Topic 2: Sample covariance and variogram in R

This topic overview several methods of estimating the covariance and variogram functions in R. In particular, we consider

- **Sample covariance** functions.
- Computing **sample variograms.**
- Computing **directional variograms.**
- Studying spatial dependencies in **s100 data**.

Sample covariances

Sample covariance and variogram are statistical estimators of covariance and semi-variogram functions by using actual spatial data.

Sample covariance and variogram are two commonly used measures in geo-statistics and spatial statistics to describe the spatial relationship between pairs of observations in a dataset.

They measure how two variables co-vary, or vary together if they are separated by a given distance. They measure the degree to which changes in one variable are associated with changes in another variable. In the context of spatial data, sample covariance can be used to measure the spatial similarity or dependence between pairs of spatial locations.

A simple way to check whether spatial correlation is present or not is to make scatter plots of pairs $X_{\mathbf{s}_i}$ and $X_{\mathbf{s}_j}$, grouped according to their separation distance $\|\mathbf{s}_i - \mathbf{s}_j\|$.

- **Variogram cloud.** Plot $\left(X_{\mathbf{s}_i} - X_{\mathbf{s}_j}\right)^2$ versus $\|\mathbf{s}_i - \mathbf{s}_j\|^{1/2}$ (Euclidean distance) for all pairs of observations.
  The variogram cloud implicitly assumes isotropy (does not differentiate any directions)
- **Sample semivariogram** The traditional sample semivariogram $\hat{\gamma}$ suggested by Matheron (1971) is:

$$\hat{\gamma}(\mathbf{v}) = \frac{1}{2N(\mathbf{v})} \sum_{N(\mathbf{v})} (X_{\mathbf{s}_i} - X_{\mathbf{s}_j})^2,$$

where $N(\mathbf{v})$ are the number of data pairs $\mathbf{s}_i$ and $\mathbf{s}_j$ separated by $\mathbf{v}$. Note that this implicitly assumes stationarity of some kind.

- One can display the variogram along selected directions and obtain **directional variograms**, for example, N-S, NW-SE, E-W, and NE-WE.

- **Sample covariance.** Similar to sample semivariogram. Plot of $\hat{C}(\mathbf{v})$ versus $\mathbf{v}$, where

$$\hat{C}(\mathbf{v}) = \frac{1}{N(\mathbf{v})} \sum_{N(\mathbf{v})} (X_{\mathbf{s}_i} - \bar{X})(X_{\mathbf{s}_j} - \bar{X}),$$

$\bar{X}$ is the sample mean.

Sample variogram in R

Sample variograms are calculated using the function **variog**.

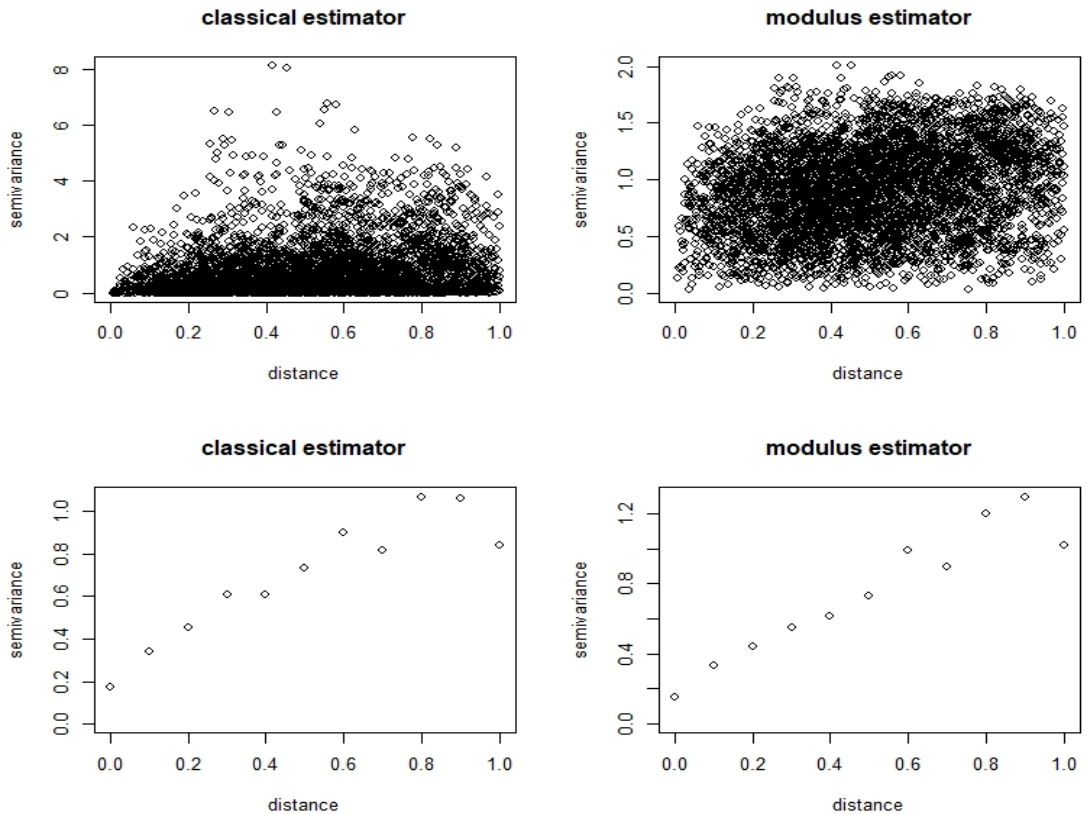
There are options for the classical or modulus estimator.

Results can be returned as variogram clouds, binned or smoothed variograms.

```
> library(geoR)
> data(s100)
> cloud1 <- variog(s100, option = "cloud", max.dist=1)
> cloud2 <- variog(s100, option = "cloud", estimator.type =
"modulus", max.dist=1)
> bin1 <- variog(s100, uvec=seq(0,1,l=11))
> bin2  <- variog(s100, uvec=seq(0,1,l=11),
estimator.type= "modulus")
```

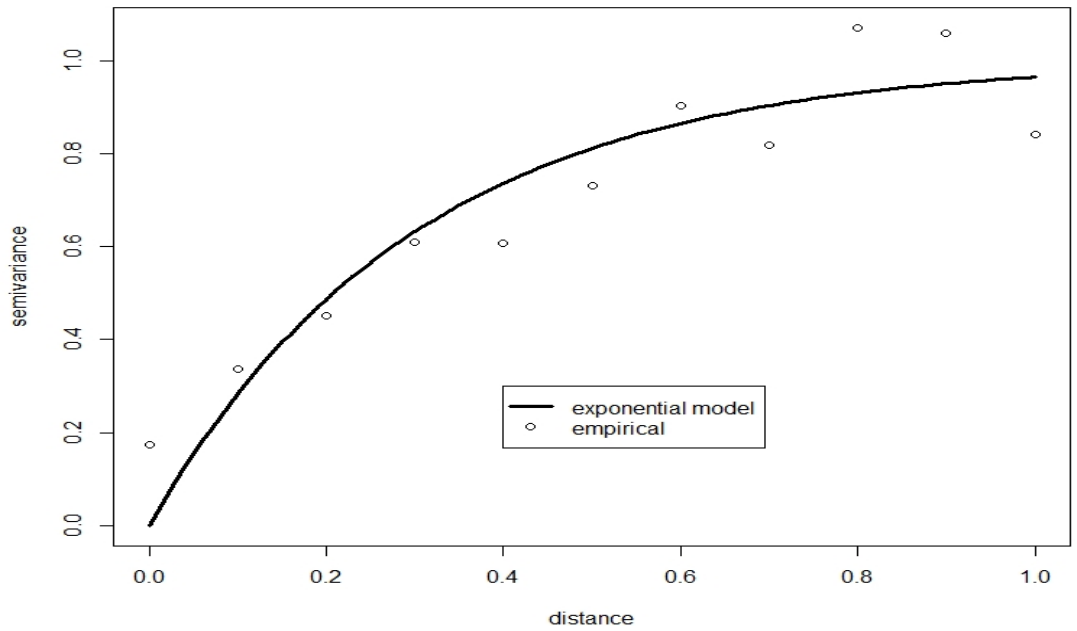
To plot the obtained variogram we create a matrix of 2 x 2 plots :

```
> par(mfrow=c(2,2))
> plot(cloud1, main = "classical estimator")
> plot(cloud2, main = "modulus estimator")
> plot(bin1, main = "classical estimator")
> plot(bin2, main = "modulus estimator")
```



Theoretical and sample variograms can be plotted and visually compared. For example, to plot a figure which shows the theoretical variogram model used to simulate the data $\mathrm{s100}$ and the estimated variogram, type

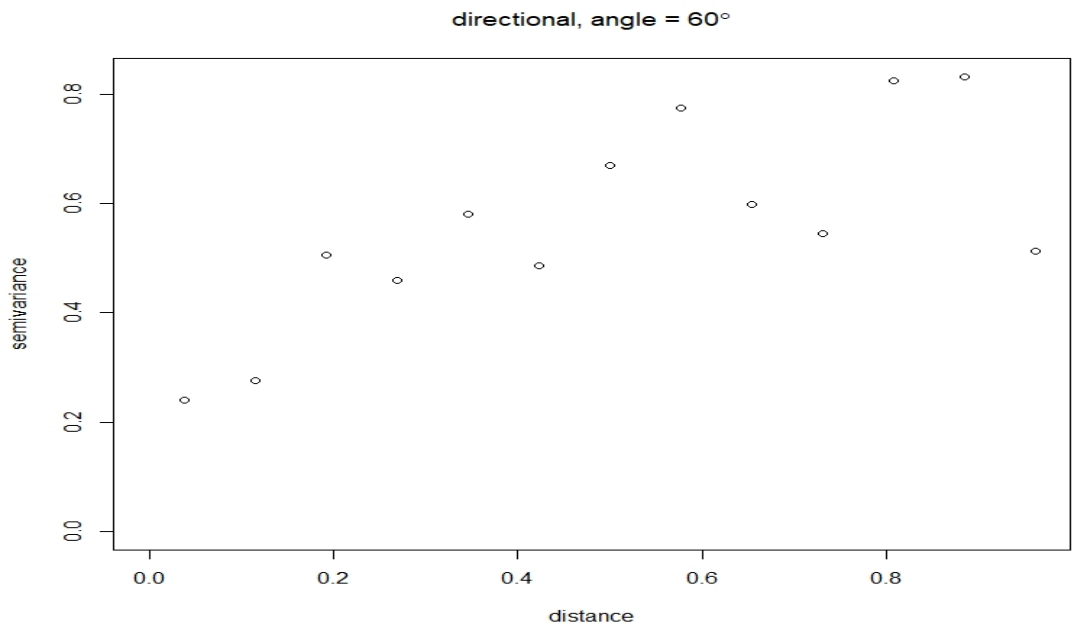
```
> plot(bin1)
> lines.variomodel(cov.model = "exp", cov.pars = c(1,0.3),
+ nugget = 0,  max.dist = 1,  lwd = 3)
> legend(0.4, 0.3, c("exponential model", "empirical"),
+ lty=c(1,0), lwd = c(3,1), pch=c(NA,1))
```



Directional variogram in R

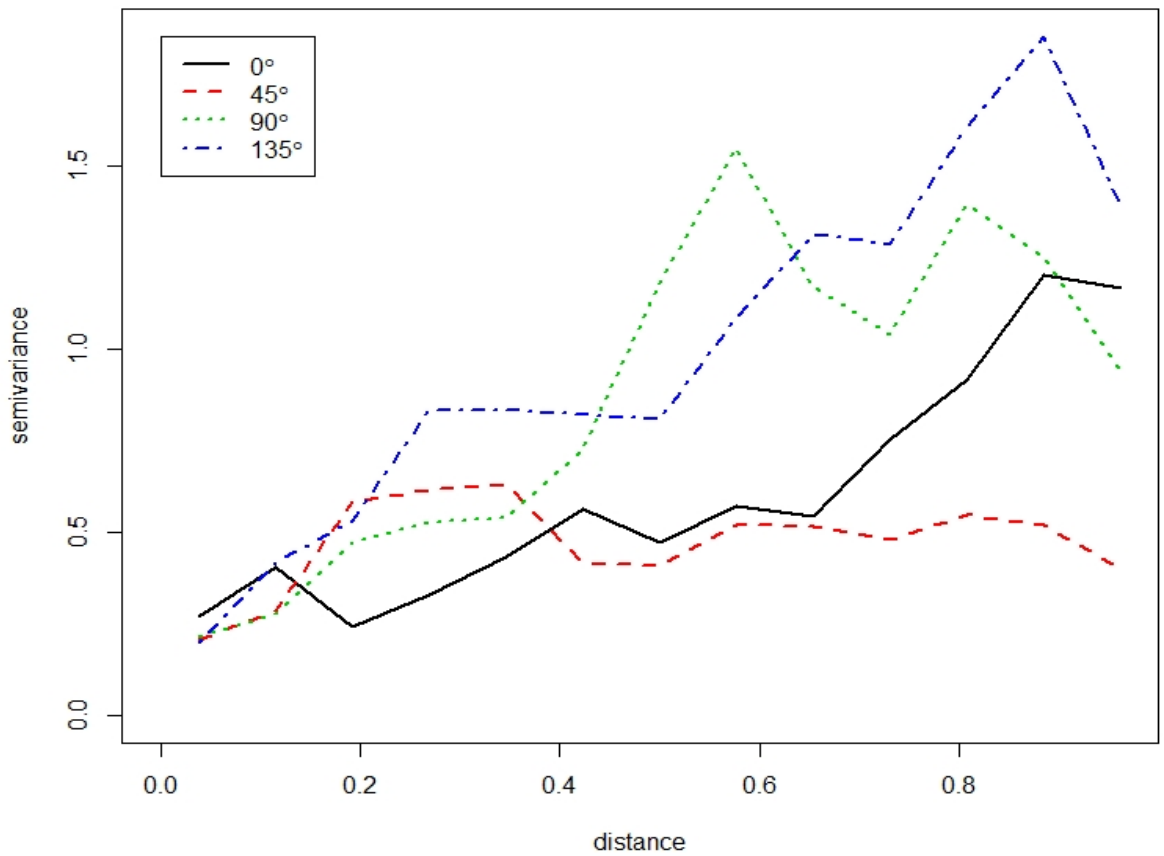
Directional variograms can be computed by the function **variog** using the arguments direction and tolerance. For example, to compute a variogram for the direction 60 degrees with the default tolerance angle (22.5 degrees) the command would be

```
> vario60 <- variog(s100, max.dist = 1, direction=pi/3)
> plot(vario60)
> title(main=expression(paste("directional, angle =",60*degree)))
```



For a quick computation of variograms in four directions the function **variog4** can be used.

```
> vario.4 <- variog4(s100, max.dist = 1)
> plot(vario.4, lwd=2)
```



Key R commands

variog(x)	*computes sample (empirical) variograms*
lines.variomodel(x)	*adds a line with a variogram model specified by a user to a current variogram plot*
legend(x)	*adds a legend to a plot*
title(main = ..., ...)	*add labels to a plot*
variog4(geodata,...)	*computes directional variograms for 4 directions*
par(mfrow=c(nrows, ncols))	*creates a matrix of nrows x ncols plots*
seq(from, to, length.out)	*generates a regular sequence using starting and end values of a sequence and its length*