

Spatial autocorrelation

Roberto Molowny-Horas (CREAF-EMF)

March 16th-17th, 2022

Point data

We first retrieve point data for the Eurasian jay.

```
suppressPackageStartupMessages(library(sf))
pxy <- st_read("Eurasian_jay.gpkg", quiet = T)
```

Transverse Mercator projection to be able to work in meters.

```
pxy <- st_transform(pxy, "epsg:3109")
```

Ripley' K and L and Nearest Neighbour Distance functions.

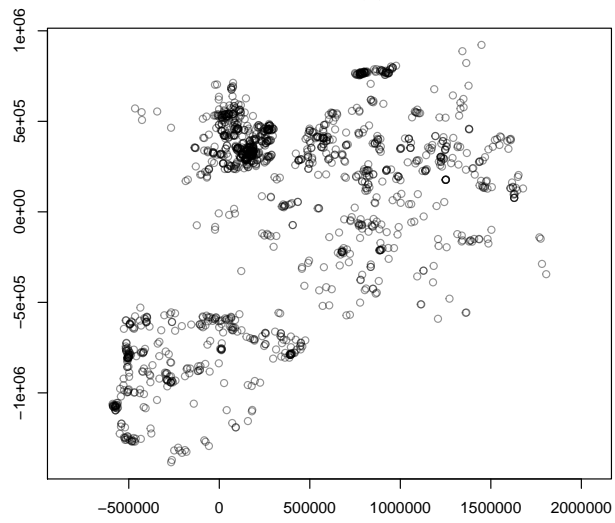
```
suppressPackageStartupMessages(library(spatstat))
suppressPackageStartupMessages(library(terra))
p <- as.ppp(st_coordinates(pxy), as.vector(ext(pxy)))
```

```
K <- Kest(p, correction="best")
L <- Lest(p, correction="best")
G <- Gest(p, correction="best")
```

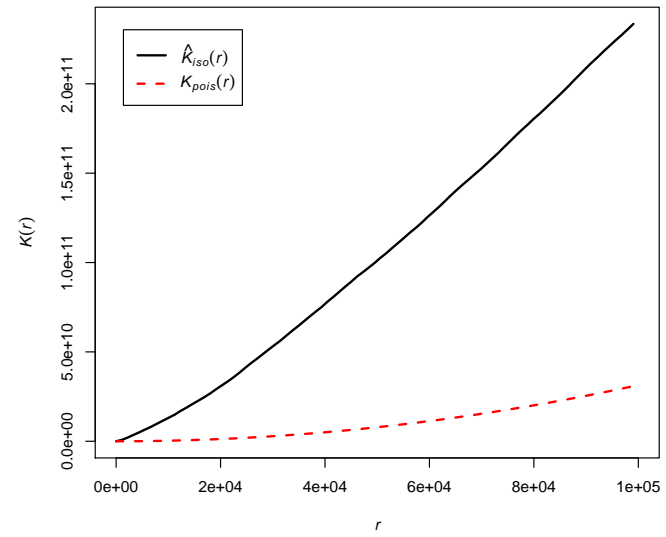
We will have a look at plots of those functions.

```
par(mfrow=c(2,2))
plot(p, axes=T, show.window=F, main="European jay")
plot(K, main="K function", xlim=c(0,1e5), lwd=2)
plot(L, ylab="L(r)", main="L function", lwd=2)
plot(G, cbind(km,theo) ~ r, main="G function", ylim=c(0,1), col=c("black","red"), lwd=2)
par(mfcol=c(1,1))
```

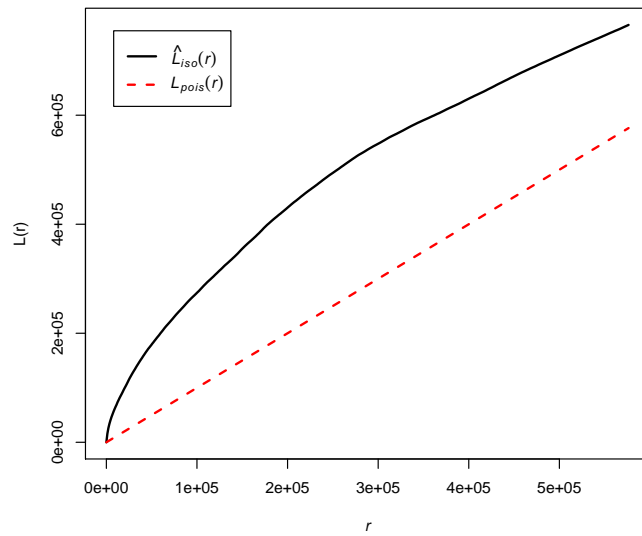
European jay



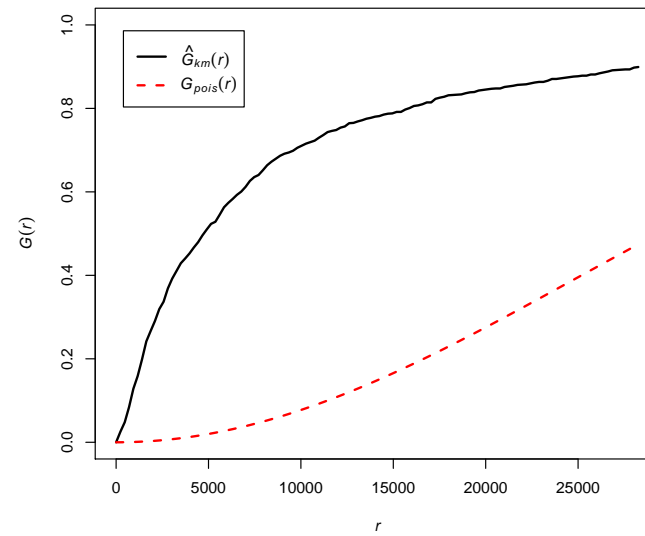
K function



L function



G function



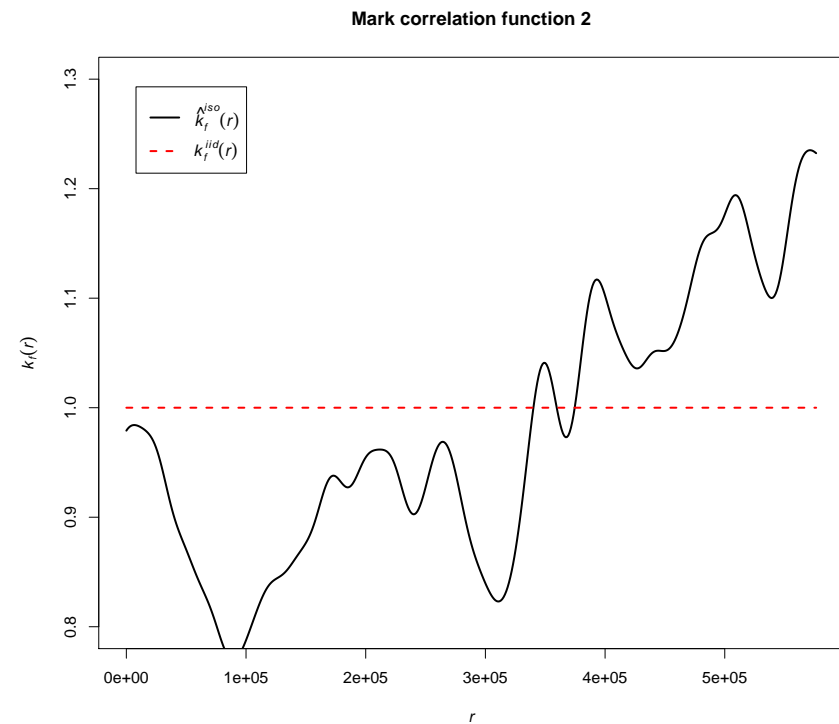
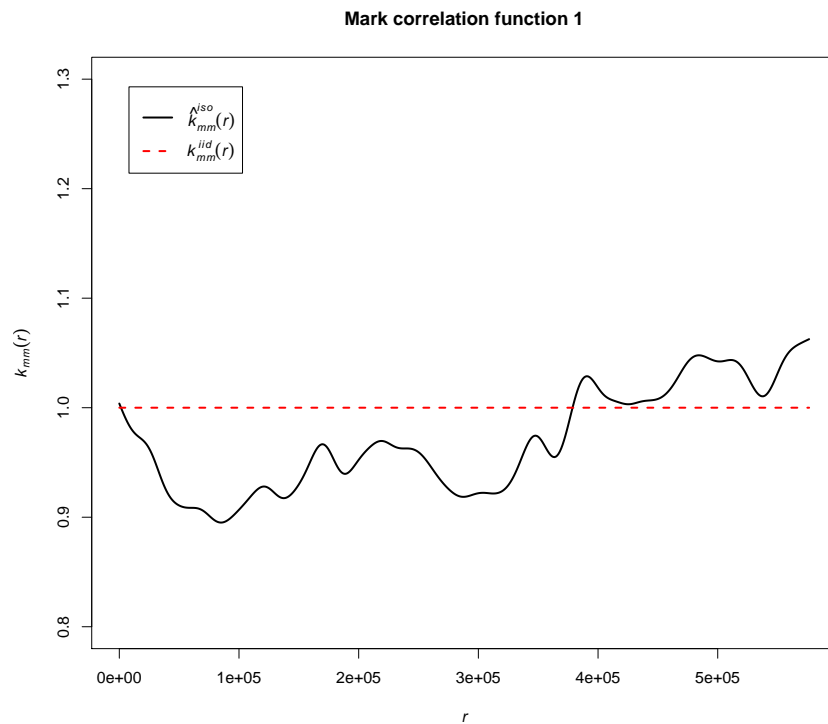
These results tell us about clustering/dispersal of the point pattern. Next, we will analyze the abundance values of the European jay observations at those locations.

```
dat <- data.frame(st_coordinates(pxy), st_drop_geometry(pxy))
pv <- as.ppp(dat, as.vector(ext(pxy)))
M1 <- markcorr(pv, correction="iso")
M2 <- markcorr(pv, correction="iso", f=function(x1,x2) (x1-x2)^2) # Similar to a variogram.
# e1 <- spatstat::envelope(pv, fun=markcorr, nsim=100, nrank=5, verbose=F, silent=T)
# e2 <- spatstat::envelope(pv, fun=markcorr, nsim=100, nrank=5, f=function(x1,x2) abs(x1-x2), verbose=F, silent=T)
print(M1)
```

```
## Function value object (class 'fv')
## for the function r -> k[mm](r)
## .....
##      Math.label      Description
## r      r            distance argument r
## theo {k[mm]^{iid}}(r) theoretical value (independent marks) for k[mm](r)
## iso  {hat(k)[mm]^{iso}}(r) Ripley isotropic correction estimate of k[mm](r)
## .....
## Default plot formula: .~r
## where "." stands for 'iso', 'theo'
## Recommended range of argument r: [0, 576360]
## Available range of argument r: [0, 576360]
```

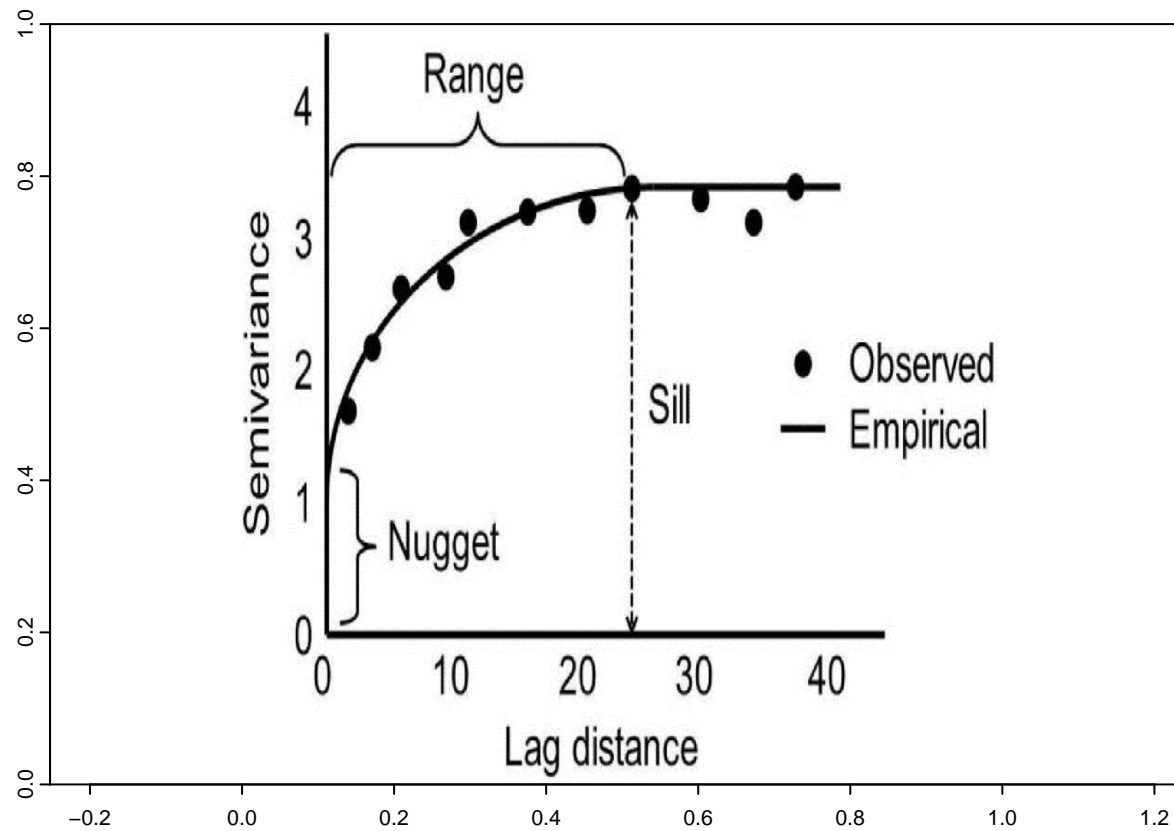
A value of 1 indicates that there is no correlation. Below/above 1 there is indication of clustering/dispersion. Don't forget to check function `spatstat::envelope` to calculate simulation envelopes (slow!).

```
par(mfcol=c(1,2))
plot(M1,main="Mark correlation function 1",ylim=c(.8,1.3),lwd=2)
plot(M2,main="Mark correlation function 2",ylim=c(.8,1.3),lwd=2)
par(mfcol=c(1,1))
```



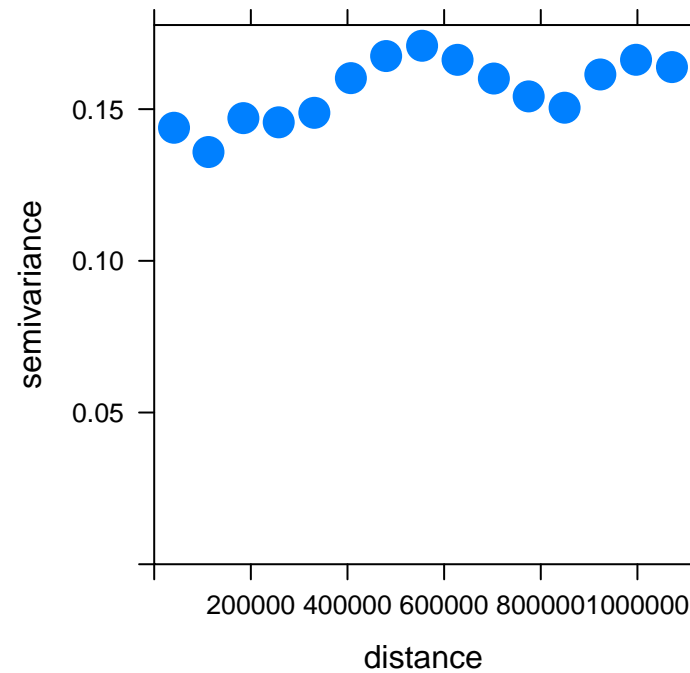
Variograms can be defined as the variance between the values of a variable at points separated by a distance h . There are three parameters to take into account:

1. Nugget: the value at the origin.
2. Sill: limit of the variogram at infinite.
3. Range: distance at which the difference with sill is negligible.



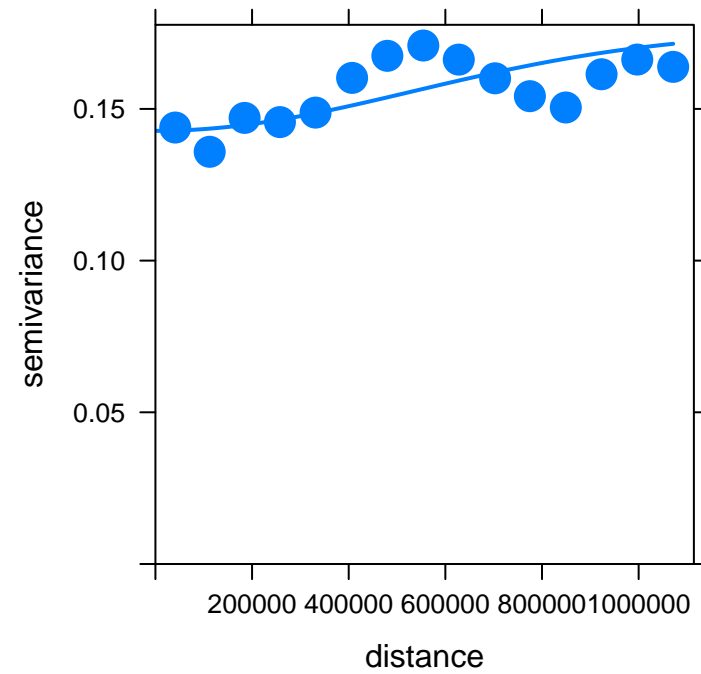
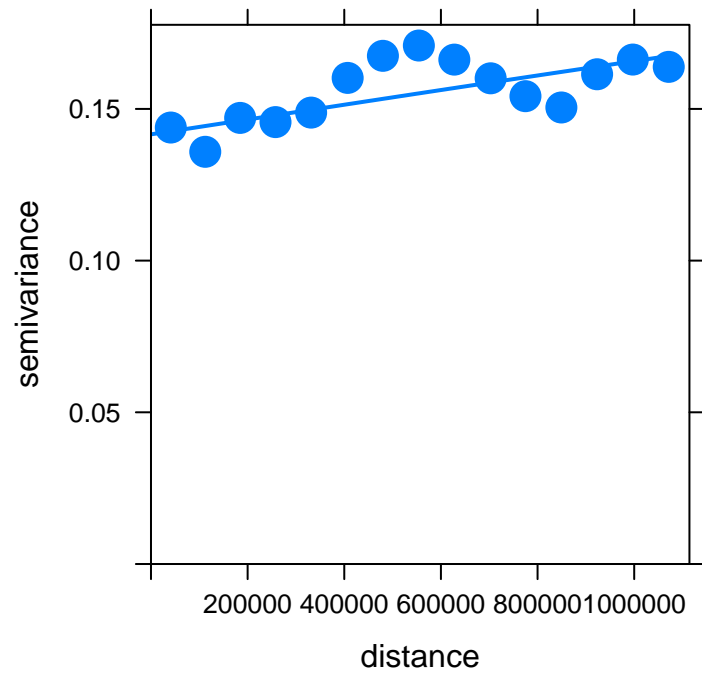
We first compute the variogram of the 'individual' variable, which determines the abundance of the European jay at each location.

```
suppressPackageStartupMessages(library(gstat))  
v <- variogram(sqrt(individualCount)~1,locations=~X+Y,data=dat)  
plot(v,cex=2,pch=16)
```



This is a very weird dataset, but we can try to fit a variogram nevertheless.

```
fitv1 <- fit.variogram(v, vgm(model="Sph",nugget=1.5,psill=1.8))
fitv2 <- fit.variogram(v, vgm(model="Gau",nugget=1.5,psill=1.8))
par(mfcol=c(1,2))
plot(v,fitv1,cex=2,pch=16,lwd=2)
plot(v,fitv2,cex=2,pch=16,lwd=2)
par(mfcol=c(1,1))
```



Raster data

First, we get two climatic raster from worldclim.org.

```
t_mean <- rast("Annual mean temperature.tif")
p_total <- rast("Annual total precipitation.tif")
```

Indices: Moran' I (-1,1) and Geary'G (close to 0 is positive; larger than 2, negative)

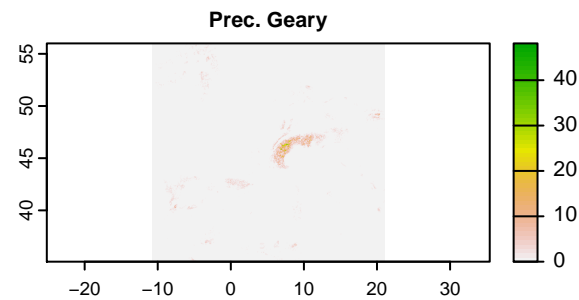
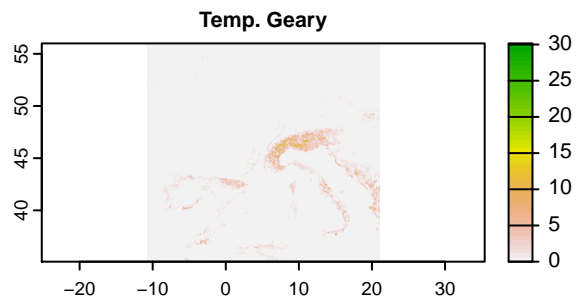
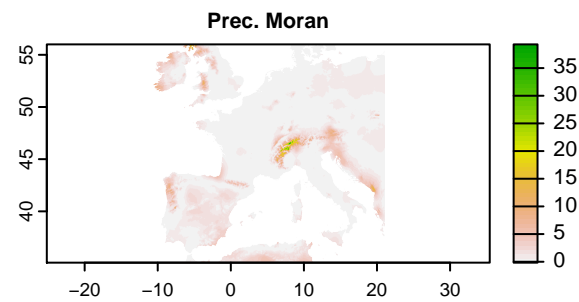
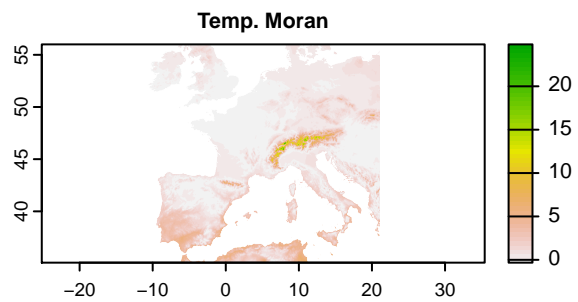
```
w <- matrix(rep(1,9),3) # Actually, the default in "autocor".
w[2,2] <- 0
t_moran <- autocor(t_mean, w, method="moran")
p_moran <- autocor(p_total, w, method="moran")
t_geary <- autocor(t_mean, w, method="geary")
p_geary <- autocor(p_total, w, method="geary")
cat(c(t_moran,p_moran,t_geary,p_geary))
```

```
## 0.9968936 0.9968936 0.02493223 0.02493223
```

The two indices tell us that spatial autocorrelation strongly affects both climatic variables, which is not surprising since we know that they have been derived via interpolation from weather stations. We can also derive a local Moran'I and Geary'G indices by specifying it in the "autocor" function.

```
t_moran_local <- setNames(autocor(t_mean,w,method="moran",global=F),"t_moran_local")
p_moran_local <- setNames(autocor(p_total,w,method="moran",global=F),"p_moran_local")
t_geary_local <- setNames(autocor(t_mean,w,method="geary",global=F),"t_geary_local")
p_geary_local <- setNames(autocor(p_total,w,method="geary",global=F),"p_geary_local")
```

```
par(mfrow=c(2,2))
plot(t_moran_local,main="Temp. Moran")
plot(p_moran_local,main="Prec. Moran")
plot(t_geary_local,main="Temp. Geary")
plot(p_geary_local,main="Prec. Geary")
par(mfcol=c(1,1))
```



Spatial correlation is higher mountainous regions probably because it is in those regions where weather stations are scarcer and, therefore, interpolation is used more heavily to derive climatic variables.

Mantel tests and correlograms.

We have left out on purpose Mantel tests and correlograms. We have seen conflicting conclusions by several authors arguing for (e.g. Borcard and Legendre 2012, *Ecology* 93(6), 1473-1481) and against (e.g. Guillot and Rousset 2013, *Method in Ecology and Evolution* 4(4), 336-344) the use of the Mantel test and the Mantel correlogram. In any case, you can find further information on the Mantel tests in several spatial analysis books and articles (e.g. Dale and Fortin 2014: “Spatial Analysis: A Guide for Ecologists”).