StatCourse_spatial

Stefano Larsen

4/4/2021

Exploring autocorrelation [in time & space]

This script provides practical examples of how to detect and account for temporal and spatial autocorrelation in statistical models. We will do this by first simulating some autocorrelated data, and then model them. We will be using the **gls** function, but note that there are multiple ways for dealing with autocorrelation.

Load packages

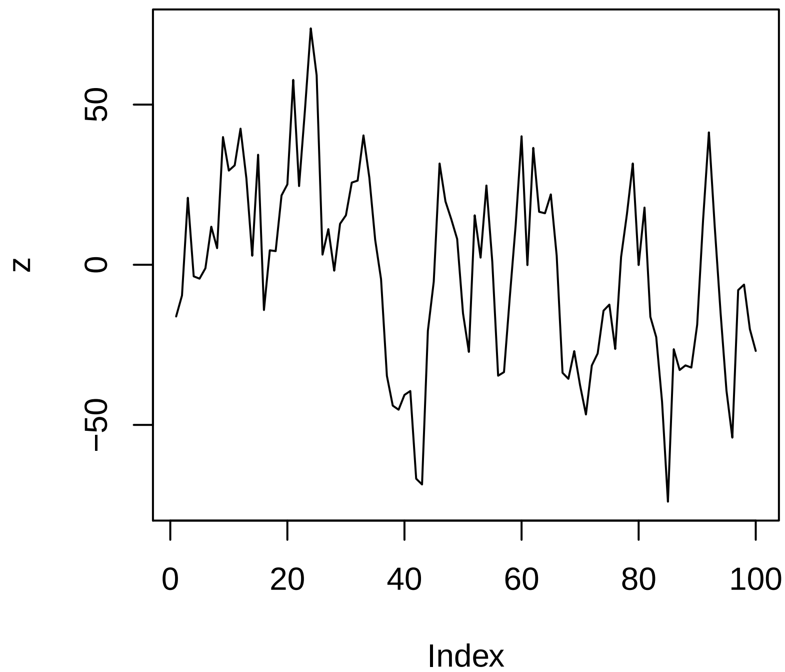
```
library(vegan)
library(tidyverse)
library(gstat)
library(sp)
#library(raster)
library(maptools)
library(nlme)
library(ncf)
```

Example with temporal autocorrelation (time-series)

Simulate some correlated noise with AR(1)

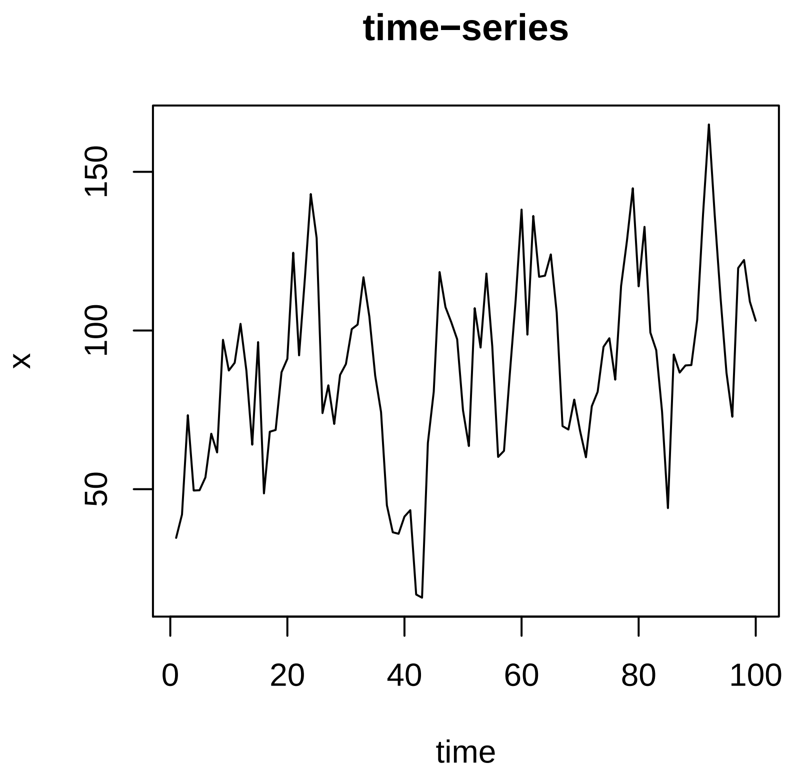
```
set.seed(2)
z<-w<-rnorm(100, sd=18)# noise; random normal with sd
for (t in 2:100) z[t]<- 0.8 * z[t-1] + w[t]# the AR(1) process, with cor=0.8

plot(z, type='l')
```



Let's simulate a time series using the autocorrelated noise.

```
Time <- 1:100
x <- 50 + 0.8 * Time + z # simulate time-series with autocor and noise and positive trend
plot(x, xlab="time", type="l", main='time-series')
```



There is apparent trend. Let's examine it. Fit a linear model to the data (ignoring autocorrelation).

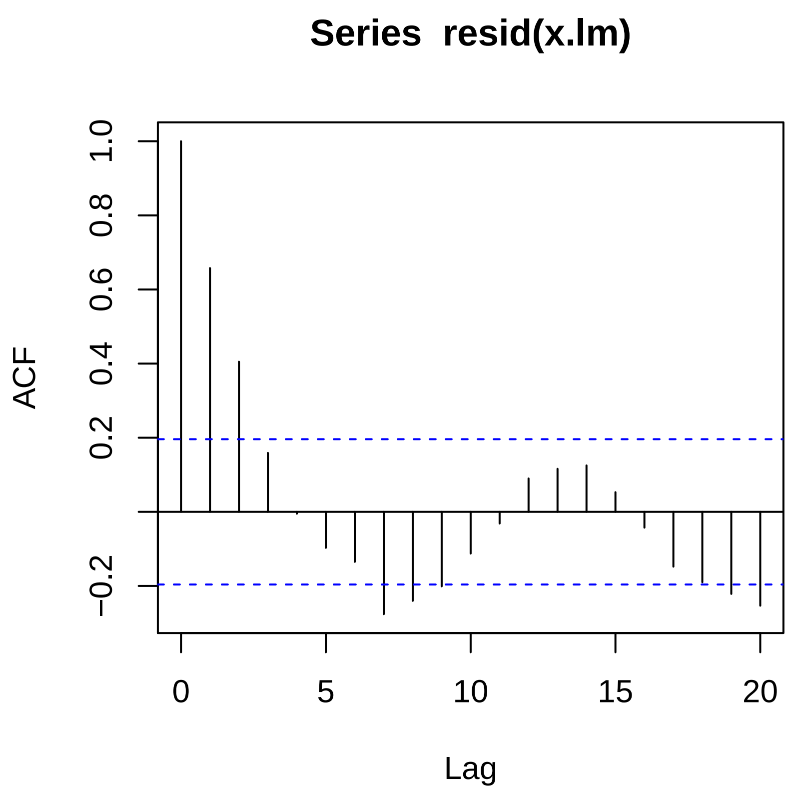
```
x.lm<-lm(x~Time)
summary(x.lm) # estimate of slope = 0.38
```

```
##
## Call:
## lm(formula = x ~ Time)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -70.013 -18.180   0.558  18.119  64.486
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.27189    5.52556  12.537  < 2e-16 ***
## Time         0.38499    0.09499   4.053 0.000101 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.42 on 98 degrees of freedom
## Multiple R-squared:  0.1435, Adjusted R-squared:  0.1348
## F-statistic: 16.43 on 1 and 98 DF,  p-value: 0.0001015
```

```
# sd error of estimates: intercept=5.5 and slope=0.09
```

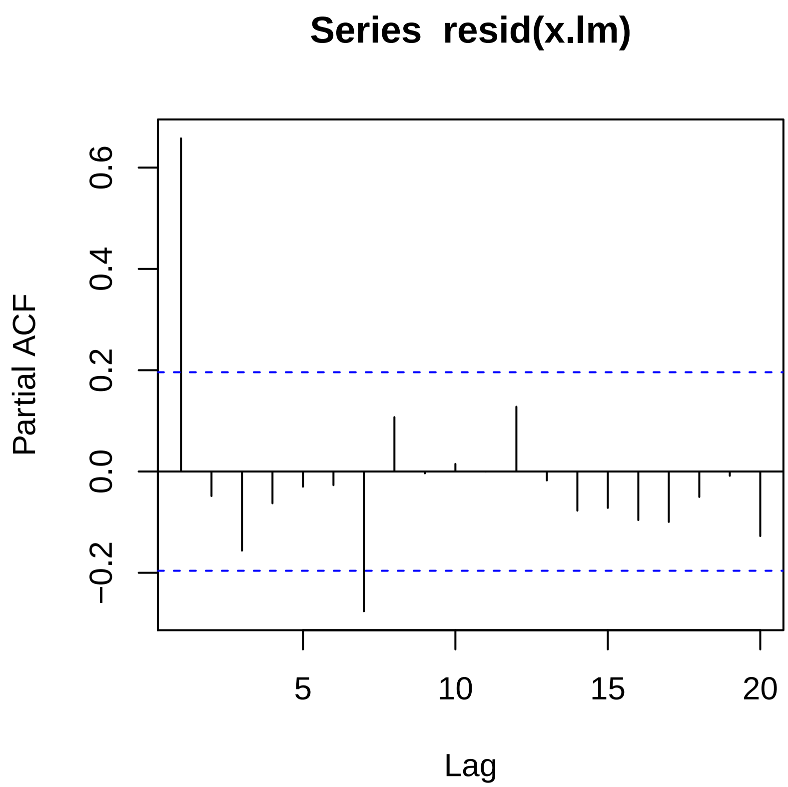
Linear regression assumes independence of observations and residuals. Let's check if this holds here. Plot autocorrelation function for model residuals. This plot shows the correlation between observations' residuals at increasing lag-distance.

```
acf(resid(x.lm))
```



Let see the *partial ACF* of residuals plot. Te partial ACF shows the correlation between observations' residuals at a given lag-distance, after accounting for the effects of other lags.

```
pacf(resid(x.lm)) # partial acf shows lag-1 autocorrelation (as expected)
```



This autocorrelation means that our estimation of the coefficients for timeseries trends can be biased

Fitting a generalised least-square model (GLS), taking AR into account.

The new bit is "cor=corAR(1)', meaning we are assuming autocor of order 1 (at lag=1). What happens to the standard errors of the estimated parameters? The temporal trend is now borderline significant and the SD of parameters larger

```
#library(nlme)
x.gls <- gls(x~Time, cor=corAR1(0.7)) # fitting autocorrelation of order 1
#(Here using cor=0.7 because we knew already)
summary(x.gls)
```

```
## Generalized least squares fit by REML
##   Model: x ~ Time
##   Data: NULL
##        AIC      BIC    logLik
##   890.3855 900.7254 -441.1928
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
```

```
## 0.6975682
##
## Coefficients:
##                Value Std.Error  t-value p-value
## (Intercept) 66.44187 12.931722 5.137898  0.0000
## Time         0.42370  0.219782 1.927822  0.0568
##
##  Correlation:
##      (Intr)
## Time -0.858
##
## Standardized residuals:
##          Min           Q1          Med           Q3          Max
## -2.4030369405 -0.6186876673  0.0006914511  0.6725653520  2.3171444898
##
## Residual standard error: 28.65019
## Degrees of freedom: 100 total; 98 residual
```

```
# the st errors of parameters is much larger now: intercept=12.9, slope= 0.21
```

Which is the "best" model? We can use the Information Criterion approach (AIC)

```
AIC(x.lm)
```

```
## [1] 950.0281
```

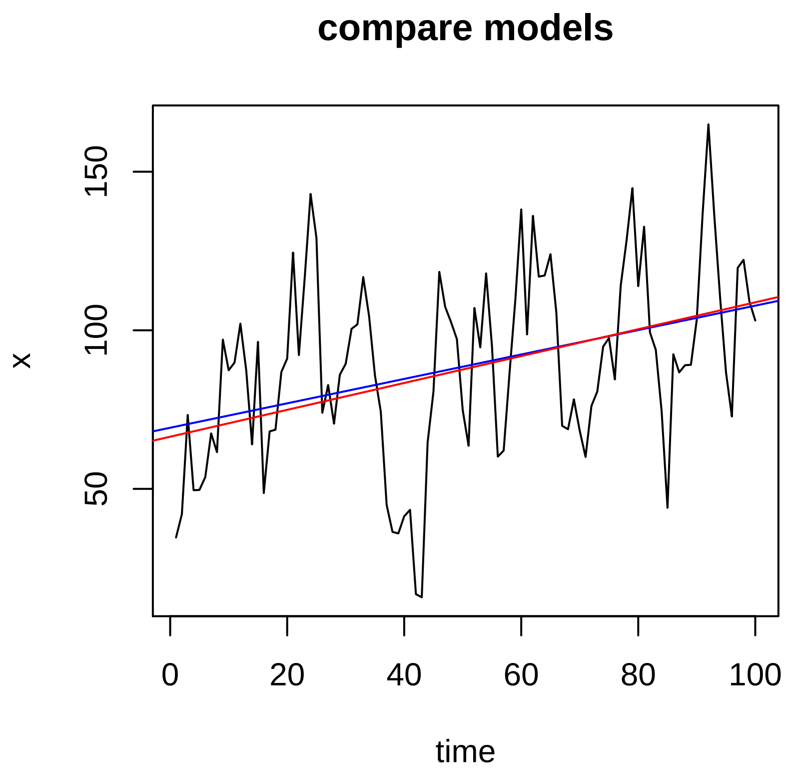
```
AIC(x.gls)# better suppored model (i.e. lower AIC)
```

```
## [1] 890.3855
```

Plot both model prediction.

Trends are only marginally different, but the confidence intervals are larger when accounting for temporal autocorrelation. Hence the trends become non-significant at alpha=0.05.

```
plot(x, xlab="time", type="l", main='compare models')
abline(a=coef(x.lm)[1],b=coef(x.lm)[2], col='blue' )
abline(a=coef(x.gls)[1],b=coef(x.gls)[2], col='red' )
```



Spatial Autocorrelation

Spatial structure is more complex than temporal; there are two dimensions. This is where the Variogram can help you.

Let's simulate some sparse coordinates [with no real meaning!]

```
set.seed(2)
Long.x=seq(from=100, to=120, 0.2)+rnorm(101, 0, 4)
Lat.y=seq(20,40, 0.2)+rnorm(101, 0, 4)
```

Simulate some spatially structured variable (e.g. bird diversity) using the **ncf::rmvr.spa** function. This function actually uses variogram parameters for simulating data (here p='range', and 'nugget').

```
#install.packages('ncf')
#library(ncf)
set.seed(2)
div.ncf=rmvn.spa(Long.x, Lat.y, p=12, method = "exp", nugget = 2)# we call it div.nfc

div.ncf=div.ncf-min(div.ncf) # dirty trick to include only positive values
```

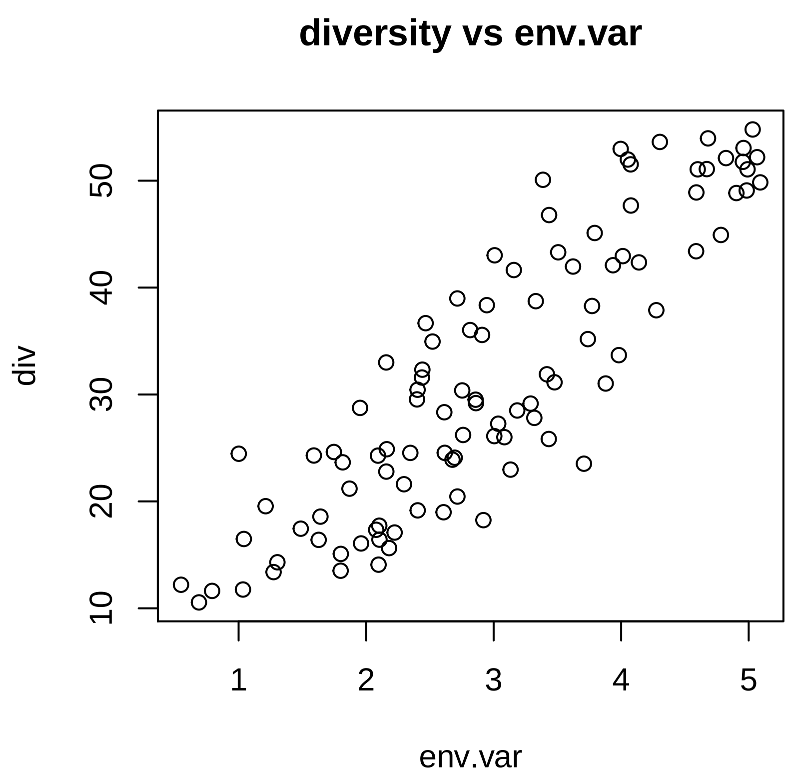
Now simulate an environmental variable

```
set.seed(2)
env.var=rnorm(101,3,1)# random environmental variable with mean= 3 and sd=1
```

And now make the simulated diversity value respond to this environmental variable We will add spatially structured noise by adding *'div.ncf'*

```
div=10+0.8*env.var+3.5*div.ncf

plot(div~ env.var, main='diversity vs env.var')
```



Put the data together, coordinates, environmental variable and diversity.

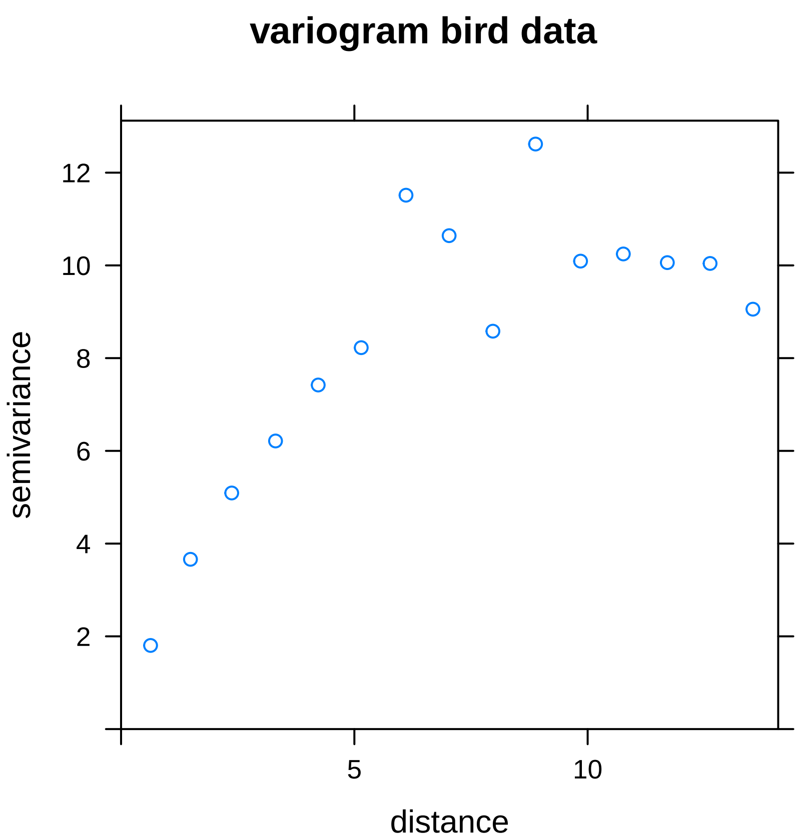
```
bird.diversity=data.frame(Lat.y=Lat.y, Long.x=Long.x, Bird_div=div, Env.var=env.var)
```

Specify what are the coordinates, for plotting variograms.

```
coordinates(bird.diversity)=~Long.x+Lat.y
```

##Lets have a look at the variogram to examine spatial patterns. We can use the *gstat::variogram* function to examine spatial patterns.

```
plot(gstat::variogram(Bird_div~ Long.x+Lat.y , bird.diversity), main='variogram bird data')
```




```
#Looks like there is strong autocorrelation pattern.
#with a range around dist=7
```

Fit a simple model with no spatial correlation.

```
model1 <- gls(Bird_div ~env.var , data = bird.diversity )
summary(model1)
```

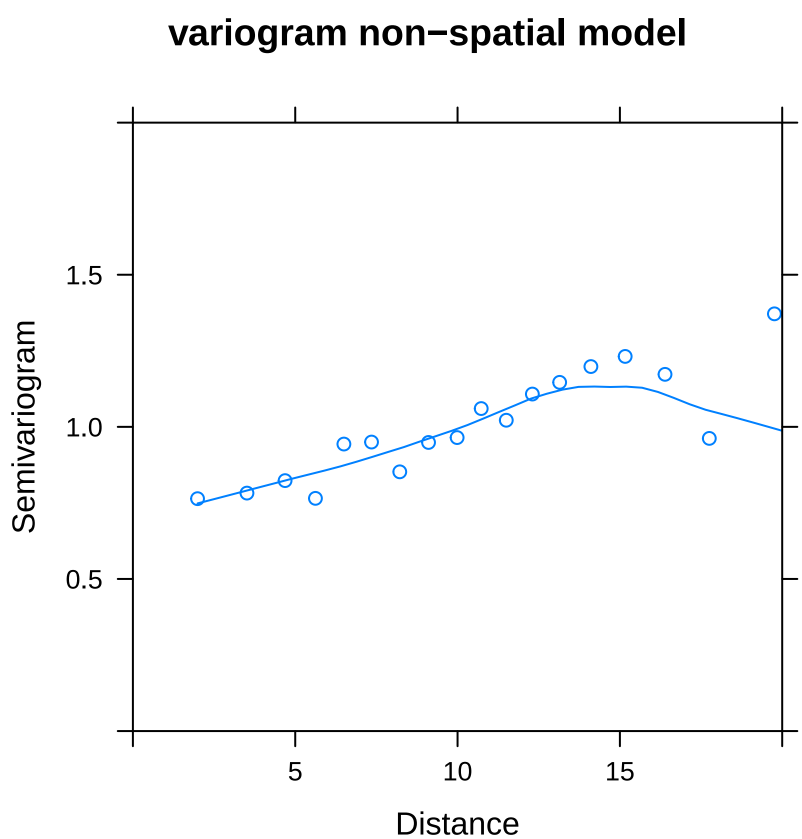
```
## Generalized least squares fit by REML
##   Model: Bird_div ~ env.var
##   Data: bird.diversity
##         AIC      BIC    logLik
##   658.0992 665.8845 -326.0496
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) 2.717601  1.711910  1.587467  0.1156
## env.var     9.748711  0.535673 18.198995  0.0000
##
##  Correlation:
##         (Intr)
## env.var -0.933
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
```

```
## -2.46972936 -0.81259701  0.01427407  0.72316927  2.31016609
##
## Residual standard error: 6.211664
## Degrees of freedom: 101 total; 99 residual
```

Check the residual variogram from the model.

We use the *nlme::Variogram* function that works directly on model residuals Some clear pattern of correlated observation at short distances.

```
plot(nlme::Variogram(model1, form = ~Long.x + Lat.y, resType = "normalized"),
     ylim=c(0,2), xlim=c(0,20), main='variogram non-spatial model')
```




```
# nearby locations show rather low variation. A clear patter of spatial autocorrelation
```

Now, lets fit a *gls model* including Gaussian autocorrelation function.

The new bit: "correlation = corGauss (. . .)". *You can check GLS help to see other correlation structures.* NOTE: convergence problems can occur.

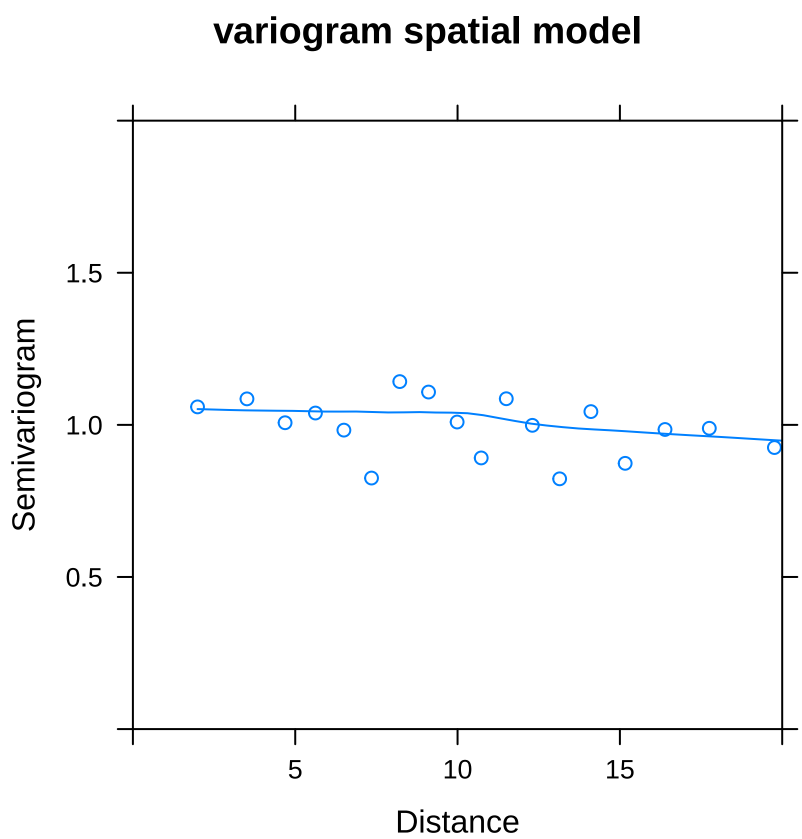
```
model.sp.gauss<-gls( Bird_div~ Env.var ,
                     correlation = corGaus(form = ~Long.x + Lat.y, nugget=T),
                     data = bird.diversity )
summary(model.sp.gauss)
```

```
## Generalized least squares fit by REML
##   Model: Bird_div ~ Env.var
##   Data: bird.diversity
##        AIC      BIC    logLik
##   241.0007 253.9763 -115.5003
##
## Correlation Structure: Gaussian spatial correlation
##  Formula: ~Long.x + Lat.y
##  Parameter estimate(s):
##       range      nugget
## 7.645656904 0.001035152
##
## Coefficients:
##                 Value Std.Error  t-value p-value
## (Intercept) 23.672597 2.8802403  8.21897       0
## Env.var      2.166373 0.0475724 45.53846       0
##
##  Correlation:
##         (Intr)
## Env.var -0.031
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -1.83718976 -0.88556764 -0.04413115  1.32986896  2.59488358
##
## Residual standard error: 7.954504
## Degrees of freedom: 101 total; 99 residual
```

```
# the correlation was well identified with a range=~ dist 7
```

And the check the residual variogram of this spatially explicit model.

```
plot(nlme::Variogram(model.sp.gauss, form = ~Long.x + Lat.y, resType = "normalized"),
     ylim=c(0,2), xlim=c(0,20), main='variogram spatial model')
```




```
#The patterns are different, and no trend with distance is evident now.
#The spatial model took care of the residual structure
```

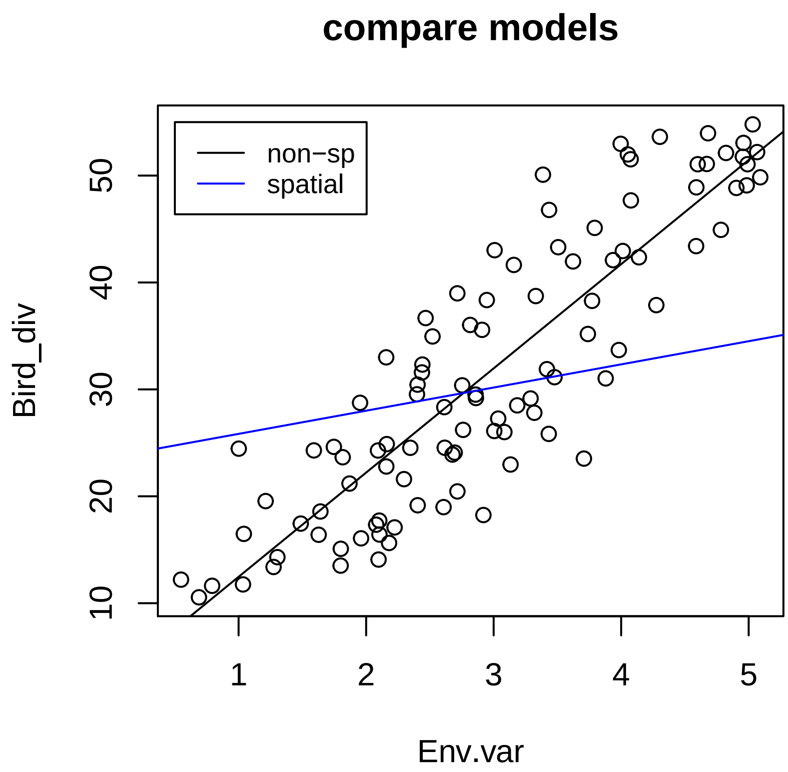
Which model is favored according to AIC. Clear support for model2 (lower AIC).

```
AIC(model1, model.sp.gauss)
```

```
##                df      AIC
## model1          3 658.0992
## model.sp.gauss  5 241.0007
```

We can visually compare the fits of the two model. Most of the apparent effect of environmental variable on bird diversity seems to come from an autocorrelation process.

```
plot(Bird_div~Env.var, bird.diversity, main='compare models')
# add fit from model1 (w/o spatial autocor
abline(a=coef(model1)[1], b=coef(model1)[2])
#add the model fit from the gaussian autocor model
abline(a=coef(model.sp.gauss)[1], b=coef(model.sp.gauss)[2], col="blue")
legend(0.5, 55, legend=c("non-sp", "spatial"),
       col=c("black", "blue"), lty=1:1, cex=0.8)
```



This was just a rather extreme simulated example. You can play with your data if these have coordinates, that is spatially-explicit data.

Excercises you can also try yourself:

- What happens if you simulate bird.div ('div') without adding the spatially structured noise (i.e. adding 'rnorm ()' instead of 'div.nfc')?
- How does the variogram look like in this case?
- Would the use spatial gls model still be justified?

TIP

To simulate another diversity variable related to *env.var*, but without the spatial structure try e.g.

```
div_nsp=10+0.8*env.var+rnorm(10,5) # here called as diversity non spatial (nsp)
```

You can then add this variable to the *bird.diversity* dataframe created previously. You can explore patterns in the variogram as done in line ~170.

TIP

To add the newly created variable into the *bird.diversity* dataframe, simply:

```
bird.diversity$div.nsp
```

```
## NULL
```