INLA for ECOFUNC

include=FALSE nothing message=FALSE no output echo=FALSE no code

## 1. Load the data

470 501 observations are exported from GRASS. For each observation point, its x- and y- coordinate, row and column number, total number of forest pixels, total number of forest and open land pixels and median values of environmental explanatory variables are stored.

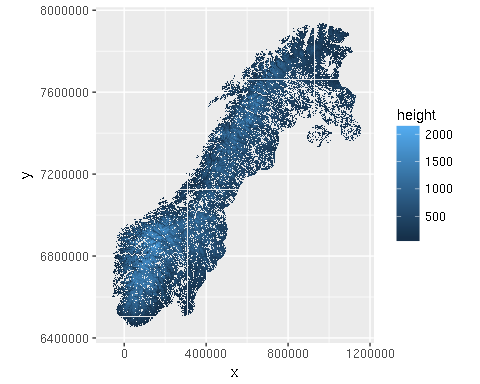
input\_data <- read.csv("variables\_avg\_med\_std.csv")

## water\_med aspect\_med curv\_med slope\_med topex\_e\_med topex\_n\_med  
## 1 206.15528 190.2394 -0.0001985216 10.23331 1.9473065 4.460040  
## 2 154.05694 190.0079 -0.0005031548 12.15138 5.3213400 4.146689  
## 3 300.00000 202.9225 0.0000000000 11.13258 2.0808316 4.915342  
## 4 85.35534 198.5654 0.0003948833 14.39963 1.6229463 6.402441  
## 5 50.00000 221.5318 0.0001157182 11.84057 0.5858289 4.687766  
## 6 50.00000 197.3695 0.0010081581 12.97935 3.9471529 3.690465  
## topex\_ne\_med topex\_nw\_med topex\_s\_med topex\_se\_med topex\_sw\_med  
## 1 2.440755 4.934301 1.890075029 1.5237084 2.26200745  
## 2 5.880727 1.632487 0.859372331 3.6048700 -0.03437747  
## 3 3.932897 4.431565 1.460725899 1.2245110 1.31757234  
## 4 5.122872 5.066975 2.976699955 1.7183580 1.72685024  
## 5 2.890981 3.407124 -0.154698233 -0.1188886 -0.05729576  
## 6 4.700571 1.203035 -0.005729578 1.0312122 0.65682630  
## topex\_w\_med tpi\_250\_med tpi\_500\_med tpi\_1000\_med tpi\_2500\_med  
## 1 2.9195552 1.018805 2.321847 2.998959 -0.2969551  
## 2 0.8498251 1.210523 2.563129 7.668012 2.0935794  
## 3 2.2906100 -1.812025 -7.540287 -24.279856 -39.5940497  
## 4 2.9731273 3.993119 8.222554 17.205572 25.3239514  
## 5 0.5729387 1.219902 3.168644 8.099478 28.9341941  
## 6 0.8250022 2.489182 2.343467 6.197872 2.2298732  
## tpi\_5000\_med tri\_med solrad\_apr\_med solrad\_autumn\_med solrad\_jan\_med  
## 1 10.66296 1.437500 35002.0 25100.0 4086  
## 2 10.51268 1.775000 35046.0 25875.0 4394  
## 3 -26.13902 1.587501 35196.0 25665.5 4267  
## 4 19.61821 2.075000 35364.5 26010.0 4257  
## 5 47.54461 1.700000 35986.0 28050.0 4975  
## 6 13.09215 1.850000 35361.0 26287.0 4475  
## solrad\_jul\_med solrad\_oct\_med solrad\_spring\_med solrad\_summer\_med  
## 1 47565.0 14058.5 126702 125533.0  
## 2 47396.0 14320.0 126566 125385.0  
## 3 47711.0 14294.0 127225 126057.5  
## 4 47600.5 14547.5 127318 126140.5  
## 5 48012.0 15283.0 129148 127961.0  
## 6 47741.0 14517.0 127511 126336.0  
## solrad\_winter\_med solrad\_year\_med bio01\_med bio02\_med bio10\_med  
## 1 25219.5 302560.0 66.83707 64.96054 142.8538  
## 2 26058.0 303672.0 65.66058 63.96765 140.8750  
## 3 25801.5 304572.5 65.29499 65.34832 141.3120  
## 4 26172.0 305227.0 66.73115 67.31291 141.7201  
## 5 28308.0 313470.0 67.90534 62.88028 144.9871  
## 6 26491.0 306306.0 65.56265 62.67090 140.4025  
## bio11\_med bio12\_med bio15\_med bio18\_med bio19\_med sea\_med sea\_open\_med  
## 1 -6.887221 1339.930 31.70146 283.1354 307.1634 196.97716 7229.972  
## 2 -8.097725 1351.973 31.03051 285.4983 310.5674 155.56349 7605.672  
## 3 -9.582597 1347.168 31.33226 284.4750 309.1902 491.93496 8174.427  
## 4 -7.199246 1386.477 31.02343 293.6740 321.1713 86.02325 8765.058  
## 5 -7.088239 1463.103 31.62241 306.1731 333.5310 28.28427 5106.490  
## 6 -8.044944 1395.067 31.36068 294.1772 319.6613 10.00000 6311.200  
## height  
## 1 23.32068  
## 2 24.40883  
## 3 54.63885  
## 4 22.52763  
## 5 16.55163  
## 6 26.73613

## 2. some filtering

All incomplete observations are removed, i.e. each observation point contains full data

input\_data\_na\_rm <- input\_data[complete.cases(input\_data),]



## 3. assign node ID for INLA

In order for INLA to work, each observation point is assigned an ID which determines its position in raster. Indices are assigned vertically from top left corner to bottom right corner.

require(INLA)  
  
nrows=1808  
ncols=1414  
  
input\_data\_na\_rm$ID <- NA  
input\_data\_na\_rm$ID = inla.lattice2node(input\_data\_na\_rm$row, input\_data\_na\_rm$col, nrows, ncols)

# 4. standardize data

For logistic regression predictors need to be standardized to zero mean and standard deviation equal to one, i.e. column mean is subtracted and the result is divided by standard deviation. Column means and standard deviations are kept in order to be utilized in prediction.

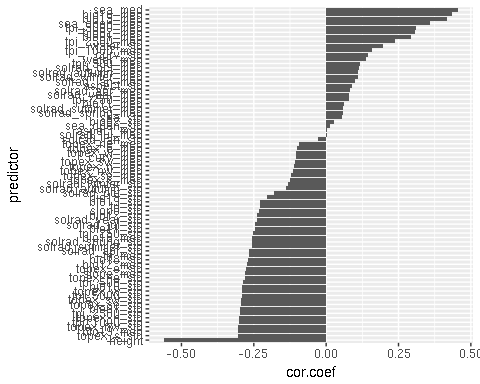
library(robustHD)  
  
input\_data\_na\_rm.scaled <- input\_data\_na\_rm  
input\_data\_na\_rm.scaled[,observations.names] <- standardize(input\_data\_na\_rm[,observations.names])  
  
input\_data\_na\_rm.means <- colMeans(input\_data\_na\_rm[,observations.names])  
input\_data\_na\_rm.stddev <- apply(input\_data\_na\_rm[,observations.names], 2, sd)

# 5. exploratory data analysis

## Correlation coefficient of response~predictor

First, correlations between response and predictors are explored.

data.cor.coef <- data.frame(cor.coef=double())  
i = 1  
for (predictor in observations.names) {  
 data.cor.coef[i,] <- cor(input\_data\_na\_rm.scaled$nsuccess/input\_data\_na\_rm.scaled$ntrials, input\_data\_na\_rm.scaled[,predictor])  
 row.names(data.cor.coef)[i]<- predictor  
 i = i+1  
}  
  
data.cor.coef$predictor <- rownames(data.cor.coef)  
data.cor.coef <- data.cor.coef[order(data.cor.coef$cor.coef),]   
data.cor.coef$predictor <- factor(data.cor.coef$predictor, levels = data.cor.coef$predictor[order(data.cor.coef$cor.coef)])

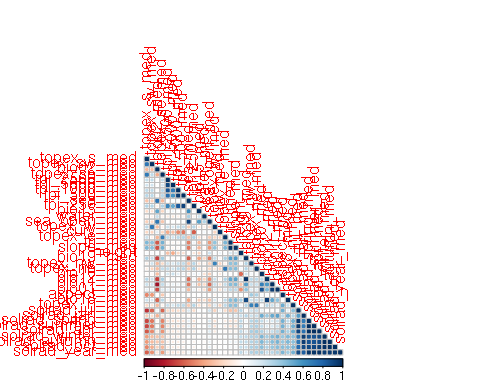


## Correlations between predictors

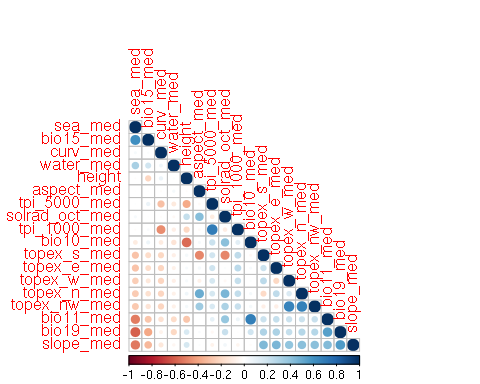
Correlations between predictors are checked and eliminated by three methods - first, correlation matrix is computed and pairs with correlation coefficient larger than 0.7 are explored. Second, full model is fitted with GLM, and predictors with variance inflation factor larger than 4 are exluded. Third, the data are projected on the first two principal components and predictors with similar directions are explored.

* Correlation matrix: In a pair of correlated predictors, the one with larger correlation coefficient with response is generally kept.

M.med <- cor(input\_data\_na\_rm.scaled[,observations.names.mod])  
M.med.lim <- M.med\*(abs(M.med)>=0.7)



observation.names.mod.exclude <- observations.names.mod[!observations.names.mod %in%   
 c("solrad\_jul\_med","solrad\_spring\_med","solrad\_summer\_med",  
 "solrad\_year\_med","solrad\_apr\_med","solrad\_jan\_med",  
 "tpi\_2500\_med","solrad\_winter\_med","solrad\_autumn\_med",  
 "bio12\_med","topex\_ne\_med","topex\_sw\_med",  
 "topex\_se\_med","tri\_med","bio01\_med","tpi\_250\_med",  
 "tpi\_500\_med","bio02\_med","sea\_open\_med","bio18\_med"  
 )]  
  
M.med.2 <- cor(input\_data\_na\_rm.scaled[,observation.names.mod.exclude])  
M.med.2.lim <- M.med.2\*(abs(M.med.2)>=0.7)



* VIF: A full logistic model is fitted with GLM, ommiting the spatial random effect. Predictiors with VIF larger than 4 are removed one-by-one using a backwards, stepwise procedure. Excluded predictors are solrad\_oct\_med, topex\_nw\_med and slope\_med.

library(car)  
  
formula.med <- lc ~ water\_med + sea\_med +   
 aspect\_med + curv\_med +   
 topex\_e\_med + topex\_n\_med+ topex\_s\_med + topex\_w\_med +  
 tpi\_1000\_med + tpi\_5000\_med +   
 bio10\_med + bio11\_med + bio15\_med + bio19\_med + height  
  
mod.glm.med <- glm(formula.med,   
 data=input\_data\_na\_rm.scaled,   
 family=binomial(link='logit'))

## Warning: non-integer #successes in a binomial glm!

vif(mod.glm.med)

## water\_med sea\_med aspect\_med curv\_med topex\_e\_med   
## 1.119094 3.038197 1.741380 1.385605 1.715937   
## topex\_n\_med topex\_s\_med topex\_w\_med tpi\_1000\_med tpi\_5000\_med   
## 1.957463 2.050504 1.610099 2.437915 2.082447   
## bio10\_med bio11\_med bio15\_med bio19\_med height   
## 2.406602 3.375371 1.494600 2.220417 1.920338

-PCA: excluded: bio11\_med, tpi\_100\_med

#fit <- princomp(input\_data\_na\_rm.scaled[,c("water\_med","sea\_med","aspect\_med","curv\_med",  
# "topex\_e\_med","topex\_n\_med","topex\_s\_med","topex\_w\_med",  
# "tpi\_5000\_med","bio10\_med",  
# "bio15\_med","bio19\_med")], cor=TRUE)   
#summary(fit) # variance accounted for, pulls out the variables on the first 2 axes  
#loadings(fit) # pc loadings