CAR

Zofie Cimburova

6 juli 2017

## Libraries

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.1

library(raster)  
library(rgdal)   
library(spdep)  
library(gstat)  
library(rasterVis)

## Warning: package 'rasterVis' was built under R version 3.4.1

# 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.

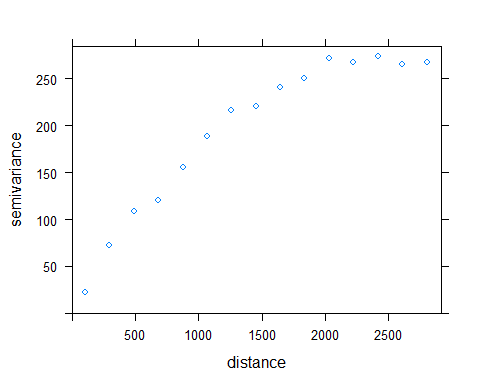
bio11 <- raster("BIO11\_kriged\_sample.tif", band=1)  
projection(bio11) <- "+proj=utm +zone=33 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no\_defs"  
  
r.explanatory <- brick(bio11,  
 raster("TPI1010.tif", band=1),  
 raster("lat.tif", band=1),   
 raster("slope.tif", band=1),  
 raster("srad10.tif", band=1),  
 raster("DEM.tif", band=1),  
 raster("forest\_line\_10m.tif", band=1))  
names(r.explanatory) <- c('BIO11','TPI1010','lat','slope','srad','height','forest\_line')

# 3. Fit CAR

## check semivariances to find range

coordinates(obs)<-c('X','Y')  
  
var.mod<-variogram(height~BIO11 + srad + lat + slope + TPI1010,data=obs)

plot(var.mod)



## 3.1. Calculate 1st order neighbourhood list for CAR model using standard weights.

!!! limit distance d2 needs to be changed so that each observation point has at least one neighbour

neighbours <- dnearneigh(cbind(obs$X, obs$Y),   
 d1 = 0,  
 d2 = 2000)  
   
print(neighbours)

## Neighbour list object:  
## Number of regions: 2901   
## Number of nonzero links: 2414454   
## Percentage nonzero weights: 28.68953   
## Average number of links: 832.2834

## 3.2. Calculate spatial weights for neighbours lists

listw <- nb2listw(neighbours = neighbours,  
 glist = NULL,  
 style = "W",  
 zero.policy = NULL)

## 3.3. Fit CAR

obs <- as.data.frame(obs)  
mod.car <- errorsarlm(formula = height~BIO11 + srad + lat + TPI1010,   
 data = obs,  
 listw = listw)

## Warning in errorsarlm(formula = height ~ BIO11 + srad + lat + TPI1010, data = obs, : inversion of asymptotic covariance matrix failed for tol.solve = 1e-10   
## reciprocal condition number = 8.4835e-16 - using numerical Hessian.

## 3.3. Fit OLS

obs <- as.data.frame(obs)  
mod.ols <- lm(formula = height~BIO11 + srad + lat + TPI1010, data = obs)

## 3.4. Predict the whole area

!!! If Error "missing values in newdata", there are missing values in newdata in fact, and need to be excluded (na.action does not help) or replaced or whatever

newdata <- as.data.frame(rasterToPoints(r.explanatory))  
newdata[is.na(newdata)] <- 0  
newdata$predicted\_height.car <-predict(mod.car, newdata)

## Warning in predict.sarlm(mod.car, newdata): some region.id are both in data  
## and newdata

r.height\_model.car <- rasterFromXYZ(newdata[,c("x","y","predicted\_height.car")])  
  
#projection(r.height\_model.car) <- "+proj=utm +zone=33"  
  
r.explanatory$forest\_line\_predict <- r.explanatory$height - r.height\_model.car  
#writeRaster(r.height\_model.car, filename="temp\_height\_car.tif", format="GTiff", overwrite=TRUE)

#theme\_set(theme\_bw())  
#gplot(r.forest\_line) + geom\_tile(aes(fill = value)) +  
# facet\_wrap(~ variable) +  
# scale\_fill\_gradient2(low = "red", mid = "white", high = "blue", midpoint = 0) +  
# coord\_equal()  
  
fl.predict <- levelplot(r.explanatory$forest\_line\_predict, margin = FALSE, main = "predicted height of forest line", at = unique(c(seq(-150,-10,length=100), seq(-10,0,length=100),seq(0,10,length=100), seq(10,100,length=100))), col.regions = colorRampPalette(c("blue","blue","white","red","red"))(1e3), alpha.regions = 0.35)  
  
fl.obs <- levelplot(r.explanatory$forest\_line, margin = FALSE, col.regions = "black" )  
  
print(fl.predict + fl.obs)

