Topic 1: Monitoring Network Optimization

In this topic we we apply the previously considered methods for monitoring network optimization. In particular, we consider

- Introduction to **monitoring network optimization**.
- Example based on **mean kriging variance**.
- Example based on **delineating contours**.
- **Illustrate results with Meuse data**.

Monitoring Network Optimization

Monitoring is an important part of data analysis and can be very costly. Monitoring programs have to be designed, implemented, evaluated, and sometimes need to be enlarged or shrunken.

The difficulty of finding optimal network designs is that a quantitative criterion is often a priori not present. The criterion should be selected together with customers and often changes later. For example, should one focus on mean kriging variances, or variance of some global mean estimator, or rather on the ability to delineate a particular contour?
We consider the meuse data again:

```
> library(gstat)
> library(sp)
> data(meuse)
> data(meuse.grid)
> coordinates(meuse) <- c("x", "y")
> coordinates(meuse.grid) <- ~x+y
> v  <- variogram(log(zinc) ~ 1, meuse)
> v.fit <- fit.variogram(v, vgm(1, "Sph", 800, 1))
```

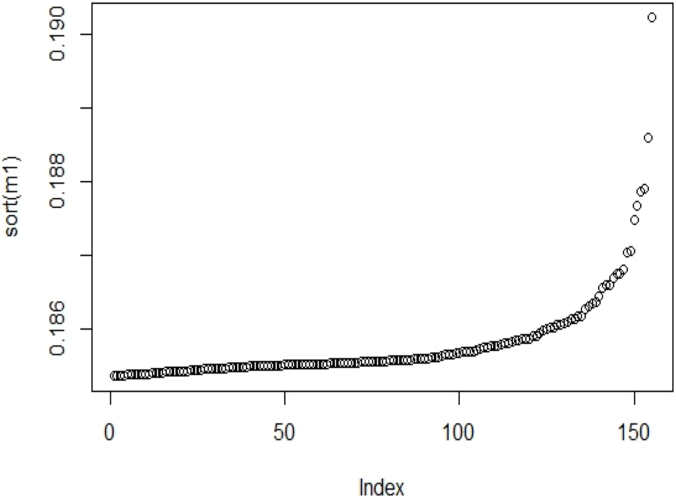
Example 1. A very simple approach towards monitoring network optimization is to find the point whose removal leads to the **smallest increase in mean kriging variance**:

```
> m1 <- sapply(1:155, function(x) mean(krige(log(zinc)~1,
+ meuse[-x,], meuse.grid, v.fit)$var1.var))
> which(m1 == min(m1))
[1] 72
```

It points to observation 72 as the first candidate for removal. Looking at the sorted magnitudes of change in mean kriging variance by

```
> plot(sort(m1))
```

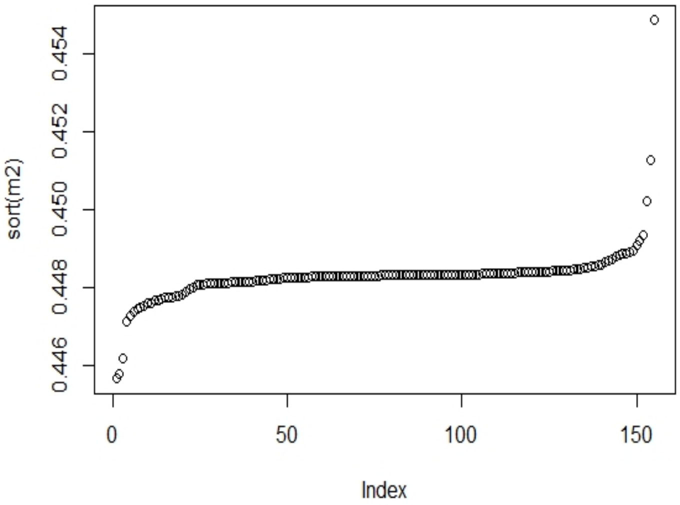
will reveal that for several other candidate points their removal will have an almost identical effect on the mean variance.


0.186
0.188
0.190
sort(m1)
0
50
100
150
Index

Example 2. Another approach could be, for example, to **delineate the 1000 ppm contour**.

We could, for example, express the doubt about whether a location is below or above 1000 as the closeness of $G\left(\frac{\hat{Z}(s_0)-1000}{\sigma(s_0)}\right)$ to 0.5, with $G(\cdot)$ the Gaussian distribution function.

```
> cutoff <- 1000
> f <- function(x) {
    +   kr = krige(log(zinc)~1, meuse[-x,], meuse.grid, v.fit)
    +   mean(abs(pnorm((kr$var1.pred - log(cutoff))/
    +   sqrt(kr$var1.var)) - 0.5))
    + }
> m2 <- sapply(1:155, f)
> which(m2 == max(m2))
[1] 82
> plot(sort(m2))
```


sort(m2)
0.446
0.448
0.450
0.452
0.454
0
50
100
150
Index

To compare the obtained results we plot top and bottom 10% candidate points for removal for both criteria:

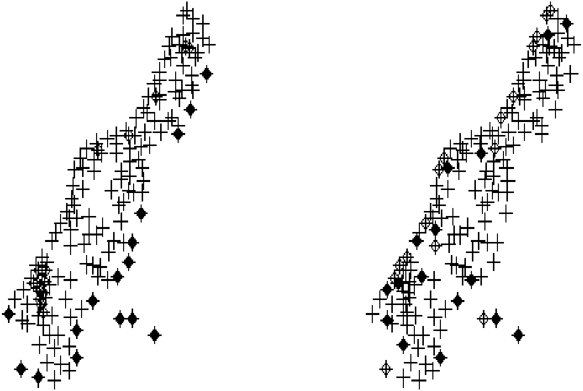
```
> layout(matrix(1:2, 1, 2))

> plot(meuse)
> points(meuse[m1 < quantile(m1,.1),], pch=1)
> points(meuse[m1 > quantile(m1,.9),], pch=16)

> plot(meuse)
> points(meuse[m2 < quantile(m2,.1),], pch=16)
> points(meuse[m2 > quantile(m2,.9),], pch=1)
```

One can see that different objectives lead to different candidate points.

Also, deciding based on the kriging variance alone results in an outcome that is highly predictable from the points configuration alone, i.e. points in the densest areas are candidate for removal.



For adding observation points, one could loop over a fixed grid of locations and find the point that increases the objective most.

Key R commands	
sapply(x, fun)	*applies fun to the corresponding element of x*
which(x)	*gives the TRUE indices of a logical object*
layout(mat, ...)	*divides the device up into the number rows and columns specified by the matrix mat*
matrix(...)	*defines a matrix*
quantile(x, prob)	*produces a sample quantile corresponding to the probability prob*
points(x, ...)	*draw a sequence of points*