

Lab 6.1: Species mingling in mixed forests

In this Lab we will learn how to:

- Generate a loop to calculate spatial distance between trees in a marteloscope
- Use *ifelse* condition to calculate mingling index
- Fit a linear model to insight of the relation between tree size and mingling index
- Fit a logistic model to test if tree size is related with mixture

We will use also the knowledge we acquire in our previous labs and the Marugan marteloscope data set (DATOS.marugan.csv) that include the following information: tree number (árbol), quadrat in the marteloscope (cuadrante), species (sp), coordinates (x and y), diameter at breast height in centimeters (dbh) and total height in meters (H). As usual, first we must define our working directory and load the data sets we will use but also we will define the number of decimals and the decimal separator we want to use through the function options ()

```
# establishing the working directory
```

```
setwd("C:/datosR")
```

```
# defining output options
```

```
options(digits=3) # number of digits to print on output
```

```
options(OutDec=".") # use dot as decimal separator
```

```
# loading the data sets
```

```
datos.m<-
```

```
read.csv2("DATOS.marugan.csv",sep=";",dec=".",header=T,na.string="NA")
```

As in other labs we should explore the basic features of our dataframe (datos.m) by using this code:

```
# basic features of our dataframe
```

```
names(datos.m)
```

```
head(datos.m)
```

```
tail(datos.m)
```

The first 4 rows of our set are the following:

	arbol	cuadrante	sp	x	y	dbh	H
1	400	16	Pinus pinea	-2.85	-20.29	28.75	9.0
2	402	12	Quercus ilex	1.37	-3.22	8.75	4.7
3	403	12	Quercus ilex	0.00	-2.94	8.30	4.5
4	404	12	Quercus faginea	2.60	-4.49	10.55	5.5

You can observe in your computer the outcomes of the names(), head() and tail() functions. Additionally, you can find useful to apply the code studied in the previous lab (specially the function summary() and the hist() and boxtplot() graphs) to the relevant variables (dbh and H) in data.m

Calculating the distance to the nearest trees

We will use the coordinates of every tree and the basic equation to obtain the Euclidean distance between trees and then we'll select the nearest neighbour trees (*nvecinos*). We will do this in a four steps process:

1. Define the predefined number of nearest trees (*nvecinos*)
2. Use a loop to
 - a. Calculate the distance between all the trees in the dataset.
 - b. Order the results, for each tree, from the closes to the farthest tree.
 - c. Keep only the predefined nearest trees (*nvecinos*)

The script to do that is the following:

```
# calculating the distance to the nearest trees

dist.m<-data.frame()

nvecinos<-3 # defining the number of neighbours

# starting the loop
for (i in 1:nrow(datos.m)){
  distij <- ((datos.m$x[i]-datos.m$x)^2+(datos.m$y[i]-datos.m$y)^2)^(1/2)
  # obtaining the distance from tree i (our target) to every tree

  # now we identify the tree i and its neighbours and order it from closest to
  # farthest distance to our target tree
  temp<-data.frame(arbol=datos.m$arbol[i],vecino=datos.m$arbol,
    sp.arbol=datos.m$sp[i],sp.vecino=datos.m$sp,
    dbh.arbol=datos.m$dbh[i],dbh.vecino=datos.m$dbh,
    H.arbol=datos.m$H[i],H.vecino=datos.m$H,
    distij)
  temp<-temp[order(distij),]

  # delete the first raw because is our target tree i (closest one to tree i)
  temp<-temp[-1,]
  vecinos<-temp[1:nvecinos,] # select the nvecinos trees closest to tree i

dist.m <- rbind(dist.m,vecinos) # now joint the result with our original
}
# close the loop

head(dist.m) # to finish this part we check the new structure of our dataset
```

Obtaining the spatial mingling index

We will use the Gadow (1993) mingling index which is an extension of the species segregation index by Pielou (1977). This index ($M_i^{(k)}$) can be expressed as follow:

$$M_i^{(k)} = \frac{1}{k} \sum_{j=1}^k 1 \text{ if } (species_i \neq species_j) \text{ or } 0 \text{ otherwise}$$

Where k corresponds with our predefined *nvecinos*, *i* is our target tree and *j* is each one of the neighbour trees. $M_i^{(k)}$ ranges between 0 and 1.

```
# Define the mixture variable for each tree, 0 if the species is the same
# 1 otherwise

dist.m$Vij <- ifelse ((dist.m$sp.arbol==dist.m$sp.vecino), 0, 1)
write.csv(dist.m,file="dist_m.csv") # write the outcome in the working directory
head (dist.m) # check if everything is ok

# we calculate the mingling index for each target tree for the nearest neighbours

nfilas =seq(1,length(dist.m[,1]), by=nvecinos)
resultado=c()
# for each different tree
for (i in nfilas){
  vectordindices = c()
  # keep the mingling index value for each target tree
  for (j in seq(0,nvecinos-1)){
    vectordindices=c(vectordindices, dist.m$Vij[i+j])
  }
  # calculate the average of the mingling index for each tree and store it
  indiceagregadoi = mean(vectordindices)
  resultado=c(resultado,rep(indiceagregadoi,nvecinos))
}
# add the mingling index to our data
resultado <-as.matrix(resultado)
colnames(resultado)<-"Mi"
dist.m=cbind(dist.m,resultado)
head(dist.m) # check if everything is ok

new <- dist.m[nfilas,]
names (new) # check the order of the columns

mixture <- new [,c(1,3,5,7,9,11)] #select the columns with the i tree information

# check if everything is ok
names(mixture)
head(mixture)
```

Fitting a linear model to insight on the relationship between tree size and mixture

We will fit a linear model to test if tree size is related with species mingling by using the `lm()` function.

```
# classifying the stands in pure (Mi lower than 0.5) and, otherwise, mixed stands
linear.model <- lm(mixture$Mi ~ mixture$dbh.arbol )
summary(linear.model)
```

To obtain the following output:

Call:

```
lm(formula = mixture$Mi ~ mixture$dbh.arbol)
```

Residuals:

```
    Min     1Q   Median     3Q      Max
-0.6943 -0.2943 -0.0163  0.3082  0.3972
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.71119	0.02291	31.05	<2e-16 ***
mixture\$dbh.arbol	-0.00240	0.00113	-2.13	0.034 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.274 on 433 degrees of freedom

Multiple R-squared: 0.0103, Adjusted R-squared: 0.00806

F-statistic: 4.53 on 1 and 433 DF, p-value: 0.0339

Fit a logistic model to test if tree size is related with mixture

First we have to classify the trees as living in mixture (Mi index equal or greater to 0.5) or not (Mi index below 0.5) by using a *ifelse* function. Then we can fit the logistic model with the *glm()* function with

```
# classifying the trees as in mixture or pure stands
mixture$mixed <- ifelse ((mixture$Mi >= 0.5), 1, 0)
head(mixture,10)
summary(mixture)

# logistic model
logit <- glm(mixture$mixed ~ mixture$dbh.arbol, family=binomial(link='logit'))
summary(logit)
```

To obtain the following output:

Call: glm(formula = mixture\$mixed ~ mixture\$distij, family = binomial(link = "logit"))

Deviance Residuals:

```
    Min     1Q   Median     3Q      Max
-1.661 -1.519  0.786  0.811  1.130
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.090	0.165	6.62	3.6e-11 ***
mixture\$distij	-0.145	0.124	-1.17	0.24

Null deviance: 513.04 on 432 degrees of freedom

Residual deviance: 511.72 on 431 degrees of freedom

AIC: 515.7