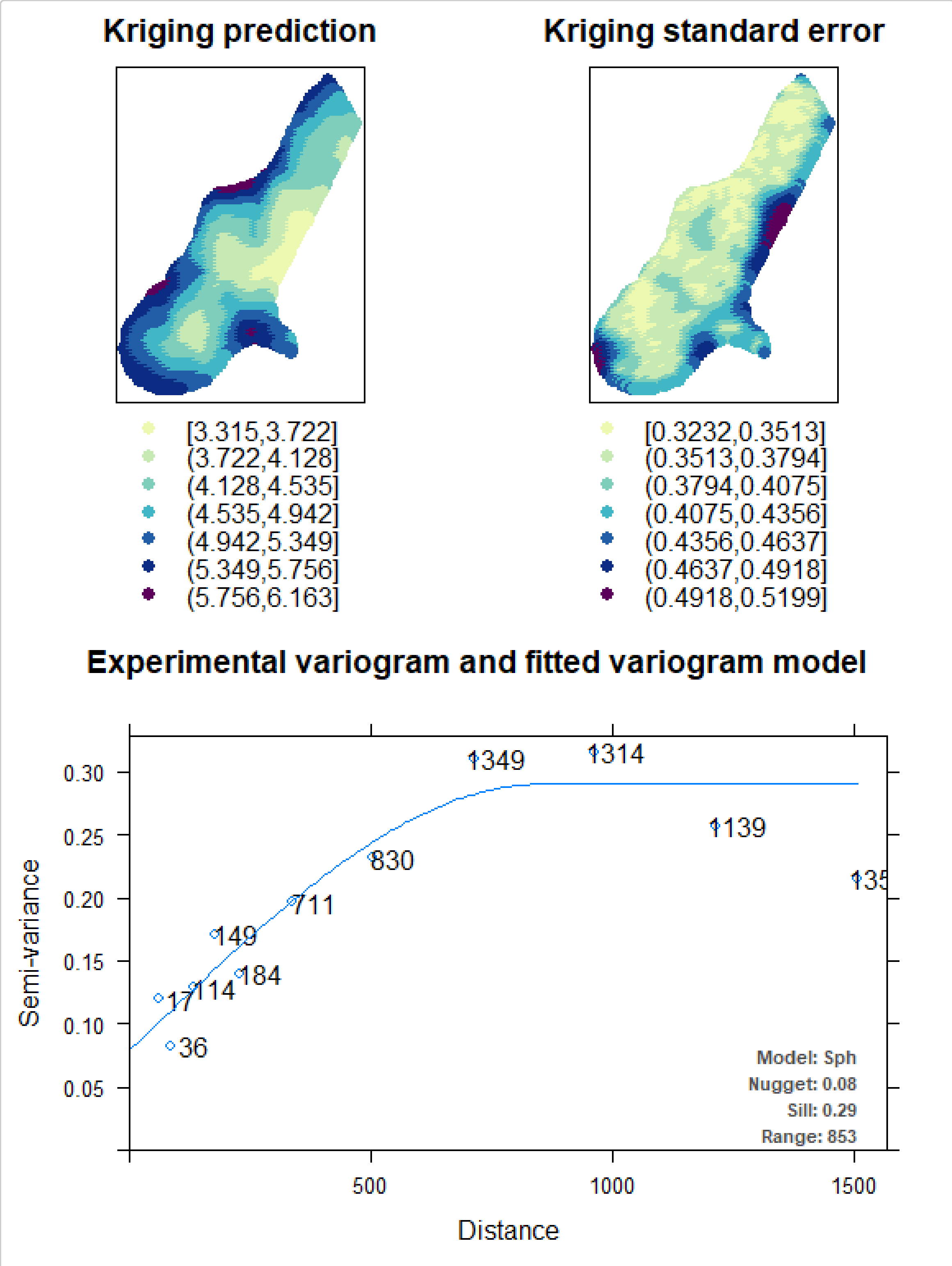
In order to illustrate the "*spatial*" method, we make use of a dataset generally employed as an example of spatially correlated phenomena (in this case, the concentration of four heavy metals in a portion of the river Meuse). This dataset comes with the library "*sp*":

```
library(sp)
# locations (155 observed points)
data("meuse")
# grid of points (3103)
data("meuse.grid")
meuse.grid$id <- c(1:nrow(meuse.grid))
coordinates(meuse)<-c("x","y")
coordinates(meuse.grid)<-c("x","y")
lm_lead <- lm(log(lead) ~ dist,data=meuse)
lm_zinc <- lm(log(zinc) ~ dist,data=meuse)
```

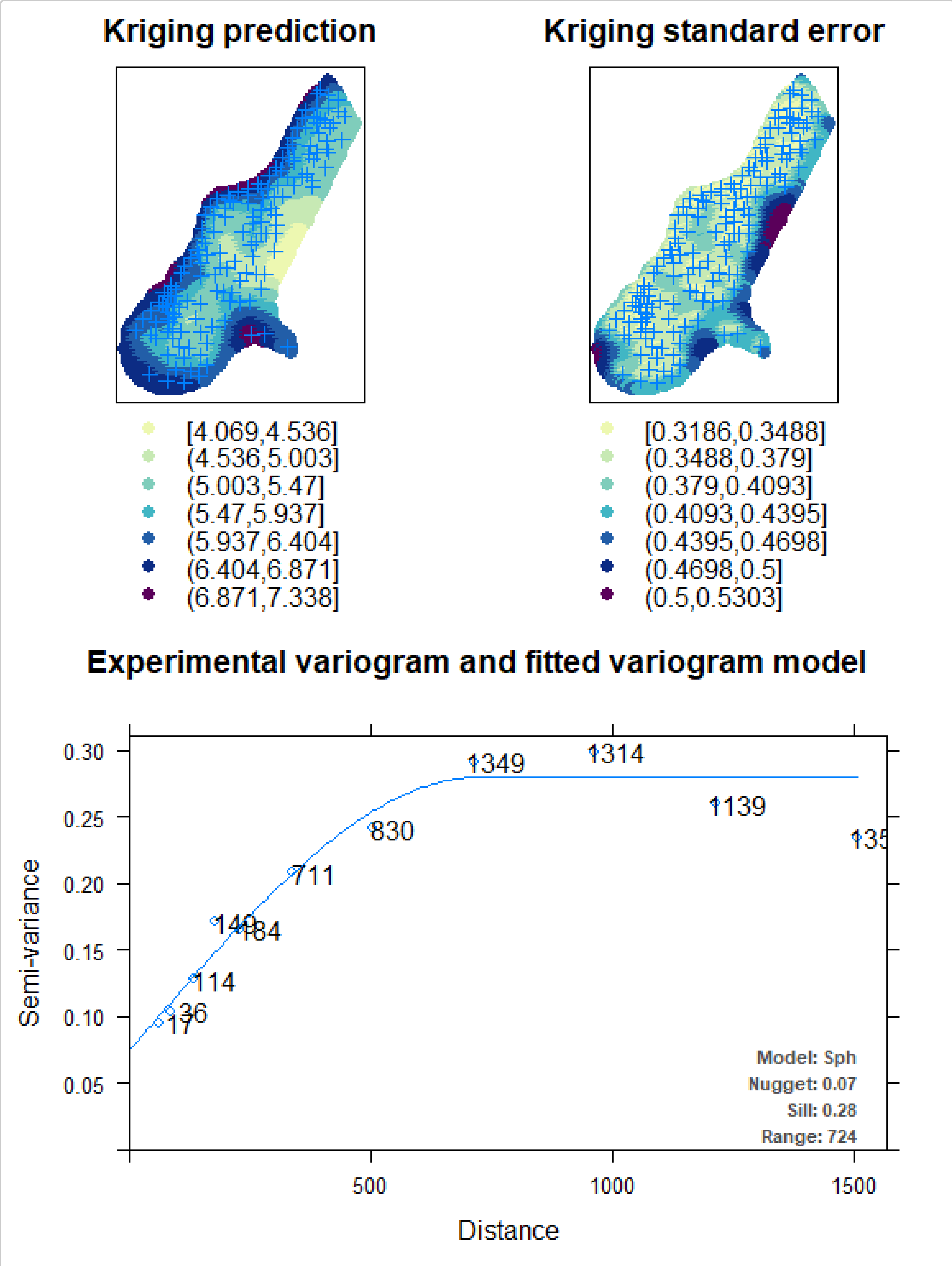


We analyse the territorial distribution of the *lead* and *zinc* concentration, and model (by using the *universal kriging*) their relations with distance from the river, using the subset of 155 points on which these values have been jointly observed:

```
library(automap)
kriging_lead = autoKrige(log(lead) ~ dist, meuse, meuse.grid)
plot(kriging_lead,sp.layout = NULL, justPosition = TRUE)
```




```
kriging_zinc = autoKrige(log(zinc) ~ dist, meuse, meuse.grid)
plot(kriging_zinc, sp.layout = list(pts = list("sp.points", meuse)))
```



Using these *kriging* models, we are able to predict the values of lead and zinc concentration on the totality of the 3,103 points in the Meuse territory:

```
df <- NULL
df$id <- meuse.grid$id
df$lead.pred <- kriging_lead$krige_output@data$var1.pred
df$lead.var <- kriging_lead$krige_output@data$var1.var
df$zinc.pred <- kriging_zinc$krige_output@data$var1.pred
df$zinc.var <- kriging_zinc$krige_output@data$var1.var
df$lon <- meuse.grid$x
df$lat <- meuse.grid$y
df$dom1 <- 1
df <- as.data.frame(df)
head(df)
```

```
#   id lead.pred  lead.var zinc.pred  zinc.var    lon    lat dom1
# 1  1  5.509360 0.1954937  6.736502 0.2007150 181180 333740    1
# 2  2  5.546006 0.1716895  6.785460 0.1749260 181140 333700    1
# 3  3  5.488913 0.1784052  6.698883 0.1826314 181180 333700    1
# 4  4  5.388320 0.1855561  6.558216 0.1906426 181220 333700    1
# 5  5  5.584415 0.1463018  6.841612 0.1465346 181100 333660    1
# 6  6  5.525538 0.1533757  6.749216 0.1549663 181140 333660    1
```

The aim is now to produce the optimal stratification of the 3,103 points under a precision constraint of 1% on the target estimates of the mean *lead* and *zinc* concentrations:

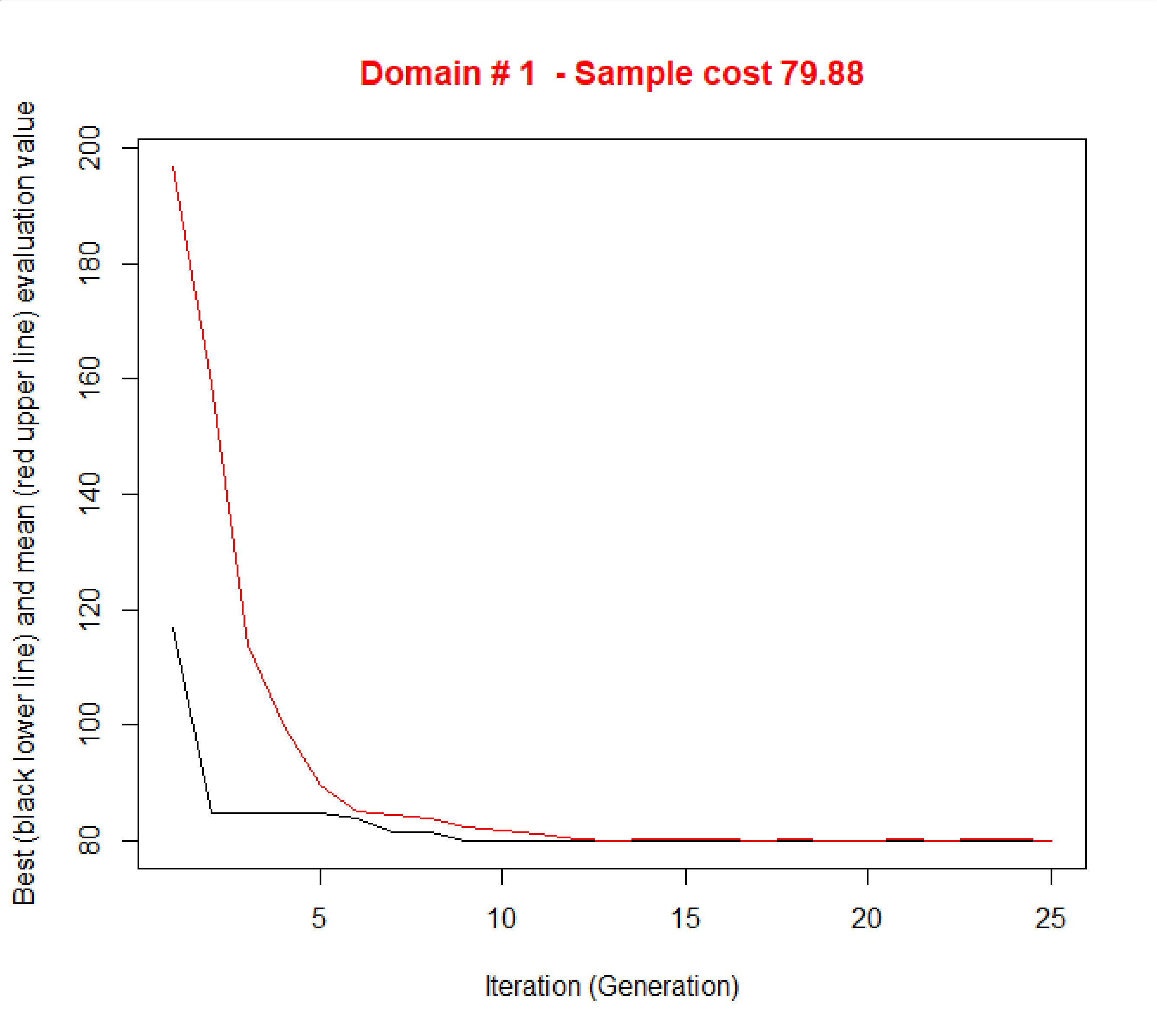
```
library(SamplingStrata)
frame <- buildFrameSpatial(df=df,
                      id="id",
                      X=c("lead.pred","zinc.pred"),
                      Y=c("lead.pred","zinc.pred"),
                      variance=c("lead.var","zinc.var"),
                      lon="lon",
                      lat="lat",
                      domainvalue = "dom1")
#  id       X1       X2       Y1       Y2      var1      var2    lon    lat domainvalue
# 1  1 5.509360 6.736502 5.509360 6.736502 0.1954937 0.2007150 181180 333740           1
# 2  2 5.546006 6.785460 5.546006 6.785460 0.1716895 0.1749260 181140 333700           1
# 3  3 5.488913 6.698883 5.488913 6.698883 0.1784052 0.1826314 181180 333700           1
# 4  4 5.388320 6.558216 5.388320 6.558216 0.1855561 0.1906426 181220 333700           1
# 5  5 5.584415 6.841612 5.584415 6.841612 0.1463018 0.1465346 181100 333660           1
# 6  6 5.525538 6.749216 5.525538 6.749216 0.1533757 0.1549663 181140 333660           1
```

```
cv <- as.data.frame(list(DOM=rep("DOM1",1),
                        CV1=rep(0.01,1),
                        CV2=rep(0.01,1),
                        domainvalue=c(1:1) ))
cv
#    DOM  CV1  CV2 domainvalue
# 1 DOM1 0.01 0.01           1
```

To this aim, we carry out the optimization step by indicating the method *spatial*:

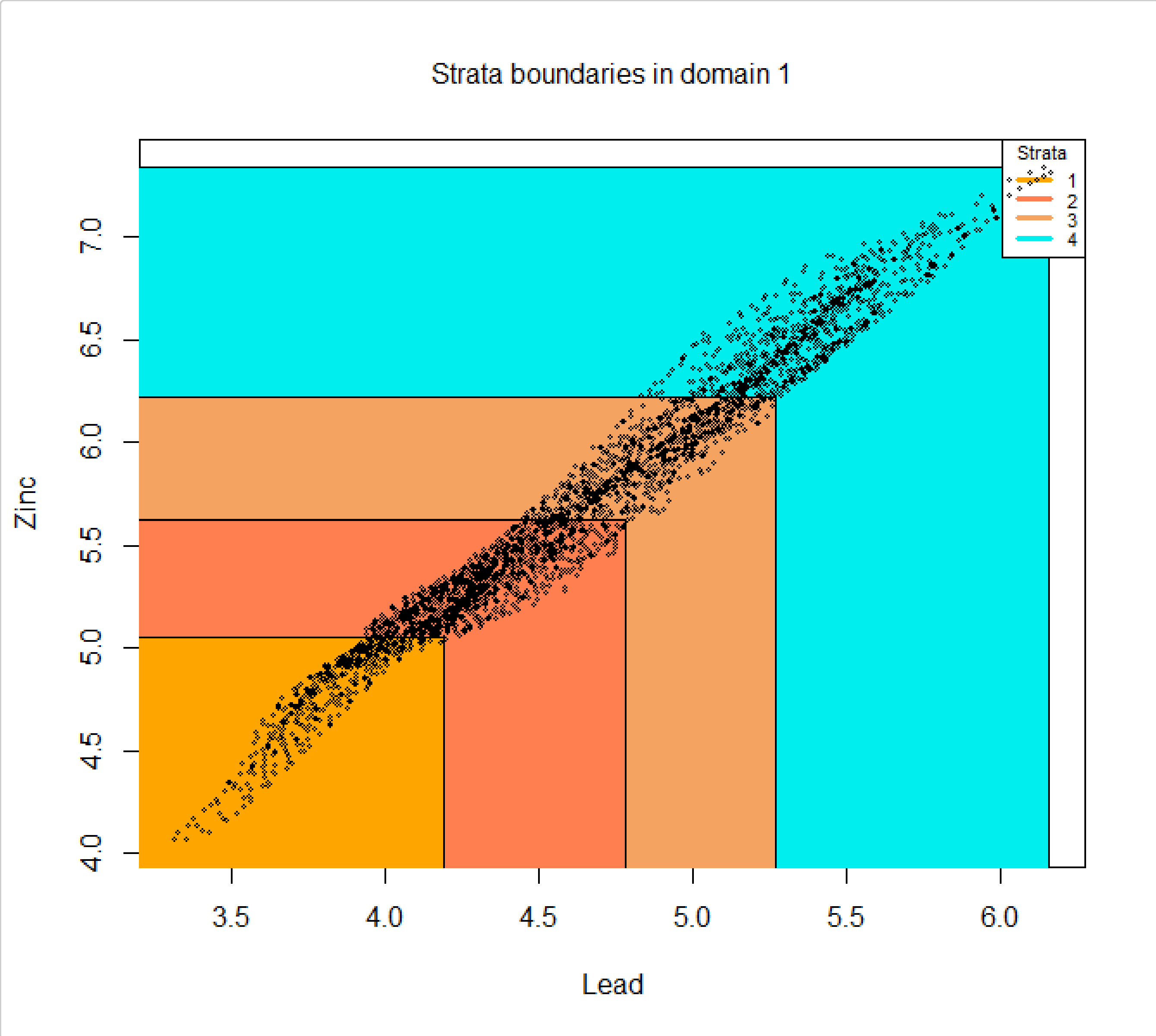
```
set.seed(1234)
solution <- optimStrata (
  method = "spatial",
  errors=cv,
  framesamp=frame,
  iter = 15,
  pops = 10,
  nStrata = 5,
  fitting = c(summary(lm_lead)$r.square,summary(lm_zinc)$r.square),
  range = c(kriging_lead$var_model$range[2],kriging_zinc$var_model$range[2]),
  kappa=1)

# *** Domain :  1   1
#  Number of strata :  3103
# GA Settings
#   Population size       = 10
#   Number of Generations = 15
```

```
#   Elitism                = 2
#   Mutation Chance        = 0.111111111111111
#
#
#
#  *** Sample cost:  79.87774
#  *** Number of strata:  4
#  *** Sample size :  80
#  *** Number of strata :  4
# ---------------------------
```



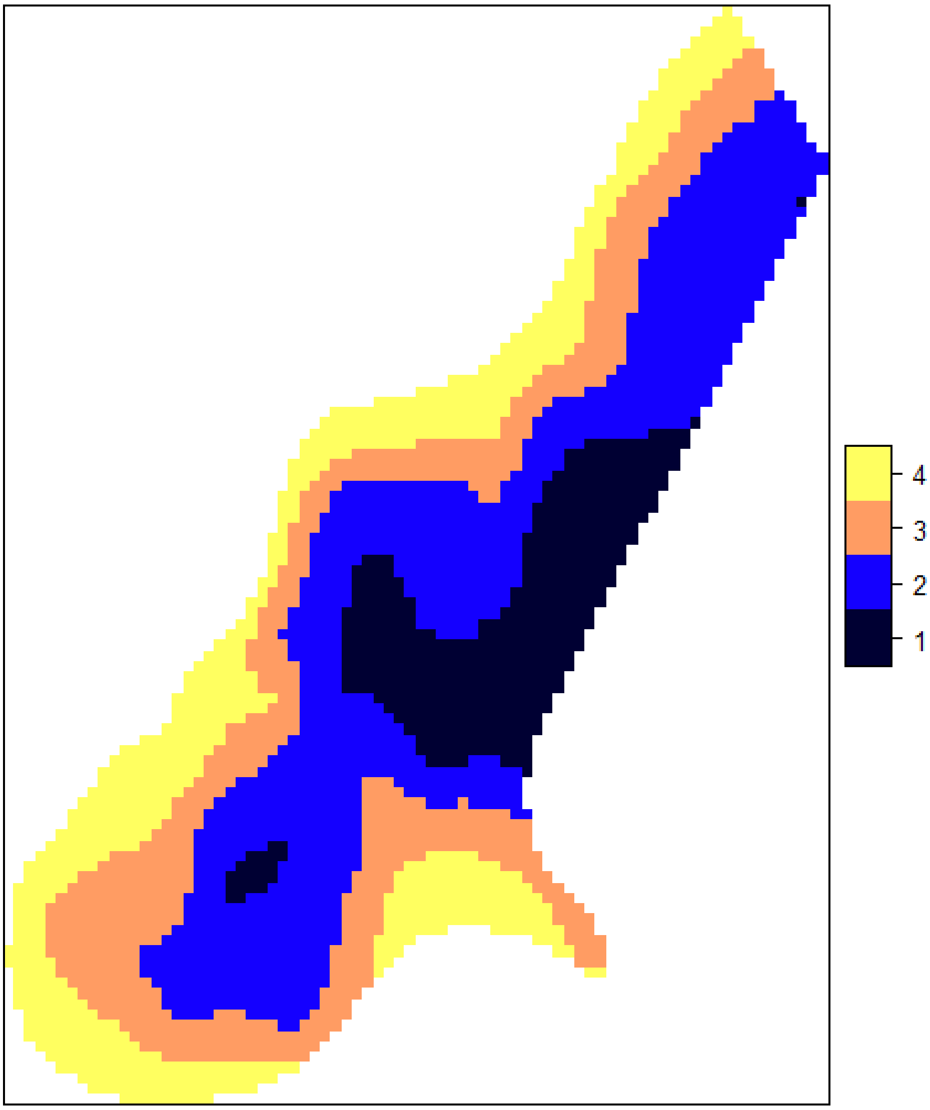
obtaining the following optimized strata:

```
plotStrata2d(solution$framenew,
             solution$aggr_strata,
             domain=1,
             vars=c("X1","X2"),
             labels=c("Lead","Zinc"))
```



that can be visualised in this way:

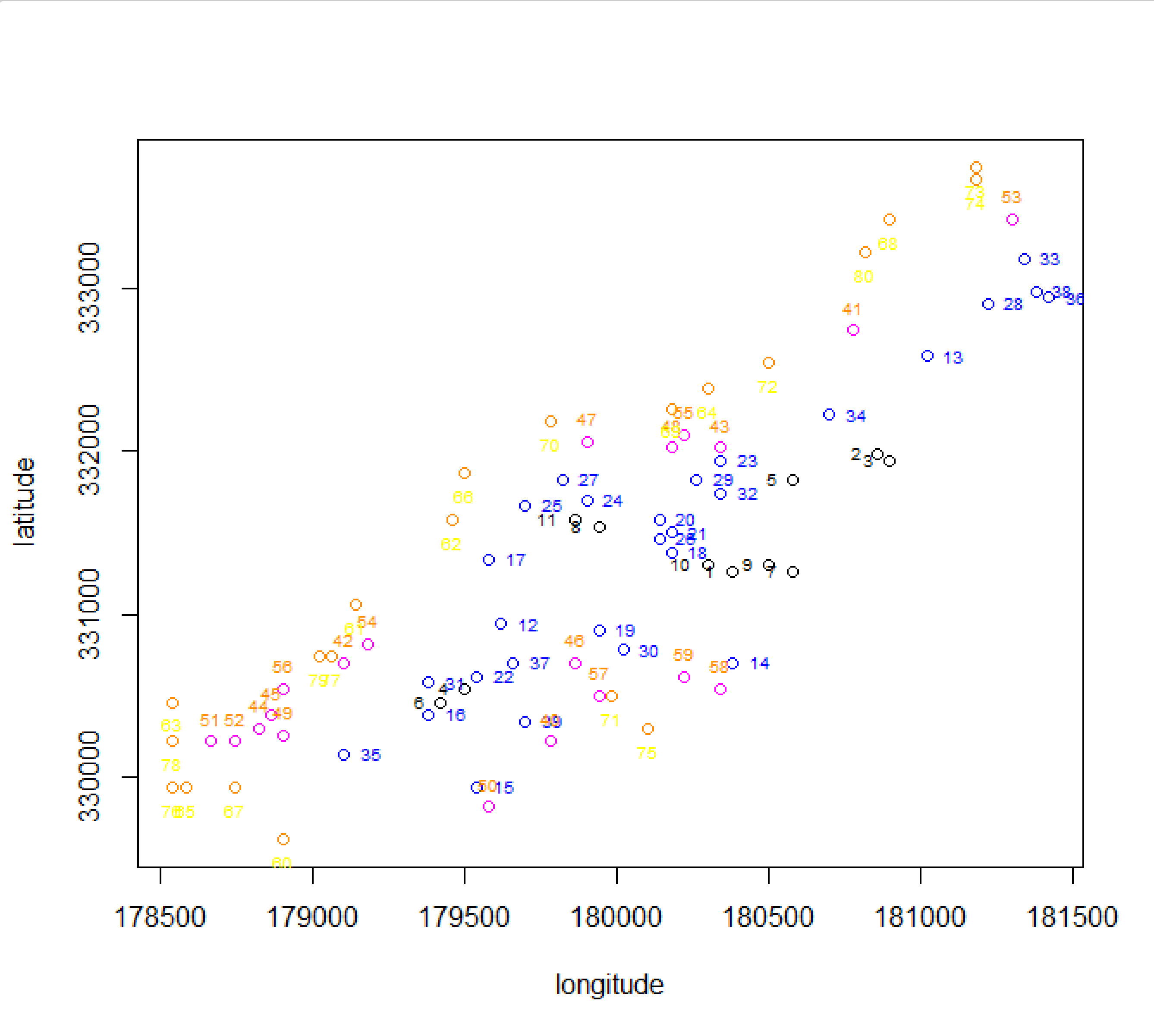
```
frameres <- SpatialPointsDataFrame(data=framenew, coords=cbind(framenew$LON,framenew$LAT) )
frameres2 <- SpatialPixelsDataFrame(points=frameres[c("LON","LAT")], data=framenew)
frameres2$LABEL <- as.factor(frameres2$LABEL)
spplot(frameres2,c("LABEL"), col.regions=bpy.colors(5))
```



We can now proceed with the selection of the sample:

```
s <- selectSample(solution$framenew,
                  solution$aggr_strata)

# *** Sample has been drawn successfully ***
#  80  units have been selected from  4  strata
```

whose units are so distributed in the territory:



References

deGruijter, J. J., A. B. McBratney, B. Minasny, I. Wheeler, B. P. Malone, and U. Stockmann. 2016. "Farm-Scale Soil Carbon Auditing." *Geoderma*, no. 265: 12–130.

deGruijter, J. J., I. Wheeler, and B. P. Malone. 2019. "Using Model Predictions of Soil Carbon in Farm-Scale Auditing - a Sofwtare Tool." *Agricultural Systems*, no. 169(C): 24–30.