GLS template

Zofie Cimburova

19 juni 2017

## Libraries

library(gstat)  
library(sp)  
library(nlme)   
library(ggplot2)  
library(grid)  
library(gridExtra)  
library(raster)  
library(rgdal)   
library(car)

# 1. Import observations

Load observation points on the forest line and only keep those observation points situated within a limit vertical distance of 15 m from a forest pixel of local maximum altitude:

forest\_line <- read.csv("explanatory\_variables.csv")  
forest\_line <- forest\_line[forest\_line$lim<=15,]

* Split observation points to one dataset for prediction and one dataset for test purposes:

obs <- forest\_line[seq(2,nrow(forest\_line),2),]  
test<- forest\_line[seq(1,nrow(forest\_line),2),]

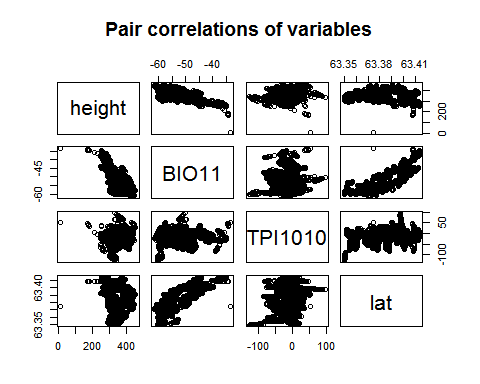
* Import rasters of explanatory variables.

bio11 <- raster("BIO11.tif", band=1)  
projection(bio11) <- "+proj=utm +zone=33 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no\_defs"  
  
r.explanatory <- brick(bio11,  
 raster("TPI1010.tif", band=1),  
 raster("lat.tif", band=1))  
names(r.explanatory) <- c('BIO11','TPI1010','lat')

* Correlations between variables

cor(obs[c("height", "BIO11", "TPI1010", "lat")])

## height BIO11 TPI1010 lat  
## height 1.00000000 -0.60429697 0.21071564 -0.06073882  
## BIO11 -0.60429697 1.00000000 -0.08022841 0.79967797  
## TPI1010 0.21071564 -0.08022841 1.00000000 0.12683051  
## lat -0.06073882 0.79967797 0.12683051 1.00000000

* Pair plots of variables 

# 2. Fit linear model

mod<-lm(height~BIO11 + TPI1010 + lat, data=obs)  
summary(mod)

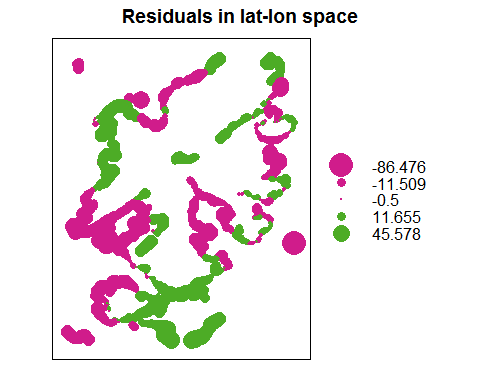
##   
## Call:  
## lm(formula = height ~ BIO11 + TPI1010 + lat, data = obs)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -86.48 -11.51 -0.50 11.65 45.58   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.962e+05 1.952e+03 -100.518 <2e-16 \*\*\*  
## BIO11 -1.155e+01 8.758e-02 -131.830 <2e-16 \*\*\*  
## TPI1010 -1.042e-01 1.093e-02 -9.536 <2e-16 \*\*\*  
## lat 3.091e+03 3.074e+01 100.582 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.76 on 2897 degrees of freedom  
## Multiple R-squared: 0.8646, Adjusted R-squared: 0.8644   
## F-statistic: 6165 on 3 and 2897 DF, p-value: < 2.2e-16

## Evaluate assumption of independence

* plot model residuals by spatial coordinates. Residuals are clearly spatially autocorrelated.

dat.ols<-obs[c("X", "Y")]  
dat.ols$resids<-data.frame(resids=resid(mod))  
coordinates(dat.ols)<-c('X','Y')

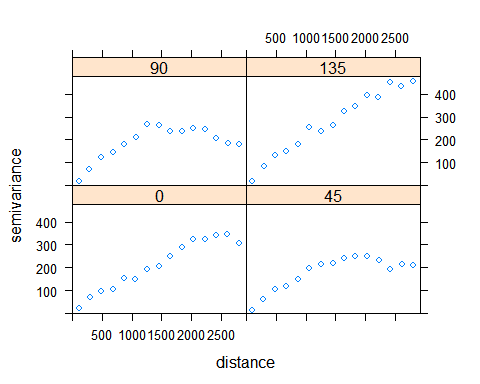
bubble(dat.ols,zcol='resids',main="Residuals in lat-lon space")



* evaluate semivariances in 4 directions. We can see that spatial autocorrelation occurs in all four directions and follows the same structure.

var.mod<-variogram(resids~1,data=dat.ols,alpha=c(0,45,90,135))

plot(var.mod)



## Evaluate multicollinearity

print(vif(mod))

## BIO11 TPI1010 lat   
## 3.058308 1.120592 3.088301

# 2. Fit generalized linear model

* Trying several variogram models to find the most optimal one (should be compared by AIC). Those models which are commented did not converge.

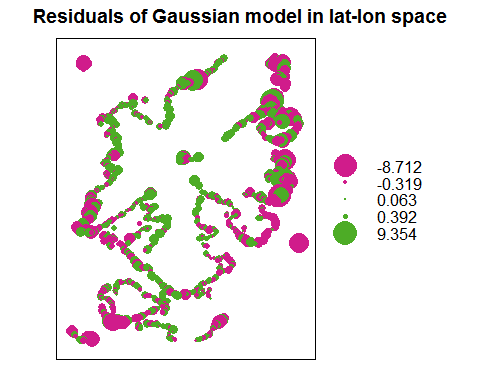
#mod.cor\_1<-gls(height~BIO11 + TPI1010 + lat, data=obs,correlation=corExp(form=~X + Y,nugget=TRUE))  
mod.cor\_2<-gls(height~BIO11 + TPI1010 + lat, data=obs,correlation=corGaus(form=~X + Y,nugget=TRUE))  
#mod.cor\_3<-gls(height~BIO11 + TPI1010 + lat, data=obs,correlation=corLin(form=~X + Y,nugget=TRUE))  
mod.cor\_4<-gls(height~BIO11 + TPI1010 + lat, data=obs,correlation=corRatio(form=~X + Y,nugget=TRUE))  
mod.cor\_5<-gls(height~BIO11 + TPI1010 + lat, data=obs,correlation=corSpher(form=~X + Y,nugget=TRUE))  
#summary(mod.cor\_1)

## Evaluate assumption of independence

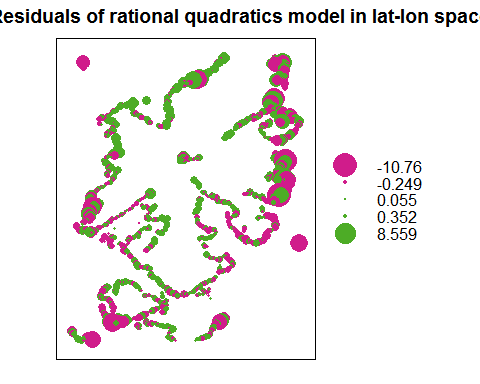
* plot model residuals by spatial coordinates. It is important to explicitly ask for NORMALIZED residuals. The residuals extracted by simple resid() are the raw or response residuals. Essentially this is the difference between the fitted values and the observed values of the response, taking into account the fixed effects terms only. These values will contain the same residual autocorrelation as that of m1 because the linear predictors are the same in the two models.
* In the plots below, the residuals of Gaussian and Rational model seem much less autocorrelated than when applying OLS. The residuals of the spherical model still show some pattern though.

#dat.gls\_1 <- obs[c("X", "Y")]  
#dat.gls\_1$resids<-data.frame(resids=resid(mod.cor\_1, type="normalized"))  
dat.gls\_2 <- obs[c("X", "Y")]  
dat.gls\_2$resids<-data.frame(resids=resid(mod.cor\_2, type="normalized"))  
#dat.gls\_3 <- obs[c("X", "Y")]  
#dat.gls\_3$resids<-data.frame(resids=resid(mod.cor\_3, type="normalized"))  
dat.gls\_4 <- obs[c("X", "Y")]  
dat.gls\_4$resids<-data.frame(resids=resid(mod.cor\_4, type="normalized"))  
dat.gls\_5 <- obs[c("X", "Y")]  
dat.gls\_5$resids<-data.frame(resids=resid(mod.cor\_5, type="normalized"))  
  
#coordinates(dat.gls\_1)<-c('X','Y')  
coordinates(dat.gls\_2)<-c('X','Y')  
#coordinates(dat.gls\_3)<-c('X','Y')  
coordinates(dat.gls\_4)<-c('X','Y')  
coordinates(dat.gls\_5)<-c('X','Y')

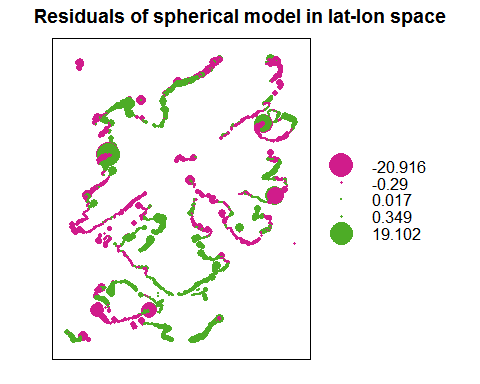
#bubble(dat.gls\_1,zcol='resids', main='Residuals of exponential model in lat-lon space')  
bubble(dat.gls\_2,zcol='resids', main='Residuals of Gaussian model in lat-lon space')



#bubble(dat.gls\_3,zcol='resids', main='Residuals of linear model in lat-lon space')  
bubble(dat.gls\_4,zcol='resids', main='Residuals of rational quadratics model in lat-lon space')



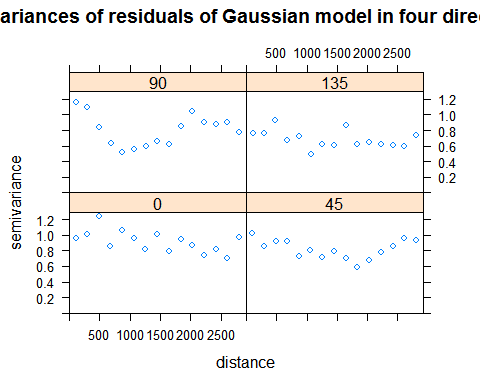
bubble(dat.gls\_5,zcol='resids', main='Residuals of spherical model in lat-lon space')



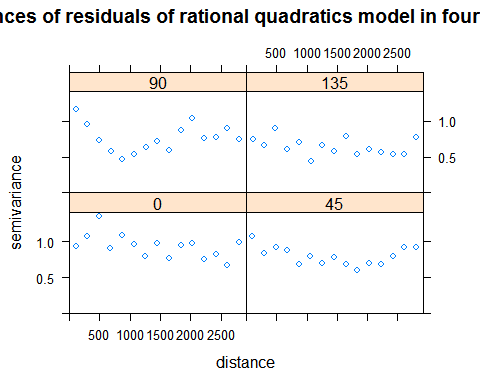
* Evaluate semivariances in 4 directions. There does not seem to be any significant structure in semivariances of all three models.

#var.mod.cor\_1<-variogram(resids~1,data=dat.gls\_1,alpha=c(0,45,90,135))  
var.mod.cor\_2<-variogram(resids~1,data=dat.gls\_2,alpha=c(0,45,90,135))  
#var.mod.cor\_3<-variogram(resids~1,data=dat.gls\_3,alpha=c(0,45,90,135))  
var.mod.cor\_4<-variogram(resids~1,data=dat.gls\_4,alpha=c(0,45,90,135))  
var.mod.cor\_5<-variogram(resids~1,data=dat.gls\_5,alpha=c(0,45,90,135))

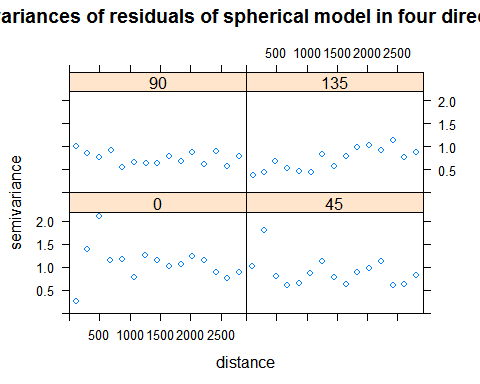
#plot(var.mod.cor\_1, main='Semivariances of residuals of exponential model in four directions')  
plot(var.mod.cor\_2, main='Semivariances of residuals of Gaussian model in four directions')



#plot(var.mod.cor\_3, main='Semivariances of residuals of linear model in four directions')  
plot(var.mod.cor\_4, main='Semivariances of residuals of rational quadratics model in four directions')



plot(var.mod.cor\_5, main='Semivariances of residuals of spherical model in four directions')



## Evaluate multicollinearity

print(vif(mod.cor\_2))

## BIO11 TPI1010 lat   
## 7.097909 7.096527 1.000717

print(vif(mod.cor\_4))

## BIO11 TPI1010 lat   
## 4.800333 4.799907 1.000209

print(vif(mod.cor\_5))

## BIO11 TPI1010 lat   
## 4.287905 4.195745 1.075870

## Use AIC to choose the best model

AIC(mod, mod.cor\_2, mod.cor\_4, mod.cor\_5)

## Warning in AIC.default(mod, mod.cor\_2, mod.cor\_4, mod.cor\_5): models are  
## not all fitted to the same number of observations

## df AIC  
## mod 5 24237.370  
## mod.cor\_2 7 -3162.850  
## mod.cor\_4 7 -4824.826  
## mod.cor\_5 7 7081.155

print(AIC)

## function (object, ..., k = 2)   
## UseMethod("AIC")  
## <bytecode: 0x0000000014701d20>  
## <environment: namespace:stats>

## Predict the whole area

newdata <- as.data.frame(rasterToPoints(r.explanatory))  
newdata$predicted\_height.ols <-predict(mod, newdata, na.action = na.pass)  
newdata$predicted\_height.gls\_2 <-predict(mod.cor\_2, newdata, na.action = na.pass)  
newdata$predicted\_height.gls\_4 <-predict(mod.cor\_4, newdata, na.action = na.pass)  
newdata$predicted\_height.gls\_5 <-predict(mod.cor\_5, newdata, na.action = na.pass)  
  
r.height\_model.ols <- rasterFromXYZ(newdata[,c("x","y","predicted\_height.ols")])  
r.height\_model.gls\_2 <- rasterFromXYZ(newdata[,c("x","y","predicted\_height.gls\_2")])  
r.height\_model.gls\_4 <- rasterFromXYZ(newdata[,c("x","y","predicted\_height.gls\_4")])  
r.height\_model.gls\_5 <- rasterFromXYZ(newdata[,c("x","y","predicted\_height.gls\_5")])  
  
projection(r.height\_model.ols) <- "+proj=utm +zone=33"  
projection(r.height\_model.gls\_2) <- "+proj=utm +zone=33"  
projection(r.height\_model.gls\_4) <- "+proj=utm +zone=33"  
projection(r.height\_model.gls\_5) <- "+proj=utm +zone=33"  
  
writeRaster(r.height\_model.ols, filename="temp\_height\_OLS.tif", format="GTiff", overwrite=TRUE)  
writeRaster(r.height\_model.gls\_2, filename="temp\_height\_GLS2.tif", format="GTiff", overwrite=TRUE)  
writeRaster(r.height\_model.gls\_4, filename="temp\_height\_GLS4.tif", format="GTiff", overwrite=TRUE)  
writeRaster(r.height\_model.gls\_5, filename="temp\_height\_GLS5.tif", format="GTiff", overwrite=TRUE)