SpatCourse_spatial2

Stefano Larsen

3/20/2021

Simulate landscapes with different spatial patterns

Here we will use *gstat* to predict (kriging simulation) gridded landscapes starting from assigned variogram parameters (range, sill and nugget). We will change the *range* parameter to simulate a gradient vs a more patchy spatial variation. We will use these landscapes to explore how the shape of variograms can inform on the underlying spatial patterns.

```
library(vegan)
library(tidyverse)
library(gstat)
library(sp)
#library(raster)
#library(maptools)
library(nlme)
library(ggplot2)
library(cowplot)
```

```
set.seed(10)
```

Simulate a *field* with spatial structure as defined by semi-variogram parameters

We will use different parameters to examine how gradient vs patchy spatial structures can be revealed by plotting variograms

Let's start with a gradient spatial autocorrelation. Defining a long 'range' autocorrelation.

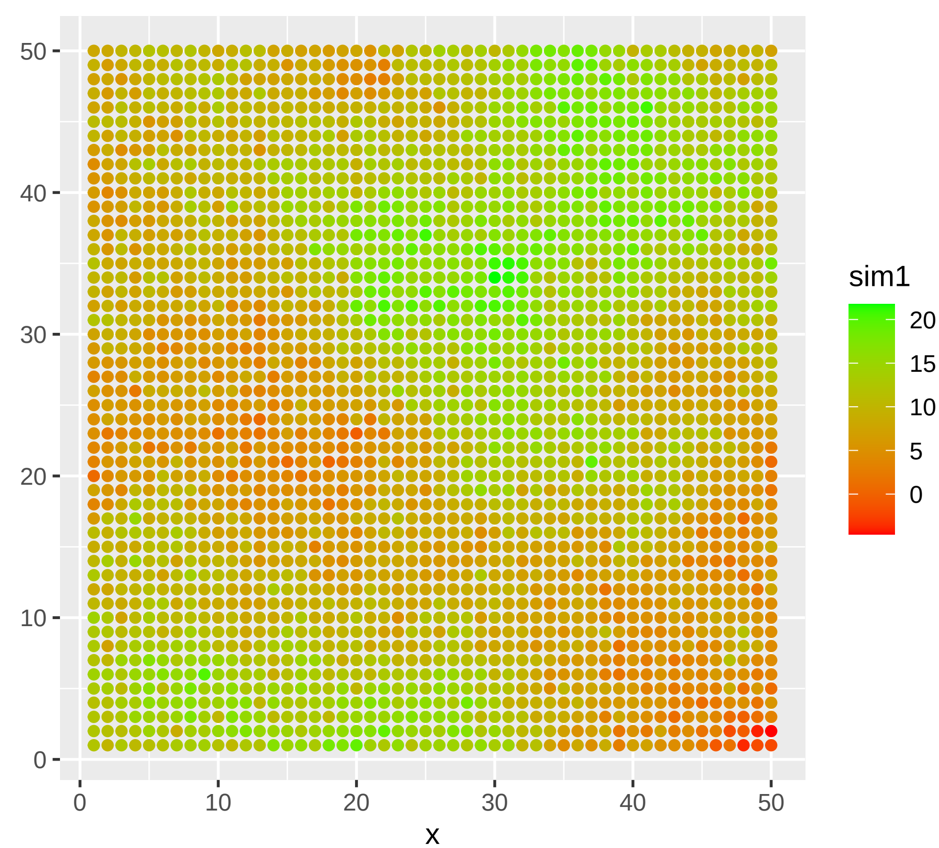
```
Field = expand.grid(1:50, 1:50)
## Set the name of the spatial coordinates within the field
names(Field)=c('x','y')

## Define the yield spatial structure inside the field
   ## Set the parameters of the semi-variogram
      Psill=25  ## Partial sill = Magnitude of variation
      Range=45  ## Maximal distance of autocorrelation
```

```
    Nugget=2  ## Small-scale variations
  ## Set the semi-variogram model
    Beta=7   ## mean value of the field
    Field_modelling=
      gstat(formula=z~1, ## We assume that  a constant trend in the data
            locations=~x+y,
            dummy=T,    ##  set to True for unconditional simulation
            beta=Beta,  ##  set the average value over the field
            model=vgm(psill=Psill,
            range=Range ,
            nugget=Nugget,
            model='Sph'), ## Spherical semi-variogram model
            nmax=40) ## number of nearest observations used for  new prediction

  gaussian_field=predict(Field_modelling, newdata=Field, nsim=1)
```

```
## [using unconditional Gaussian simulation]
```

We can examine the patterns. plot the observations as points with a colored yield gradient.

```
plot.field=
ggplot()+  ## Initialize the ggplot layer
geom_point(data=gaussian_field,aes(x=x,y=y,col=sim1))+
scale_colour_gradient(low="red",high="green") +ylab(NULL)

plot.field
```



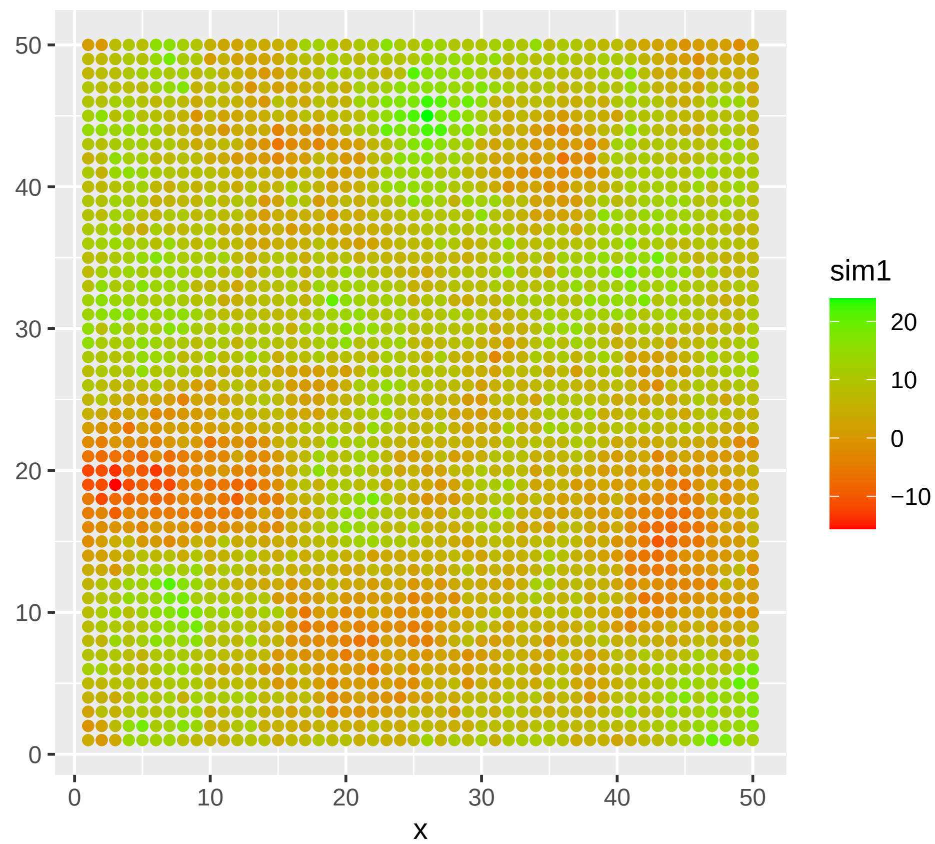
Let's now make it a little more patchy, defining shorter 'range' of spatial correlation

```
Field2 = expand.grid(1:50, 1:50)
## Set the name of the spatial coordinates within the field
names(Field2)=c('x','y')

## Define the yield spatial structure inside the field
   ## Set the parameters of the semi-variogram
      Psill=25  ## Partial sill = Magnitude of variation
      Range=10  ## Maximal distance of autocorrelation
      Nugget=2  ## Small-scale variations
   ## Set the semi-variogram model
      Beta=7   ##
      Field_modelling2=
         gstat(formula=z~1, ## We assume a constant trend in the data
         locations=~x+y,
         dummy=T,    ##  set to True for unconditional simulation
         beta=Beta,  ## Naverage value
         model=vgm(psill=Psill,
         range=Range ,
         nugget=Nugget,
         model='Sph'), ## Spherical semi-variogram model
         nmax=40) ## number of nearest observations for each new prediction

  gaussian_field2=predict(Field_modelling2, newdata=Field2, nsim=1)
```

```
## [using unconditional Gaussian simulation]
```

We can see that this is a more patchy pattern.

```
plot.field2=
ggplot()+  ## Initialize the ggplot layer
geom_point(data=gaussian_field2,aes(x=x,y=y,col=sim1))+
scale_colour_gradient(low="red",high="green") +ylab(NULL)

plot.field2
```



Now we can examine the semivariogram pattern for each field.

We will first randomly 'sample' 350 locations for each field, then we plot the empirical semi-variogram. Sampling is needed to 1) save time in computing variograms, and 2) simulate a sampling process that generate the data.

Subset 350 sites from Field; build variogram.

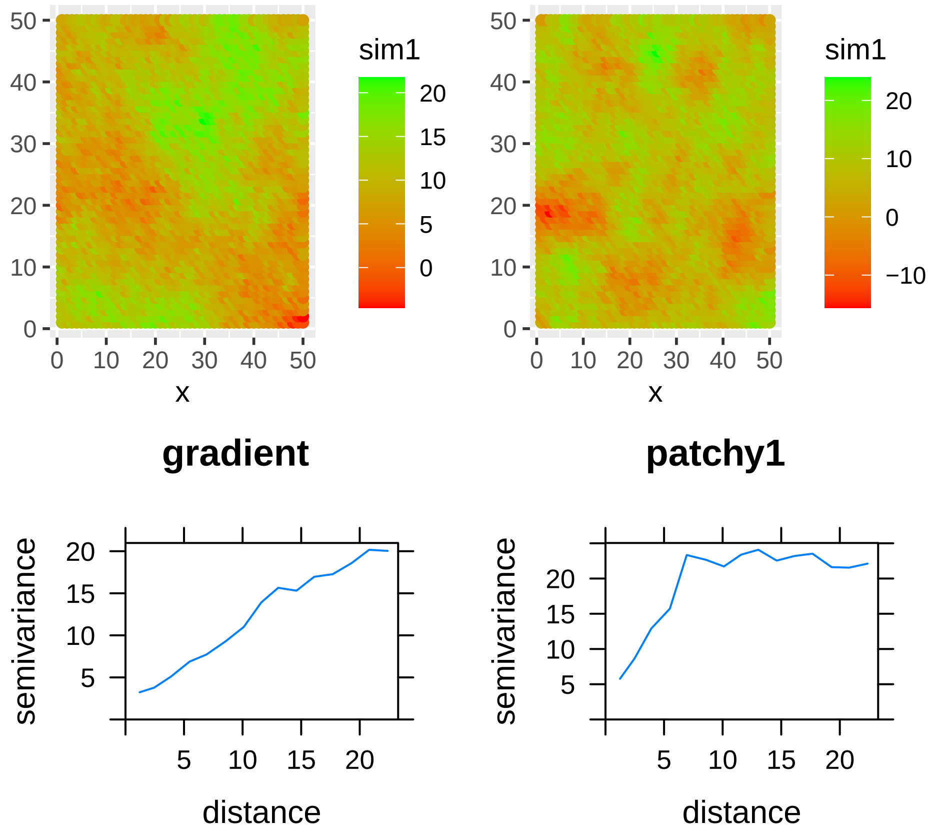
```
d=gaussian_field[sample(1:2500, 250),]

coordinates(d)=~x+y

vario1=plot(gstat::variogram(sim1~1, d), type='l', main='gradient')
```

Subset 350 sites from Field2; build variogram.

```
d2=gaussian_field2[sample(1:2500, 250),]

coordinates(d2)=~x+y

vario2=plot(gstat::variogram(sim1~1, d2), type='l', main='patchy1')
```

Let plot the field and respective variograms.

```
plot_grid(plot.field, plot.field2,  vario1, vario2,  nrow = 2, ncol = 2 )
```



We see that, as expected, the range is larger for the first (gradient) landscape. The variogram 'rises' more gradually. The patchy landscape has shorter gradient (data are independent beyond ~ 7), and the variogram has a more irregular shape.

Excercises you can try yourself:

- variogram assumes **isotropy**. That is, the strength and pattern of autocorrelation is the same in all directions. Try to plot variograms of the landscapes showing correlation over different directions. *Hint*: use "alpha=c(0,45,90…etc)" inside the variogram function to provide the *angles* of directions.