Humidity model

# General Libraries

library(dplyr)  
library(pls)  
library(glmnet)

# Load the data

data <- read.csv("./data.csv",  
 head=TRUE,  
 sep=";",  
 stringsAsFactors = FALSE  
 ) %>%  
 `rownames<-`(.$site) %>%  
 select(-type, -site)  
  
dry\_season <- data %>%  
 slice(1:13) %>%  
 select(which(!colSums(., na.rm=TRUE) %in% 0)) # filter out "zero"-species  
  
wet\_season <- data %>%  
 slice(14:nrow(data)) %>%  
 select(which(!colSums(., na.rm=TRUE) %in% 0)) # filter out "zero"-species  
  
## selecting species trapped both in wet and dry seasons  
names\_in\_common <- intersect(names(wet\_season), names(dry\_season))

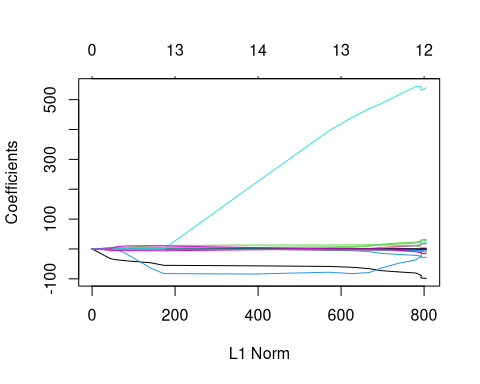
Selecting only species in common in wet and dry seasons.

wet\_season <- wet\_season[names\_in\_common]  
dry\_season <- dry\_season[names\_in\_common]

# The Lasso

require(glmnet)  
x <- model.matrix(humidity ~ ., data = wet\_season)[, -1]  
y <- wet\_season$humidity  
  
## values from 10^10 to 10^2 too choose the best lambda  
grid <- 10^seq(10, -2, length = 100)  
  
lasso.mod <- glmnet(x, y, alpha = 1, lambda = grid)  
plot(lasso.mod)

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):  
## collapsing to unique 'x' values



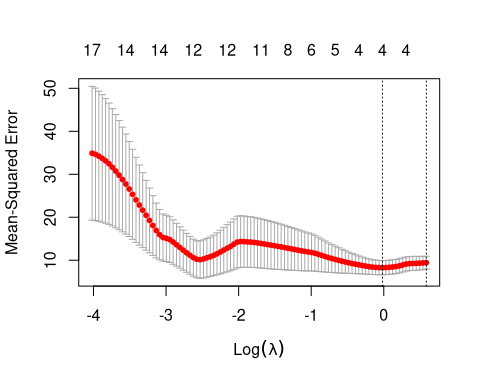
set.seed(1279)  
cv.out <- cv.glmnet(x, y, alpha = 1)

## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per  
## fold

cv.out

##   
## Call: cv.glmnet(x = x, y = y, alpha = 1)   
##   
## Measure: Mean-Squared Error   
##   
## Lambda Measure SE Nonzero  
## min 0.981 8.27 1.65 4  
## 1se 1.797 9.43 1.51 0

plot(cv.out)



bestlam=cv.out$lambda.min  
## 0.9814  
  
x2 <- model.matrix(humidity ~ ., data = dry\_season)[, -1]  
  
  
lasso.pred <- predict(lasso.mod, s=bestlam, newx = x2)  
lasso.coef <- predict(lasso.mod, type="coefficients", s=bestlam)[1:22,]  
lasso.coef[lasso.coef!=0]

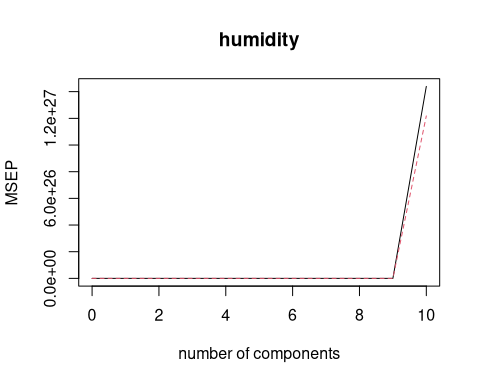
## (Intercept) No.palustris Ca.granulatus Pt.melanarius   
## 6.181 -20.982 3.011 -1.744   
## Pt.oblongopunctatus   
## -2.769

## Principal Components Regression (PCR)

# Principal Components Regression  
require(pls)  
set.seed(2)  
  
## Making equal number of rows in both wet and dry seasons  
wet\_season13 <- wet\_season[-c(8:9), ]  
  
pcr.fit <- pcr(humidity ~ ., data = wet\_season13, scale = FALSE, validation = "CV")  
  
summary(pcr.fit)

## Data: X dimension: 13 21   
## Y dimension: 13 1  
## Fit method: svdpc  
## Number of components considered: 10  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 3.247 3.821 3.791 3.986 3.423 3.633 4.248  
## adjCV 3.247 3.748 3.719 4.000 3.321 3.531 4.123  
## 7 comps 8 comps 9 comps 10 comps  
## CV 5.242 4.703 7.835 3.794e+13  
## adjCV 5.085 4.560 7.545 3.490e+13  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps  
## X 86.07 97.11 98.1 98.94 99.45 99.75 99.91  
## humidity 37.05 42.22 45.1 71.61 71.63 71.76 74.71  
## 8 comps 9 comps 10 comps  
## X 99.95 99.98 100.00  
## humidity 78.64 88.26 97.07

validationplot(pcr.fit, val.type="MSEP")



pcr.pred <- predict(pcr.fit, dry\_season[,-22], ncomp=2)  
  
mean((( pcr.pred )- dry\_season$humidity)^2)

## [1] 9.756

knitr::kable(data.frame(Site = rownames(lasso.pred),  
 "Humidity in dry season" = dry\_season$humidity,  
 "Predicted by the Lasso" = unname(lasso.pred),  
 "Predicted by PCR" = unname(pcr.pred)) %>%  
 format(digits = 1),  
 caption = "Actual and predicted humidity values")

**Table 5**. Actual and predicted humidity values.

|  |  |  |  |
| --- | --- | --- | --- |
| Site | Humidity in dry season | Predicted by the Lasso | Predicted by PCR |
| drybot1 | 4 | 6 | 7 |
| drybot2 | 4 | 6 | 7 |
| drybot3 | 4 | 6 | 7 |
| drybot4 | 4 | 6 | 6 |
| drybot5 | 4 | 6 | 7 |
| drybot6 | 4 | 6 | 7 |
| drybot7 | 5 | 7 | 7 |
| drybot8 | 5 | 6 | 7 |
| dryforest.1 | 2 | 6 | 6 |
| dryforest.2 | 2 | 5 | 5 |
| dryforest.3 | 2 | 4 | 5 |
| dryslope.1 | 2 | 6 | 5 |
| dryslope.2 | 2 | 6 | 5 |