Exploratory data analysis for ECOFUNC

# 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("observations\_fenoscandia.csv")

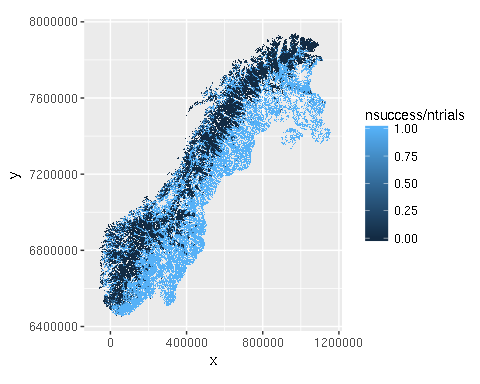
input\_data$lc\_proportion <- input\_data$nsuccess/input\_data$ntrial

## water curvature slope tpi\_250 tpi\_500 tpi\_1000  
## 1 206.15528 -0.0001985216 10.23331 1.018805 2.321847 2.998959  
## 2 154.05694 -0.0005031548 12.15138 1.210523 2.563129 7.668012  
## 3 300.00000 0.0000000000 11.13258 -1.812025 -7.540287 -24.279856  
## 4 85.35534 0.0003948833 14.39963 3.993119 8.222554 17.205572  
## 5 50.00000 0.0001157182 11.84057 1.219902 3.168644 8.099478  
## tpi\_2500 tpi\_5000 tri solrad\_apr solrad\_autumn solrad\_jan  
## 1 -0.2969551 10.66296 1.437500 35002.0 25100.0 4086  
## 2 2.0935794 10.51268 1.775000 35046.0 25875.0 4394  
## 3 -39.5940497 -26.13902 1.587501 35196.0 25665.5 4267  
## 4 25.3239514 19.61821 2.075000 35364.5 26010.0 4257  
## 5 28.9341941 47.54461 1.700000 35986.0 28050.0 4975  
## solrad\_jul solrad\_oct solrad\_spring solrad\_summer solrad\_winter  
## 1 47565.0 14058.5 126702 125533.0 25219.5  
## 2 47396.0 14320.0 126566 125385.0 26058.0  
## 3 47711.0 14294.0 127225 126057.5 25801.5  
## 4 47600.5 14547.5 127318 126140.5 26172.0  
## 5 48012.0 15283.0 129148 127961.0 28308.0  
## solrad\_year bio01 bio02 bio10 bio11 bio12 bio15  
## 1 302560.0 66.83707 64.96054 142.8538 -6.887221 1339.930 31.70146  
## 2 303672.0 65.66058 63.96765 140.8750 -8.097725 1351.973 31.03051  
## 3 304572.5 65.29499 65.34832 141.3120 -9.582597 1347.168 31.33226  
## 4 305227.0 66.73115 67.31291 141.7201 -7.199246 1386.477 31.02343  
## 5 313470.0 67.90534 62.88028 144.9871 -7.088239 1463.103 31.62241  
## bio18 bio19 sea sea\_open height aspect\_sin aspect\_cos  
## 1 283.1354 307.1634 196.97716 7229.972 23.32068 -0.09053572 0.05859006  
## 2 285.4983 310.5674 155.56349 7605.672 24.40883 -0.11596365 -0.50103589  
## 3 284.4750 309.1902 491.93496 8174.427 54.63885 -0.16900828 0.09480834  
## 4 293.6740 321.1713 86.02325 8765.058 22.52763 -0.17354431 -0.08344940  
## 5 306.1731 333.5310 28.28427 5106.490 16.55163 -0.31900030 0.24175638  
## tpi\_10000 exposure  
## 1 23.40649 0.15660833  
## 2 28.43821 0.58439671  
## 3 -11.77459 1.81325779  
## 4 33.65990 0.09740273  
## 5 45.64451 0.88714711

# 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. Indexes 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[,pred.names] <- standardize(input\_data\_na\_rm[,pred.names])  
  
input\_data\_na\_rm.means <- colMeans(input\_data\_na\_rm[,pred.names])  
input\_data\_na\_rm.stddev <- apply(input\_data\_na\_rm[,pred.names], 2, sd)

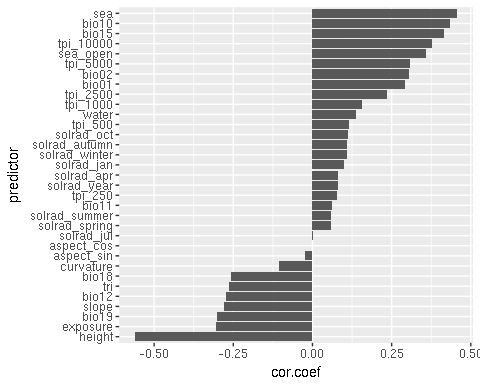
# 5. exploratory data analysis

## 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 pred.names) {  
 data.cor.coef[i,] <- cor(input\_data\_na\_rm.scaled$lc\_proportion, 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.

This procedure results in two sets of predictors - one keeping the indirectly influencing covariates - often correlated with similar directly influencing ones, but having better correlation coefficient with the response, the other one keeping only the directly influencing covariates. An example is the correlation between distance from sea, and bio02 temperature (approximation of continentality). Both of them represent a measure of continentality, but whilst distance from sea influences the probability of tree occurence indirectly (i.e. influences the climatic conditions), bio02 has a direct influence.

### Exluding indirectly influencing predictors

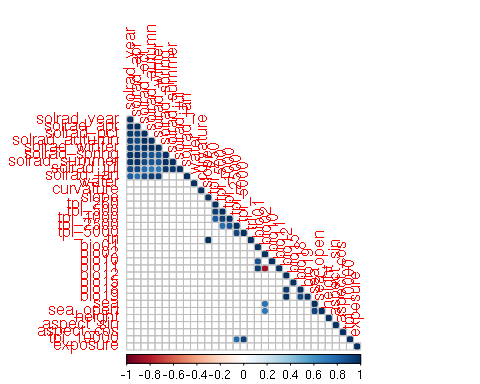
Following predictors were excluded:

|  |  |
| --- | --- |
| Prioritizing direct influences | Prioritizing indirect influences |
| height |  |
| aspect\_sin |  |
| aspect\_cos |  |

### Correlation matrix

Correlation matrix was computed and pairs of covariates with **correlation coefficient larger than 0.7** were explored. Predictor with larger correlation coefficient with response was generally kept.

library(corrplot)  
  
M.cor <- cor(input\_data\_na\_rm.scaled[,pred.names])  
M.cor.lim <- M.cor\*(abs(M.cor)>=0.7)

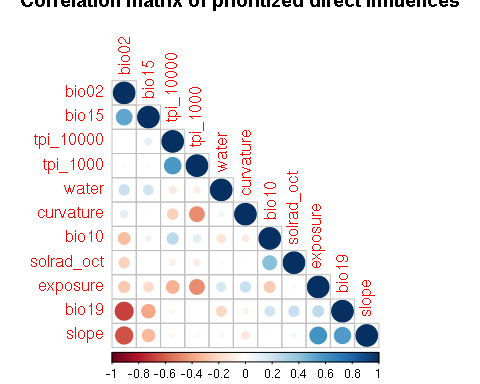


Following predictors were excluded:

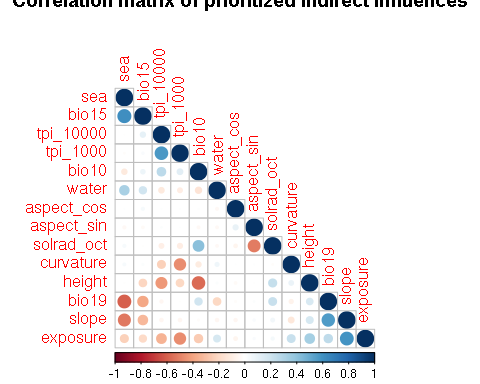
|  |  |
| --- | --- |
| Prioritizing direct influences | Prioritizing indirect influences |
| solrad\_jul | solrad\_jul |
| solrad\_spring | solrad\_spring |
| solrad\_summer | solrad\_summer |
| solrad\_year | solrad\_year |
| solrad\_apr | solrad\_apr |
| solrad\_jan | solrad\_jan |
| solrad\_winter | solrad\_winter |
| solrad\_autumn | solrad\_autumn |
| tpi\_2500 | tpi\_2500 |
| tpi\_250 | tpi\_250 |
| tpi\_500 | tpi\_500 |
| tpi\_5000 | tpi\_5000 |
| bio12 | bio12 |
| bio18 | bio18 |
| bio11 | bio11 |
| tri | tri |
| bio01 | bio01 |
| sea\_open | sea\_open |
| sea | bio02 |

pres.names.exclude.direct <- pred.names[!pred.names %in%   
 c("solrad\_jul","solrad\_spring","solrad\_summer","solrad\_year",  
 "solrad\_apr","solrad\_jan","solrad\_winter","solrad\_autumn",  
 "tpi\_2500","tpi\_250","tpi\_500","tpi\_5000",  
 "bio12","bio18","bio11","tri","bio01","sea\_open","sea",  
 "longitude","latitude","height","aspect\_sin","aspect\_cos")]  
pres.names.exclude.indirect <- pred.names[!pred.names %in%   
 c("solrad\_jul","solrad\_spring","solrad\_summer","solrad\_year",  
 "solrad\_apr","solrad\_jan","solrad\_winter","solrad\_autumn",  
 "tpi\_2500","tpi\_250","tpi\_500","tpi\_5000",  
 "bio12","bio18","bio11","tri","bio01","sea\_open","bio02")]  
  
M.cor.direct <- cor(input\_data\_na\_rm.scaled[,pres.names.exclude.direct])  
M.cor.indirect <- cor(input\_data\_na\_rm.scaled[,pres.names.exclude.indirect])  
  
M.cor.direct.lim <- M.cor.direct\*(abs(M.cor.direct)>=0.7)  
M.cor.indirect.lim <- M.cor.indirect\*(abs(M.cor.indirect)>=0.7)

Correlation matrix of prioritized direct influences



Correlation matrix of prioritized indirect influences



### VIF

A full logistic model (using all covariates left from previous step) was fitted with GLM, ommiting the spatial random effect. Predictors with **VIF larger than 4** were removed one-by-one using a backwards, stepwise procedure.

Following predictors were excluded:

|  |  |
| --- | --- |
| Prioritizing direct influences | Prioritizing indirect influences |
| solrad\_oct |

library(car)  
formula.glm.direct <- lc\_proportion ~ water + curvature + slope + tpi\_1000 + bio10 + bio15 + bio19 + tpi\_10000 + exposure + bio02 + solrad\_octformula.glm.indirect <- lc\_proportion ~ water + curvature + slope + tpi\_1000 + bio10 + bio15 + bio19 + tpi\_10000 + exposure + sea + height + aspect\_sin + aspect\_cos   
  
glm.direct <- glm(formula.glm.direct, data=input\_data\_na\_rm.scaled,  family=binomial(link='logit'))  
glm.indirect <- glm(formula.glm.indirect, data=input\_data\_na\_rm.scaled,   
 family=binomial(link='logit'))

#vif(glm.direct)  
vif(glm.indirect)

## water curvature slope tpi\_1000 bio10 bio15   
## 1.216729 1.375629 2.649207 1.946300 1.647095 1.486832   
## bio19 tpi\_10000 exposure sea height aspect\_sin   
## 2.127064 1.585076 2.437121 2.974916 1.873779 1.010439   
## aspect\_cos   
## 1.014841

### PCA

Original variables were plotted in the 2D space of first two principal components. In case the cumulative propostion of variance explained by the first two principal components is high, the angles between vectors of original variables may be interpreted as correlations - small angles indicate the variables are positively correlated, an angle of 90 degrees indicates the variables are not correlated, and angles close to 180 degrees indicate the variables are negatively correlated.

For subset of variables prioritizing direct influences, the cumulative proportion of variance explained by the first two components is 52 %, and for subset of variables prioritizing indirect influences, the cumulative proportion of variance explained by the first two components is 42 %. I assume this proportion is too low to enable interpreting the correlation between variables.

Following predictors were excluded:

|  |  |
| --- | --- |
| Prioritizing direct influences | Prioritizing indirect influences |
| tpi\_1000 | tpi\_1000 |

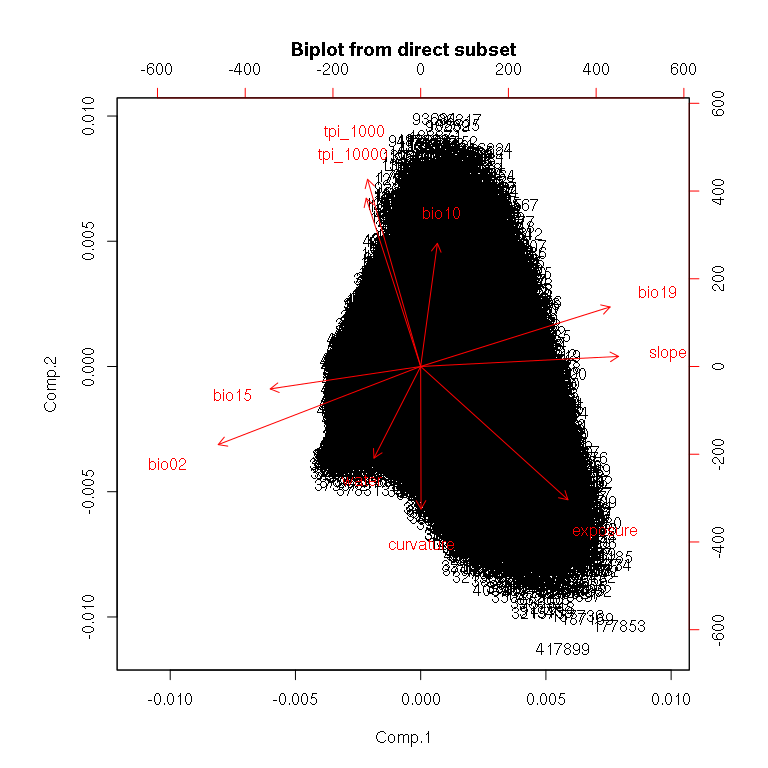
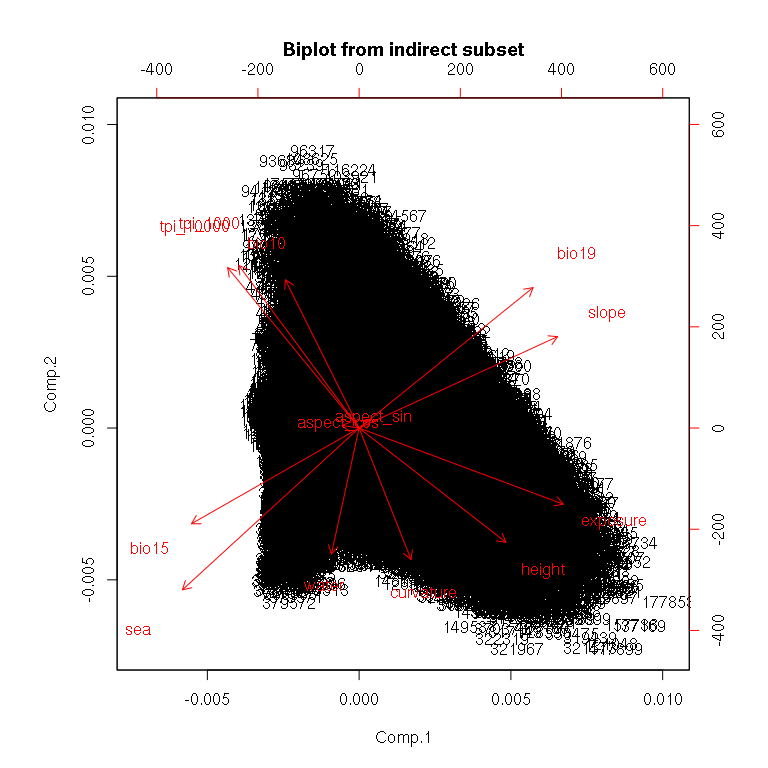
pca.direct <- princomp(input\_data\_na\_rm.scaled[,c("water", "curvature", "slope", "bio10", "bio15" , "tpi\_1000",  
 "bio19" , "tpi\_10000" ,"exposure", "bio02")], cor=TRUE)   
  
pca.indirect <- princomp(input\_data\_na\_rm.scaled[,c("water", "curvature", "slope" , "bio10", "bio15" , "tpi\_1000",  
 "bio19" , "height" , "aspect\_sin" ,"aspect\_cos" , "tpi\_10000" ,"exposure", "sea")], cor=TRUE)   
  
summary(pca.indirect) # variance accounted for

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 1.7396272 1.5682262 1.13802755 1.0736403 1.03351862  
## Proportion of Variance 0.2327925 0.1891795 0.09962359 0.0886695 0.08216621  
## Cumulative Proportion 0.2327925 0.4219720 0.52159561 0.6102651 0.69243131  
## Comp.6 Comp.7 Comp.8 Comp.9 Comp.10  
## Standard deviation 0.9488397 0.86286119 0.8308218 0.7481132 0.59700417  
## Proportion of Variance 0.0692536 0.05727149 0.0530973 0.0430518 0.02741646  
## Cumulative Proportion 0.7616849 0.81895641 0.8720537 0.9151055 0.94252197  
## Comp.11 Comp.12 Comp.13  
## Standard deviation 0.5561357 0.47706075 0.45862897  
## Proportion of Variance 0.0237913 0.01750669 0.01618004  
## Cumulative Proportion 0.9663133 0.98381996 1.00000000

summary(pca.direct) # variance accounted for

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 1.7059723 1.5278436 1.0695195 0.98967792 0.85326883  
## Proportion of Variance 0.2910342 0.2334306 0.1143872 0.09794624 0.07280677  
## Cumulative Proportion 0.2910342 0.5244648 0.6388520 0.73679820 0.80960497  
## Comp.6 Comp.7 Comp.8 Comp.9  
## Standard deviation 0.82543253 0.68482820 0.56381723 0.49386546  
## Proportion of Variance 0.06813389 0.04689897 0.03178899 0.02439031  
## Cumulative Proportion 0.87773886 0.92463782 0.95642681 0.98081712  
## Comp.10  
## Standard deviation 0.43798266  
## Proportion of Variance 0.01918288  
## Cumulative Proportion 1.00000000

#loadings(pca.direct) # pc loadings



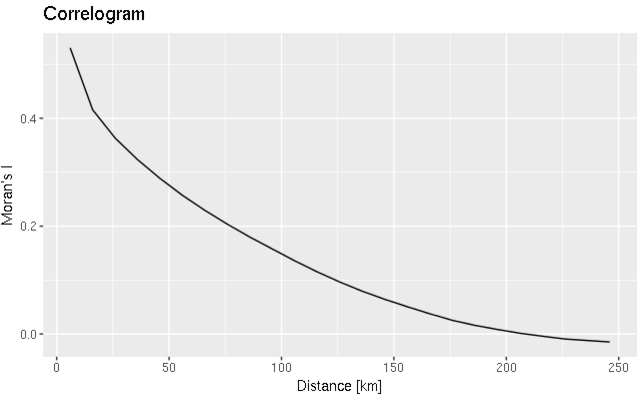
# 8. Estimating range of spatial autocorrelation

Spatial autocorrelation in response is estimated based on observations of *Moran's I correlogram* and a *variogram*.

## Moran's I - correlogram

Moran's I results in values from 1 (complete positive spatial autocorrelation) through 0 (no spatial autocorrelation) to -1 (complete negative spatial autocorrelation). Correlogram with 100 km range and 10 km bins.

library(raster)  
  
raster <- rasterFromXYZ(cbind(input\_data\_na\_rm.scaled$x, input\_data\_na\_rm.scaled$y, input\_data\_na\_rm.scaled$lc))  
  
# set bins width and range of correlogram  
lim.dist <- 250 # range, no.cells (half)  
width <- 10 # width, no.cells  
  
# prepare distance matrix  
distances <- vector()  
for (row in -lim.dist:lim.dist){  
 for (col in -lim.dist:lim.dist){  
 distances <- c(distances,sqrt(row\*row + col\*col))  
 }  
}  
distance.matrix <- matrix(data = distances, nrow = 2\*lim.dist+1, ncol = 2\*lim.dist+1)   
centre <- lim.dist+1  
   
# compute Moran's I in bins  
M.lags <- data.frame("x"=double(), "moran"=double())  
i = 1  
for (bin in seq(1,lim.dist,width)){  
 # limit distances for bins  
 lim.lower <- bin  
 lim.upper <- bin + width  
  
 # compute weight matrix  
 weight.matrix <- (distance.matrix >= lim.lower & distance.matrix < lim.upper)\*1  
   
 # shrink weight matrix  
 border.lower <- max(0,centre-lim.upper+1)  
 border.upper <- min(centre+lim.upper-1,nrow(distance.matrix))  
   
 weight.matrix <- weight.matrix[border.lower:border.upper,border.lower:border.upper]  
  
 # compute Moran's I  
 M.lags[i,]$x <- (lim.lower+lim.upper)/2  
 M.lags[i,]$moran <- Moran(raster, w=weight.matrix)   
 i <- i+1  
}



## Variogram

Variogram with 250 km range and 10 km bins.

library(gstat)  
coordinates(input\_data\_na\_rm.scaled)=~x+y  
  
myvariogram <- variogram(lc\_proportion~x+y, data=input\_data\_na\_rm.scaled, cutoff=1000, width=1000)

