Interpolation of spatial data (1)

Lecture 9

Joaquin Cavieres

1. (Semi)variogram fitting

Semivariogram is a function describing the degree of spatial correlation of a spatial random variable. In spatial modeling, semivariogram begins with a graph of the empirical semivariogram, which is the half of average squared difference between points separated by a distance. The semivariogram is calculated as:

$$2\hat{\gamma}(\boldsymbol{h}) = \frac{1}{|N(\boldsymbol{h})|}\sum_{N(\boldsymbol{h})}(Y(\boldsymbol{s}_i) - Y(\boldsymbol{s}_j))^2, \tag{1}$$

where N is the pairs separated by the vector $\boldsymbol{h}$, $\boldsymbol{s}_i$ is the starting locations and $\boldsymbol{s}_j$ is the ending location

1.1. Calfornia air pollution data

We will use the airqual dataset to interpolate the levels of ozone for California (averages for 1980-2009). The response variable is "OZDLYAV" (unit is parts per billion).

```
# Installing the rspat package
#if (!require("rspatial")) remotes::install_github('rspatial/rspatial')

# Read the data
library(rspatial)
x <- sp_data("airqual")
x$res <- x$OZDLYAV * 1000
```

Now, we create a `SpatialPointsDataFrame` and transform to Teale Albers. Note the units=km, which was needed to fit the variogram.

```r
library(rgdal)
coordinates(x) <- ~LONGITUDE + LATITUDE
proj4string(x) <- CRS('+proj=longlat +datum=NAD83')
TA <- CRS("+proj=aea +lat_1=34 +lat_2=40.5 +lat_0=0 +lon_0=-120 +
          x_0=0 +y_0=-4000000 +datum=WGS84 +units=km")
aq <- spTransform(x, TA)
```

Create an template raster file to interpolate. Coerce that to a 'SpatialGrid' object (a different representation of the same idea)

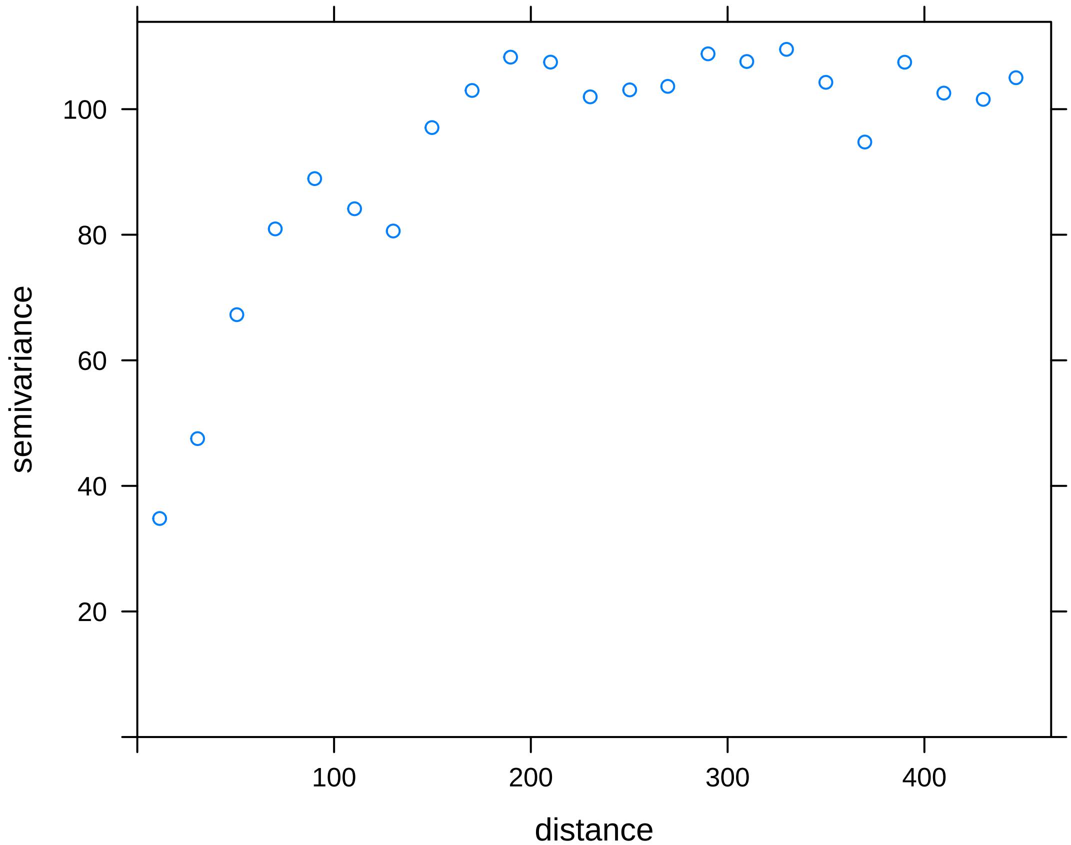
```r
cageo <- sp_data('counties.rds')
ca <- spTransform(cageo, TA)
r <- raster(ca)
res(r) <- 10  # 10 km if your CRS's units are in km
g <- as(r, 'SpatialGrid')
```

Now we will create an empirical variogram using the package "gstat" as follow

```r
library(gstat)
gs <- gstat(formula = res~1, locations = aq)
v <- variogram(gs, width=20)
head(v)
```

```
##     np      dist    gamma dir.hor dir.ver   id
## 1 1010  11.35040 34.80579       0       0 var1
## 2 1806  30.63737 47.52591       0       0 var1
## 3 2355  50.58656 67.26548       0       0 var1
## 4 2619  70.10411 80.92707       0       0 var1
## 5 2967  90.13917 88.93653       0       0 var1
## 6 3437 110.42302 84.13589       0       0 var1
```

```r
plot(v)
```

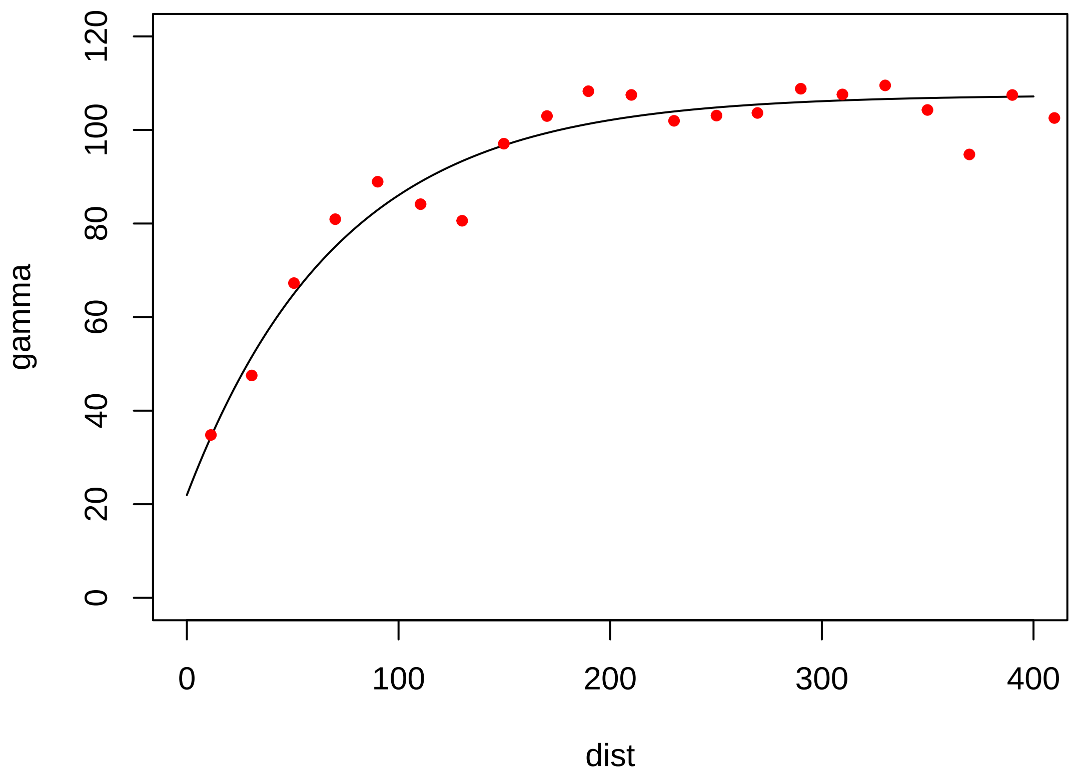


and then fit a model variogram:

```
fve <- fit.variogram(v, vgm(85, "Exp", 75, 20))
fve
```

```
##   model    psill    range
## 1   Nug 21.96600  0.00000
## 2   Exp 85.52957 72.31404
```

```
plot(variogramLine(fve, 400), type='l', ylim=c(0,120))
points(v[,2:3], pch=20, col='red')
```

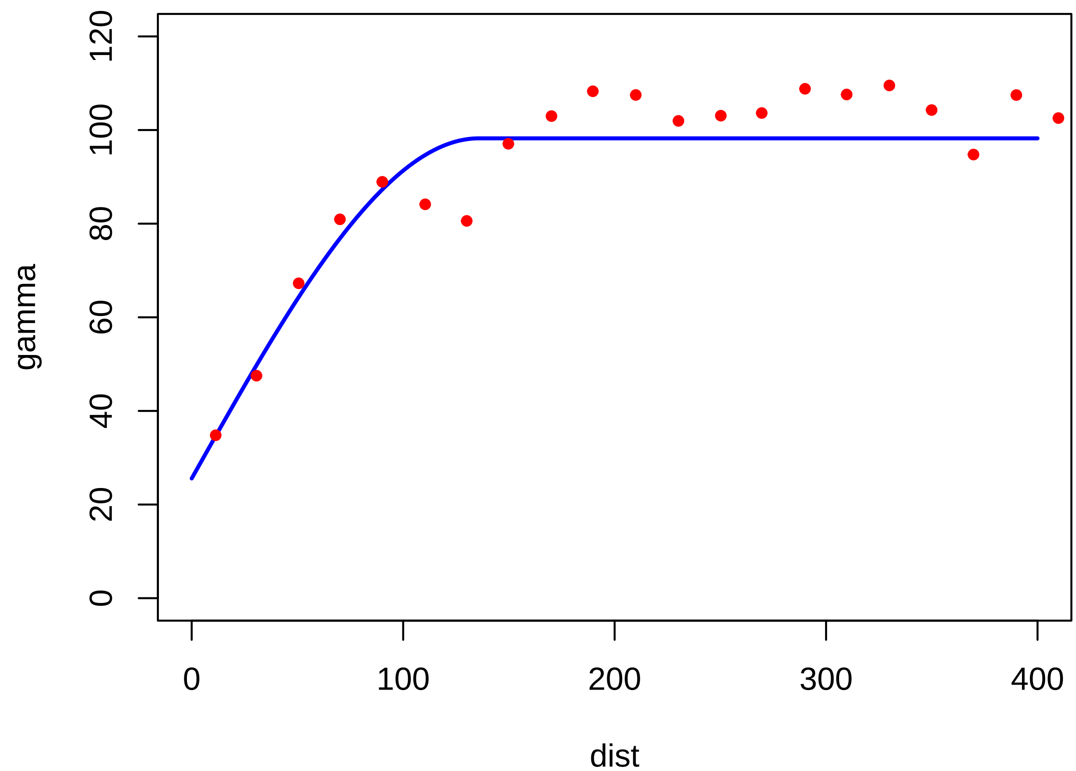


You can change the type of variograms (spherical in stead of exponential) and compare them:

```
fvs <- fit.variogram(v, vgm(85, "Sph", 75, 20))
fvs
```

```
##   model    psill    range
## 1   Nug 25.57019   0.0000
## 2   Sph 72.65881 135.7744
```

```
plot(variogramLine(fvs, 400), type='l', ylim=c(0,120) ,col='blue', lwd=2)
points(v[,2:3], pch=20, col='red')
```



1.2. Meuse (concentrations) data

By the `variovariogram()` function of "gstat" package we can calculate the experimental variogram of the meuse data, for this we will use the zinc variable. Before that we have to define "x" and "y" variables to coordinates

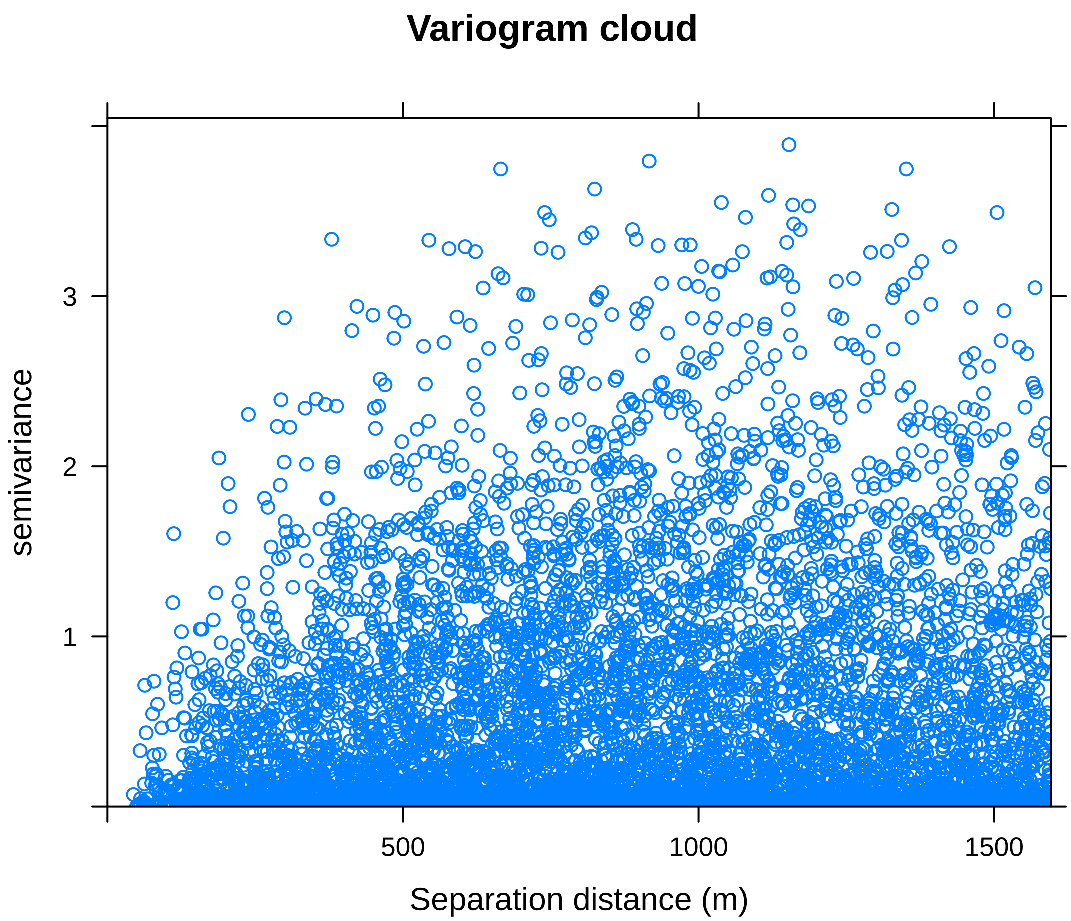
```
library(sp)
data(meuse)

# Coordinates
coordinates(meuse) = ~x+y
```

We will use all default values to calculate the semivariogram values for all pairs of locations within a dataset and plots them as a function of the distance that separates the two locations.

```r
vario1_meuse <- variogram(log(zinc)~ 1, data = meuse, cloud=T)
head(vario1_meuse)
```

```
##        dist       gamma dir.hor dir.ver   id left right
## 1  70.83784 0.006065804       0       0 var1    2     1
## 2 118.84864 0.109534743       0       0 var1    3     1
## 3 141.56624 0.167153095       0       0 var1    3     2
## 4 259.23927 0.952808244       0       0 var1    4     1
## 5 282.85155 1.110920725       0       0 var1    4     2
## 6 143.17123 0.416229664       0       0 var1    4     3
```

```r
plot(vario1_meuse, main = "Variogram cloud", xlab = "Separation distance (m)")
```

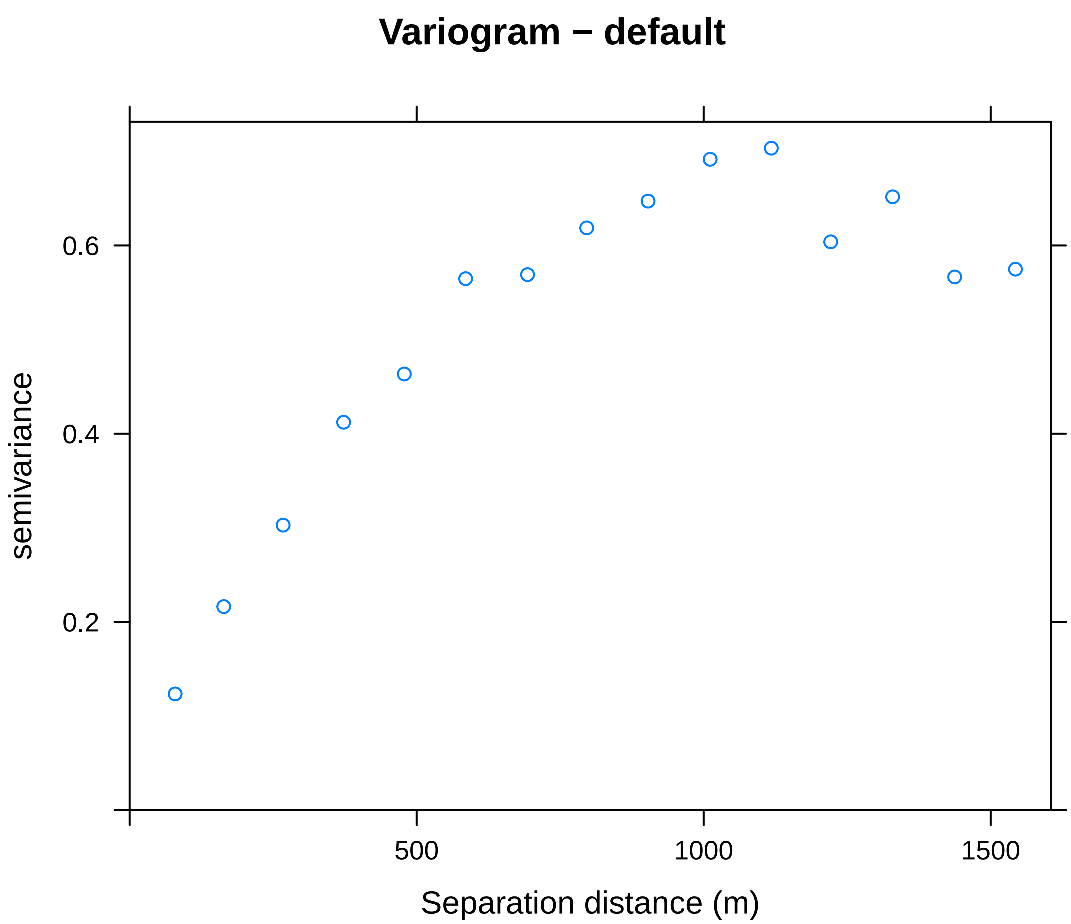


The variogram cloud shows all point-pairs, but it is difficult to examine the general pattern of spatial dependence. For examine the spatial dependence, we will calculate the empirical variogram, which organizes the cloud into bins, like a histogram.

```r
vario2_meuse <- variogram(log(zinc)~ 1, data = meuse, cloud = F)
head(vario2_meuse)
```

```
##    np      dist     gamma dir.hor dir.ver   id
## 1  57  79.29244 0.1234479       0       0 var1
## 2 299 163.97367 0.2162185       0       0 var1
## 3 419 267.36483 0.3027859       0       0 var1
## 4 457 372.73542 0.4121448       0       0 var1
## 5 547 478.47670 0.4634128       0       0 var1
## 6 533 585.34058 0.5646933       0       0 var1
```

```r
plot(vario2_meuse, main = "Variogram - default", xlab = "Separation distance (m)")
```



References

- Moraga, P., 2023. Spatial Statistics for Data Science: Theory and Practice with R

- Bivand, R. S., Pebesma, E. J., Gomez-Rubio, V., & Pebesma, E. J. (2008). Applied spatial data analysis with R (Vol. 747248717, pp. 237-268). New York: Springer.

- Spatial Data Science with R and “terra”. https://rspatial.org/index.html.