Forest Modeling Exercises

Process based and empirical modeling

Olalla Díaz-Yáñez[1], Laura Dobor[2], Katarina Merganicova[3] and Mats Nieberg[4]

1. ETH Zurich (olalla.diaz@usys.ethz.ch)
2. CZU (laura.dobor@gmail.com)
3. CZU (merganicova@fld.czu.cz)
4. PIK-Potsdam | EFI (mats.nieberg@pik-potsdam.de)

Interdisciplinary Summer School
Ljubljana 2023
Repository available in GitHub

Table of contents

1 Overview

This document contains the instructions of the modeling group assignments. It contains a section for the process model approach and a different section for the empirical modelling.

This document describes all the exercises, but your group has been assigned to only one, including one specific modeling approach. Each group has two questions to be answered (one question per subgroup inside each group). Please get in touch with your group coach if you are unsure what section you should focus on. The table below summarizes and links to the different parts of this document:

Group	Subgroup Question	Question	Approach	Go to section
1	A		iLand	
1	B		iLand	
1	C	Does a more diverse forest in structure and composition have more Bryophites species?	GLM	Section 3.4.4
1	D	Is the number of Bryophites species affected by forest management type and the forest structural diversity?	GLM	Section 3.4.5
2	A		iLand	
2	B		iLand	
2	C	Does a more diverse forest in structure and composition have more bird species?	GLM	Section 3.4.6
2	D	Is the number of bird species affected by forest management type and the forest structural diversity?	GLM	Section 3.4.7
3	A		iLand	
3	B		iLand	
3	C	Is the presence of the Great spotted woodpecker affected by forest density?	BRT	Section 3.4.8
3	D	Is the presence of the Great spotted woodpecker affected by forest diversity?	BRT	Section 3.5
4	A		iLand	

Group	Subgroup Question	Question	Approach	Go to section
4	B		iLand	
4	C	Is the presence of the Eurasian treecreeper affected by forest density?	BRT	Section 3.5.1
4	D	Is the presence of the Eurasian treecreeper affected by forest management?	BRT	Section 3.5.2

2 Process based

3 Empiricial modeling

In the empricial modelling we will cover two examples of creating two models based on observed data. We will use two approaches Generalized Linear Models (GLMs) and Boosted Regression trees (BRTs). Please take all the results with a grain of salt, we are making very generalized statements from a limited data set, and we are not getting into the details of how each of the models should be assessed; the goal here is that you learn how observed data can be used to create models that help you to understand the data and relationships better.

3.1 Data

We will work with one dataset derived from the inventory developed to assess forest structure and deadwood properties of six representative forest areas in the Czech Republic (Hošek, n.d.). This dataset collects observations of the presence /absence of certain species of biodiversity importance across 99 plots.

The data was collected from square sampling plots (2500 m2 each) in six forested Czech Republic regions. These regions are representative of the main bio-regions and elevation range of forests, considering their importance in territorial representation, forestry, and ecology. The inventory sampled for biodiversity variables such as the presence/absence of different species of birds, Tracheophyta, bryophytes, fungi, lichens, and beetles.

The dataset has been generously provided by XXXXXX to be used in the context of this exercise. It should not be used for other purposes or be further distributed. If you want to use this data or learn more about it, please get in touch with XXXX here: XXXX.

The data you can download for this exercise is an aggregated summary of the original data, some aspects have been modified from the observed data, so the results you will obtain will only partially match the observed reality. This data is very similar in structure and observed variables to the one you collected in this summer school. The idea is to move from data collection, analysis to this part, where you use the data to create a model.

3.1.1 Data description

These are the variables available in the data

- `longitud`: longitude of the plot location
- `latitude`: latitude of the plot location
- `forestManagementType`: forest management type applied in this plot
- `forestStructure`: current forest structure in the plot
- `slope`: slope of the plot
- `A.pseudoplatanus`: proportion of this tree species in the plot in volume
- `F.sylvatica`: proportion of this tree species in the plot in volume
- `L.decidua`: proportion of this tree species in the plot in volume
- `Q.robur`: proportion of this tree species in the plot in volume
- `S.aucuparia` : proportion of this tree species in the plot in volume
- `B.pendula`: proportion of this tree species in the plot in volume
- `P.abies`: proportion of this tree species in the plot in volume
- `P.sylvestris`: proportion of this tree species in the plot in volume
- `F.excelsior`: proportion of this tree species in the plot in volume
- `A.alba`: proportion of this tree species in the plot in volume
- `A.platanoides`: proportion of this tree species in the plot in volume
- `T.cordata`: proportion of this tree species in the plot in volume
- `S.racemosa`: proportion of this tree species in the plot in volume
- `U.glabra`: proportion of this tree species in the plot in volume
- `S.nigra`: proportion of this tree species in the plot in volume
- `P.alba`: proportion of this tree species in the plot in volume
- `U.minor`: proportion of this tree species in the plot in volume
- `S.caprea`: proportion of this tree species in the plot in volume
- `C.betulus`: proportion of this tree species in the plot in volume
- `P.nigra`: proportion of this tree species in the plot in volume
- `Q.petraea`: proportion of this tree species in the plot in volume
- `S.torminalis`: proportion of this tree species in the plot in volume
- `A.campestre`: proportion of this tree species in the plot in volume
- `P.strobus`: proportion of this tree species in the plot in volume
- `Q.rubra`: proportion of this tree species in the plot in volume
- `volAllha`: total volume in the plot
- `GiniDBH`: Gini index calculated to assess the forest structural diversity in diameter sizes, higher values indicate more structural heterogeneity; lower values indicate more homogeneous stands
- `ShannonIndexTreeSpp`: The Shannon index is way to measure the diversity of tree species in the plot
- `Tracheophyta_rich`: Species richness across Tracheophyta species
- `Birds_rich`: Species richness across Birds species
- `Bryophytes_rich` : Species richness across Bryophytes species
- `Fungi_rich`: Species richness across Fungi species

- `Lichens_rich`: Species richness across Lichens species
- `Beetles_rich`: Species richness across Beetles species
- `dendrocoposMajor`: presence / absence of Dendrocopos major
- `certhia`: presence / absence of Certhia familiaris
- `bryophitaNumObs`: number of observed Bryophytes species
- `birdNumObs`: number of observed Bird species
- `PlotID`: ID number for the plot

3.2 Species description

3.2.1 Species 1 - Bryophytes

Bryophytes constitute an important and permanent component of the forest flora and diversity. They colonize various substrates, which are unsuitable for vascular plants, because of low light intensity or low nutrient level, such as deadwood, bark, rocks, and open soil. They provide shelter habitats, food, and nest material for many animals.

In forests, different ecological guilds of bryophytes can be distinguished by the substrate on which they are growing, including terricolous, lignicolous, corticolous and saxicolous species that occur on soil, deadwood, bark of living trees and shrubs, or rocks, respectively. As diversity and quality of these substrates is affected by forest management, bryophytes are suitable indicators for the effect of management on forest conditions. Especially typical woodland bryophytes, which are strictly depending on forest conditions. It is interesting to better understand the relation of forest management effects on bryophytes and some studies have already demonstrated their sensitivity to management practices.

3.2.2 Species 2 - Great spotted woodpecker (Dendrocopos major)

The great spotted woodpecker (*Dendrocopos major*) is a medium-sized woodpecker with pied black and white plumage and a red patch on the lower belly. It is found in a wide variety of woodlands, broadleaf, coniferous or mixed forests. The great spotted woodpecker spends much of its time climbing trees. It a quite generalist bird species.

3.2.3 Species 3 - Eurasian treecreeper

The Eurasian treecreeper or common treecreeper (Certhia familiaris) is a small passerine bird. It prefers mature trees, and in most of Europe, it tends to be found mainly in coniferous forest, especially spruce and fir.

3.3 Models

3.3.1 Generalized Linear Models (GLMs)

We propose to use a generalized linear model (GLM) to understand the abundance of different bryophytes species in the 99 plots. Count data often conform to a Poisson distribution; in this case, we have a count of the number of species recorded at each plot.

Fitting a Poisson GLM in R is similar to analyzing covariance (or linear model), except that we now need to use the glm function. To run a GLM, we need to provide one extra piece of information beyond that required for a linear model: the family of models we want to use. In this case, we want a Poisson family, family=poisson.

3.3.2 Boosted regression trees (BRTs)

Boosted regression trees (BRT) are a combination of two powerful statistical techniques: boosting and regression trees. Boosting is a machine learning technique similar to model averaging, where the results of several competing models are merged. Unlike model averaging, however, boosting uses a forward, stage-wise procedure, where tree models are fitted iteratively to a subset of the training data. Subsets of the training data used at each iteration of the model fit are randomly selected without replacement, where the proportion of the training data used is determined by the modeler, this is defined with the "bag fraction" parameter. This procedure, known as stochastic gradient boosting, introduces an element of stochasticity that improves model accuracy and reduces overfitting (Elith, Leathwick, and Hastie 2008).

The BRT model calibration is defined by four parameters:

- the `learning rate` (or shrinkage parameter): The learning rate determines the contribution of each new tree to the growing model, and it is always substantially lower than 1, higher values being related to faster learning.
- the `bag fraction`: The bag fraction provides in- formation on which fraction of the entire data should be drawn randomly to fit the new tree. This parameter includes a random probabilistic component, making each run model different, and is aimed at improving model accuracy, speed of model creation, and the reduction of overfitting (Friedman, Hastie, and Tibshirani, n.d.).
- the `tree complexity`: Tree complexity controls the number of fitted interactions among variables, and determines the number of splits in each tree; for example, a value of 1 will present only one split, meaning that the model does not consider interactions; a value of 2 will result in two splits, and two interactions.

- The `number of trees` required for optimal prediction: The optimal number of trees is selected based on the three previous parameters. The values fitted by the final model are computed as the sum of all of the predictions of the trees, multiplied by their respective learning rates.

3.4 The exercise

3.4.1 The project folder

All the project is available in a github repository. You can download the project with this document, code, the `.rproj` and the correct folder structure in here [2].

[2] https://github.com/oldiya/summerSchoolExercise

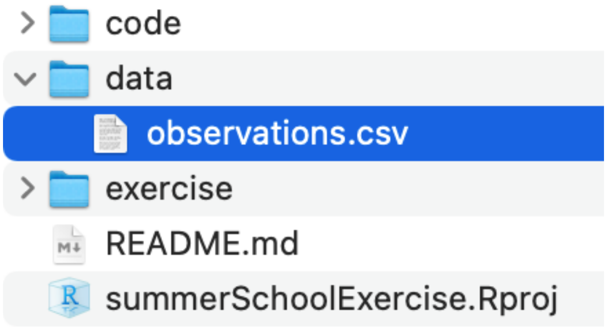
In this project you will find a folder called `code` where the codes for each question are storaged. A folder called `data` where you have to storage the data that you have downloaded. A folder called `exercise`, that contains this document and all the required files to build it.

3.4.2 Download the data

You can download the data in here [1]

[1] https://polybox.ethz.ch/index.php/s/qERYKjzmFTr81Sq

Our recommendation is that you follow this folder structure and you put the data in the folder data of the project and the `.rproj` into the main folder. If you decide to organized things in a different way then you will have to change the path to the code provided to load the data in the section "Explore the data"



code
data
observations.csv
exercise
README.md
summerSchoolExercise.Rproj

3.4.3 Explore the data

You can load the data in R by doing:

```r
observations <- read.csv(here::here("data/observations.csv"))
```

Once you have obtained the data, your next step is to explore it. It is important to carefully analyze the structure of the data and understand its meaning. Each variable must be examined closely, along with the data structure. Remember that each row in this dataset represents one plot, identified by its `PlotID`, and each column represents one variable.

You can explore each of the variables by doing:

```r
str(observations)
```

```
'data.frame':	99 obs. of  45 variables:
 $ X                   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ longitud            : num  13.5 13.5 13.5 13.6 13.6 ...
 $ latitude            : num  49.5 49.5 49.5 49.5 49.5 ...
 $ forestManagementType: chr  "simple clearcutting" "simple clearcutting" "simple clearcu
 $ forestStructure     : chr  "even-aged" "even-aged" "even-aged" "even-aged" ...
 $ slope               : num  16.85 6.51 4.91 5.55 14.55 ...
 $ A.pseudoplatanus    : int  0 0 0 0 0 4 1 7 16 0 ...
 $ F.sylvatica         : int  61 0 0 100 0 82 85 91 75 0 ...
 $ L.decidua           : int  38 0 0 0 4 6 0 0 0 0 ...
 $ Q.robur             : int  0 1 0 0 0 0 0 0 0 0 ...
 $ S.aucuparia         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ B.pendula           : int  0 0 1 0 28 0 0 0 0 0 ...
 $ P.abies             : int  0 99 93 0 30 0 0 0 8 99 ...
 $ P.sylvestris        : int  0 0 7 0 1 0 0 0 0 0 ...
 $ F.excelsior         : int  0 0 0 0 38 0 0 0 0 0 ...
 $ A.alba              : int  0 0 0 0 0 4 0 0 0 1 ...
 $ A.platanoides       : int  0 0 0 0 0 2 2 2 0 0 ...
 $ T.cordata           : int  0 0 0 0 0 2 12 0 0 0 ...
 $ S.racemosa          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ U.glabra            : int  0 0 0 0 0 0 0 0 1 0 ...
 $ S.nigra             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ P.alba              : int  0 0 0 0 0 0 0 0 0 0 ...
 $ U.minor             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ S.caprea            : int  0 0 0 0 0 0 0 0 0 0 ...
 $ C.betulus           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ P.nigra             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Q.petraea           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ S.torminalis        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ A.campestre         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ P.strobus           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Q.rubra             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ volAllha            : num  592 377 438 615 194 ...
 $ GiniDBH             : num  0.448 0.416 0.12 0.133 0.378 ...
 $ ShannonIndexTreeSpp : num  0.68 0.07 0.28 0 1.28 0.78 0.54 0.39 0.75 0.06 ...
 $ Tracheophyta_rich   : num  0.218 0.233 0.209 0.189 0.354 ...
 $ Birds_rich          : num  0.358 0.46 0.485 0.307 0.383 ...
 $ Bryophytes_rich     : num  0.0876 0.0876 NA 0.2629 0.3505 ...
 $ Fungi_rich          : num  0.199 0.133 0.114 0.218 0.262 ...
 $ Lichens_rich        : num  0.0659 0.1976 0.1318 0.1537 0.1537 ...
 $ Beetles_rich        : num  0.16 0.16 0.234 0.258 0.221 ...
 $ dendrocoposMajor    : int  1 1 1 0 1 1 1 1 1 1 ...
 $ certhia             : int  0 0 0 0 0 0 1 0 1 0 ...
 $ bryophitaNumObs     : int  1 1 0 3 4 5 7 4 7 1 ...
 $ birdNumObs          : int  14 18 19 12 15 14 12 14 18 18 ...
 $ PlotID              : int  1 2 3 4 5 6 7 8 9 10 ...
```

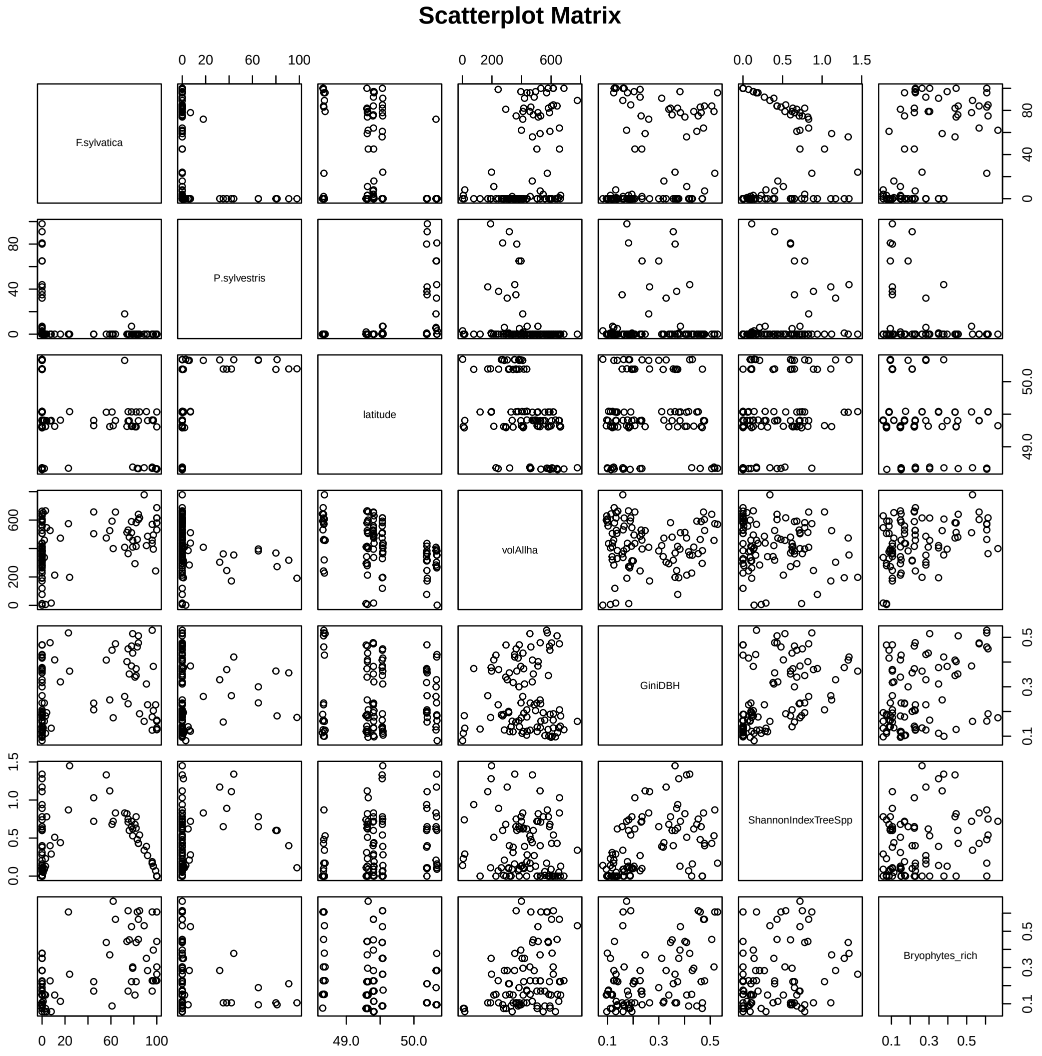
You have a description of each of the variables in the section Section 3.1.1.

You could do further analysis by exploring the data correlations and you can even plot them to explore their values and ranges better. During this process you should start thinking:

- What am I trying to understand with this model? (your question)
- What variables are important to answer my question?

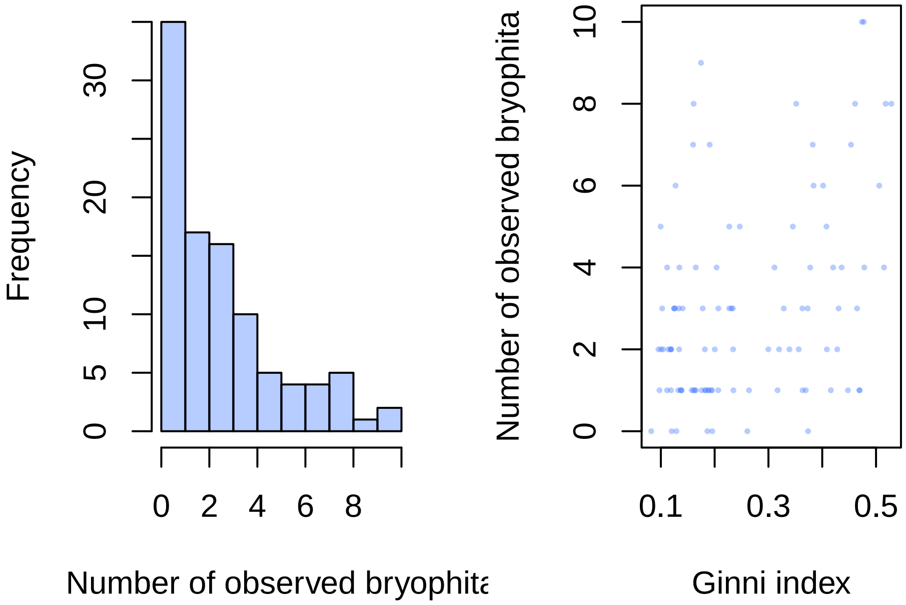
You could for example explore the behavior of the variables by plotting a scatterplot:

```
pairs(~F.sylvatica + P.sylvestris + latitude + volAllha +  GiniDBH + ShannonIndexTreeSpp
      data = observations, main = "Scatterplot Matrix")
```



You can also create individual scatterplots or histograms for a variable of interest, for example:

```
#Divide the screen in 1 line and 3 columns
par(mfrow = c(1, 2), oma = c(0, 2, 0, 0))

#Make the margin around each graph a bit smaller
par(mar = c(4, 4, 2, 2))
# Histogram and Scatterplot
hist(observations$bryophitaNumObs, main = "", breaks = 10,
     col = rgb(0.3, 0.5, 1, 0.4) ,
     xlab = "Number of observed bryophita")
plot(y = observations$bryophitaNumObs,
     x = observations$GiniDBH,
     main = "" , pch = 20, cex = 0.4, col = rgb(0.3, 0.5, 1, 0.4),
     xlab = "Ginni index", ylab = "Number of observed bryophita" )
```



3.4.4 Question 1

- Does a more diverse forest in structure and composition have more Bryophytes species?

3.4.4.1 Fitting a Poisson GLM in R

During your data exploration, you should have selected your response variable, `bryophitaNumObs`. In this case, since we are trying to understand forest structure and composition, you should also choose explanatory variables for that, such as `GiniDBH,` which indicates the forest structural diversity in diameter sizes; higher values indicate more structural heterogeneity, and lower values indicate more homogeneous stands, or the `ShannonIndexTreeSpp` which assess the diversity of tree species in the plot.

We could start by only looking at how the forest structural diversity affects the number of Bryophytes species that we have in a plot. You can create a GLM model with bryophitaNumObs as the response variable and GiniDBH as an explanatory variable. You will use the function glm in R and the family = poisson. You can see how to create and see this model here:

```
bryoModel1 <- glm(bryophitaNumObs ~ GiniDBH,
                  family = poisson,
                  data = observations)

summary(bryoModel1)
```

```

Call:
glm(formula = bryophitaNumObs ~ GiniDBH, family = poisson, data = observations)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)   0.4627     0.1388   3.333 0.000859 ***
GiniDBH       2.2797     0.4221   5.401 6.64e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 188.58  on 98  degrees of freedom
Residual deviance: 159.92  on 97  degrees of freedom
AIC: 424.2

Number of Fisher Scoring iterations: 5
```

Here you can see that the coefficient table produced by a GLM is very similar to a linear model. The intercept tells us the estimated value of the response variable when the continuous explanatory variables (here, just the Gini index) has a value of 0. We then also have coefficients describing the slope of the relationship with our continuous explanatory variables. We can see here that the number of Bryophytes species appears to show a positive relationship with the Gini index, which means that increasing structural diversity in tree diameter sizes has a positive relationship with the number of Bryophytes species in the plot.

We are also interested in understanding the relationship between the number of Bryophytes species and the tree species diversity; we can now try to add this variable into the model and see if it helps us to understand things. You can do that by adding the variable ShannonIndexTreeSpp to the model:

```
bryoModel2 <- glm(bryophitaNumObs ~ GiniDBH + ShannonIndexTreeSpp,
                  family = poisson,
                  data = observations)

summary(bryoModel2)
```

```
Call:
glm(formula = bryophitaNumObs ~ GiniDBH + ShannonIndexTreeSpp,
    family = poisson, data = observations)

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          0.45277    0.14060   3.220  0.00128 **
GiniDBH              2.17411    0.46413   4.684 2.81e-06 ***
ShannonIndexTreeSpp  0.09209    0.16202   0.568  0.56977
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 188.58  on 98  degrees of freedom
Residual deviance: 159.60  on 96  degrees of freedom
AIC: 425.88

Number of Fisher Scoring iterations: 5
```

Here we can see that the variable Shannon index also has a positive relationship with the number of Bryophytes species in the plot, but its effect is much smaller. We can also observe that this variable is not significant. This does not mean it is wrong to add this variable because we want to understand its effect, but keeping this variable in the model depends on what you're trying to do and what "reality" is. Adding variables that are not needed will not help your model (particularly your estimates) but also might not matter much (e.g., predictions). However, removing actual variables can create a useless model even if they don't meet significance.

Variable selection is a long and complicated topic. Some general rules of thumb include: (1) Include the variable if it is of interest, (2) Include the variable if you have some prior knowledge that it should be relevant. This can be misleading because it's a confirmation bias, but in most cases, this makes sense. (3) If you want a model that can generalize to many cases, you should favor fewer variables.

3.4.4.2 Explanatory Power of the model

When we ran linear models, we used the coefficient of determination, or R^2 to assess how much of the variability in our response variable is explained by a given model. R^2 is based on the sums of squares of our model, and so cannot be calculated for GLMs. Instead, we can calculate the deviance explained by our model:

```r
# Extract the null and residual deviance from the model
dev.null <- bryoModel1$null.deviance
dev.resid <- bryoModel1$deviance

# Calculate the deviance explained by the model
dev.explained <- (dev.null - dev.resid)/dev.null

# Round to 3 decimal places
dev.explained <- round(dev.explained, 3)

dev.explained
```

```
[1] 0.152
```

Variability in forest structure (Gini index) explain 15% of the variation in Bryophytes species richness in this study. That is an ok explanatory power for a very simple model of a complex ecological system (many factors determine the species richness for Bryophytes and we are attempting to explain everything with just one variable).

3.4.4.3 Model Assumptions

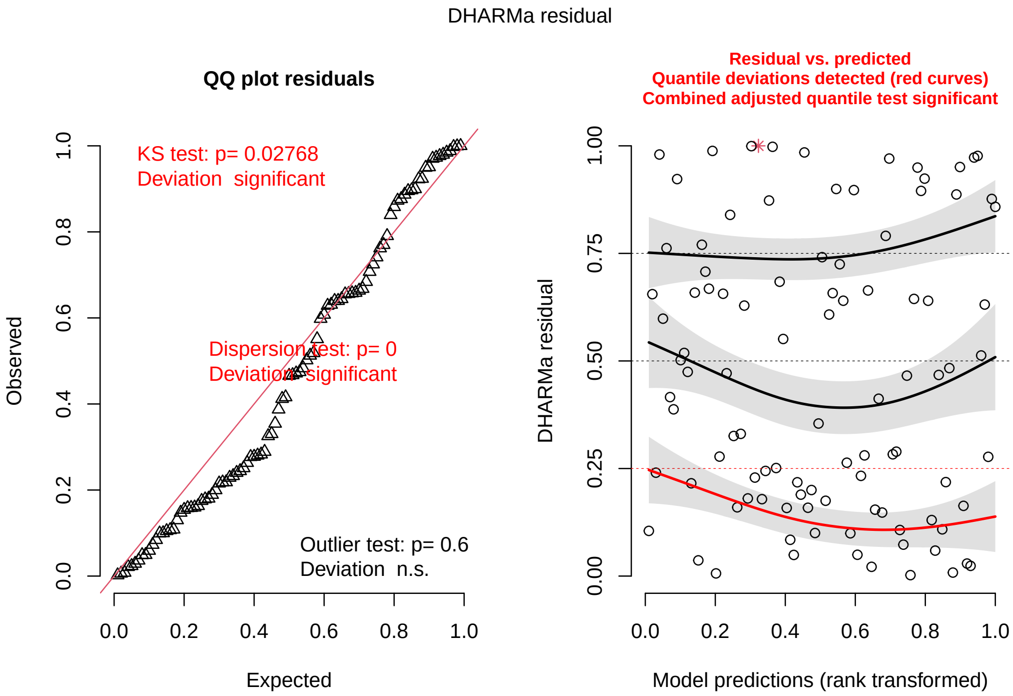
For Poisson GLMs, there is one further assumption that we have not encountered before. If data follow a Poisson distribution, then the mean of the distribution is equal to the variance. Accordingly, a Poisson distribution is represented by just one parameter , which describes both the mean and the variance of the distribution.

Count data in ecology are often **overdispersed**, where the variance is greater than the mean. This violates the assumption of a Poisson GLM, and means that any statistics that we calculate from the model may be unreliable.

We can get an indication of whether a model is over-dispersed by inspecting the model summary. As a rule of thumb, if the response variable conforms to a true Poisson distribution, we expect the residual deviance to be approximately equal to the residual degrees of freedom. If the deviance is much greater than the degrees of freedom, this indicates over-dispersion. This is the case in our models.

To check the model assumptions in a GLM is not as straight forward as with a linear model. This is because classical residuals are not expected to behave in the same way for GLMs. We can use the DHARMa package in R for working with GLMs, which uses a simulation-based approach to compare the residuals from the actual model with the expectation if the model is behaving normally:

```
# Simulate residuals
simResids <- DHARMa::simulateResiduals(bryoModel1)

# Generate plots to compare the model residuals to expectations
plot(simResids)
```



These plots show us that this model is not behaving as we would expect in terms of homogeneity of variance and distribution of residuals. A follow up to this would be to try alternatives to deal with over-dispersed count data in GLMs such as fit a quasi-Poisson GLM or a negative binomial GLM. Unfortunately we do not have time to continuou in this exercise.

3.4.5 Question 2

- Is the number of Briophites species affected by forest management type and the forest structural diversity?

3.4.5.1 Fitting a Poisson GLM in R

Fitting a Poisson GLM in R is very similar to fitting an analysis of covariance (or linear model), except that now we need to use the glm function. To run a GLM, we need to provide one extra piece of information beyond that needed for a linear model: the family of model we want to use. In this case, we want a Poisson family.

We could start by only looking at how the forest structural diversity affects the number Bryophytes species that we have in a plot. You can do this by creating a GLM model which has `bryophitaNumObs` as response variable and `GiniDBH` as explanatory variable. For that you will use the function `glm` in R and use the `family = poisson`. You can see how to create and see this model here:

```r
bryoModel1 <- glm(bryophitaNumObs ~ GiniDBH,
                  family = poisson,
                  data = observations)

summary(bryoModel1)
```

```

Call:
glm(formula = bryophitaNumObs ~ GiniDBH, family = poisson, data = observations)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)   0.4627     0.1388   3.333 0.000859 ***
GiniDBH       2.2797     0.4221   5.401 6.64e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 188.58  on 98  degrees of freedom
Residual deviance: 159.92  on 97  degrees of freedom
AIC: 424.2

Number of Fisher Scoring iterations: 5
```

Here you can see that the coefficient table produced by a GLM is very similar to a linear model. The intercept tells us the estimated value of the response variable when the continuous explanatory variables (here just Gini index) have a value of 0. We then also have coefficients describing the slope of the relationship with our continuous explanatory variables. We can see here that the bryophita numbers appears to show a positive relationship with Gini index, which means that increasing structural diversity in trees diameter sizes has a positive relationship with the number of bryophita species in the plot.

We are also interested in understanding the relationship with of the number of observed Bryophytes species and forest management, we can now try to add this variable into the model.

```
bryoModel2 <- glm(bryophitaNumObs ~ forestManagementType,
                  family = poisson,
                  data = observations)

summary(bryoModel2)
```

```
Call:
glm(formula = bryophitaNumObs ~ forestManagementType, family = poisson,
    data = observations)

Coefficients:
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                               0.9651     0.1543   6.254 3.99e-10
forestManagementTypesimple clearcutting  -0.1335     0.1750  -0.763    0.445
forestManagementTypeunmanaged             0.7633     0.1821   4.192 2.76e-05

(Intercept)                             ***
forestManagementTypesimple clearcutting
forestManagementTypeunmanaged           ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 188.58  on 98  degrees of freedom
Residual deviance: 141.54  on 96  degrees of freedom
AIC: 407.82

Number of Fisher Scoring iterations: 5
```

The intercept here tells us the estimated value of the response variable when the reference groups in our grouping (categorical) variables (here for retention clear-cutting). We then also have coefficients describing the slope of the relationship with our continuous explanatory variables, and coefficients giving the estimated difference in the response variable for non-reference groupings. We can see here that number of bryophites species appears to show a negative relationship with simple clearcutting, and appears to have a positive relationship with unmanaged and retention clear-cutting.

The type simple clearcutting it appears to be no significant, which only tell us about the pairwise differences between the levels. To test whether the categorical predictor, as a whole, is significant is equivalent to testing whether there is any heterogeneity in the means of the levels of the predictor. When there are no other predictors in the model, this is a classical ANOVA problem.

3.4.5.2 Explanatory Power of the model

When we ran linear models, we used the coefficient of determination, or R^2 to assess how much of the variability in our response variable is explained by a given model. R^2 is based on the sums of squares of our model, and so cannot be calculated for GLMs. Instead, we can calculate the the deviance explained by our model:

```r
# Extract the null and residual deviance from the model
dev.null <- bryoModel1$null.deviance
dev.resid <- bryoModel1$deviance

# Calculate the deviance explained by the model
dev.explained <- (dev.null - dev.resid)/dev.null

# Round to 3 decimal places
dev.explained <- round(dev.explained, 3)

dev.explained
```

```
[1] 0.152
```

Variability in forest structure (Gini index) explain 15% of the variation in bryophita species richness in this study system. That is an ok explanatory power for a very simple model of a complex ecological system (many factors determine the species richness for bryophitas and we are attempting to explain everything with one variable).

3.4.5.3 Model Assumptions

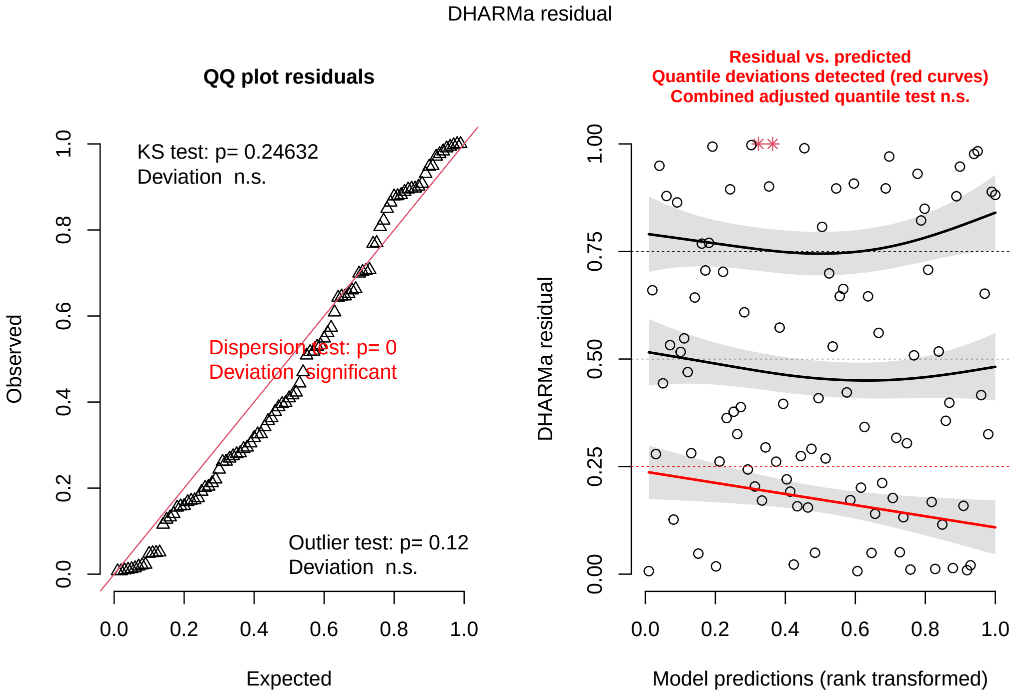
For Poisson GLMs, there is one further assumption that we have not encountered before. If data follow a Poisson distribution, then the mean of the distribution is equal to the variance. Accordingly, a Poisson distribution is represented by just one parameter , which describes both the mean and the variance of the distribution.

Count data in ecology are often ***overdispersed***, where the variance is greater than the mean. This violates the assumption of a Poisson GLM, and means that any statistics that we calculate from the model may be unreliable.

We can get an indication of whether a model is over-dispersed by inspecting the model summary. As a rule of thumb, if the response variable conforms to a true Poisson distribution, we expect the residual deviance to be approximately equal to the residual degrees of freedom. If the deviance is much greater than the degrees of freedom, this indicates over-dispersion. This is the case in our models.

To check the model assumptions in a GLM is not as straight forward as with a linear model. This is because classical residuals are not expected to behave in the same way for GLMs. We can use the DHARMa package in R for working with GLMs, which uses a simulation-based approach to compare the residuals from the actual model with the expectation if the model is behaving normally.

```r
# Simulate residuals
simResids <- DHARMa::simulateResiduals(bryoModel1)

# Generate plots to compare the model residuals to expectations
plot(simResids)
```



These plots show us that this model is not behaving as we would expect in terms of homogeneity of variance and distribution of residuals. A follow up to this would be to try alternatives to deal with over-dispersed count data in GLMs such as fit a quasi-Poisson GLM or a negative binomial GLM. Unfortunately we do not have to continuou in this exercise.

3.4.6 Question 3

- Does a more diverse forest in structure and composition have more bird species?

3.4.6.1 Fitting a Poisson GLM in R

During your data exploration you should have selected your response variable: `bryophitaNumObs`. In this case since we are trying to understand forest structure and composition you should also select explanatory variables for that such as `GiniDBH` which indicates the forest structural diversity in diameter sizes, higher values indicate more structural heterogeneity and lower values indicate more homogeneous stands, or the `ShannonIndexTreeSpp` which assess the diversity of tree species in the plot.

We could start by only looking at how the forest structural diversity affects the number Bryophytes species that we have in a plot. You can do this by creating a GLM model which has `bryophitaNumObs` as response variable and `GiniDBH` as explanatory variable. For that you will use the function `glm`in R and use the `family = poisson`. You can see how to create and see this model here:

```r
birdModel1 <- glm(birdNumObs ~ GiniDBH,
                  family = poisson,
                  data = observations)

summary(birdModel1)
```

```

Call:
glm(formula = birdNumObs ~ GiniDBH, family = poisson, data = observations)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  2.63827    0.05636  46.809   <2e-16 ***
GiniDBH      0.42725    0.19030   2.245   0.0248 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 59.277  on 98  degrees of freedom
Residual deviance: 54.280  on 97  degrees of freedom
AIC: 511.57

Number of Fisher Scoring iterations: 4
```

Here you can see that the coefficient table produced by a GLM is very similar to a linear model. The intercept tells us the estimated value of the response variable when the continuous explanatory variables (here just Gini index) has a value of 0. We then also have coefficients describing the slope of the relationship with our continuous explanatory variables. We can see here that the number of bird species appears to show a positive relationship with Gini index, which means that increasing structural diversity in trees diameter sizes has a positive relationship with the number of bird species in the plot.

We are also interested in understanding the relationship with the number of bird species and the trees species diversity, we can now try to add this variable into the model and see if it help us to understand things. You can do that by adding the variable `ShannonIndexTreeSpp` into the model

```
birdModel2 <- glm(birdNumObs ~ GiniDBH + ShannonIndexTreeSpp,
                  family = poisson,
                  data = observations)

summary(birdModel2)
```

```

Call:
glm(formula = birdNumObs ~ GiniDBH + ShannonIndexTreeSpp, family = poisson,
    data = observations)

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          2.63499    0.05665  46.513   <2e-16 ***
GiniDBH              0.33323    0.21661   1.538    0.124
ShannonIndexTreeSpp  0.06923    0.07444   0.930    0.352
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 59.277  on 98  degrees of freedom
Residual deviance: 53.419  on 96  degrees of freedom
AIC: 512.71

Number of Fisher Scoring iterations: 4
```

Here we can see that the variable Shannon index also has a positive relationship with the number of bird species in the plot but its effect is much smaller. We can also observed that this variable it is not significant and that including this variable also made Gini index variable not significant. This does not mean that is wrong to add this variable, because we want to understand its effect, but keeping this variable in the model or not it depends on what you're trying to do, and what "reality" is. Adding variables that are not needed will not help your model (particularly your estimates), but also might not matter much (e.g. predictions). However, removing variables that are real, even if they don't meet significance, can create a useless model.

Variable selection is a long and complicated topic. some general rules of thumb include: (1) Include the variable if it is of interest, (2) Include the variable if you have some prior knowledge that it should be relevant. This can be misleading, because it's a confirmation bias, but in most cases this makes sense. (3) If you want a model that can generalize to many cases, you should favor fewer variables.

3.4.6.2 Explanatory Power of the model

When we ran linear models, we used the coefficient of determination, or R^2 to assess how much of the variability in our response variable is explained by a given model. R^2 is based on the sums of squares of our model, and so cannot be calculated for GLMs. Instead, we can calculate the the deviance explained by our model:

```
# Extract the null and residual deviance from the model
dev.null <- birdModel1$null.deviance
dev.resid <- birdModel1$deviance

# Calculate the deviance explained by the model
dev.explained <- (dev.null - dev.resid)/dev.null

# Round to 3 decimal places
dev.explained <- round(dev.explained, 3)

dev.explained
```

```
[1] 0.084
```

Variability in forest structure (Gini index) explain 8% of the variation in bird species richness in this study. That is an ok explanatory power for a very simple model of a complex ecological system (many factors determine the species richness for birds and we are attempting to explain everything with just one variable).

3.4.6.3 Model Assumptions

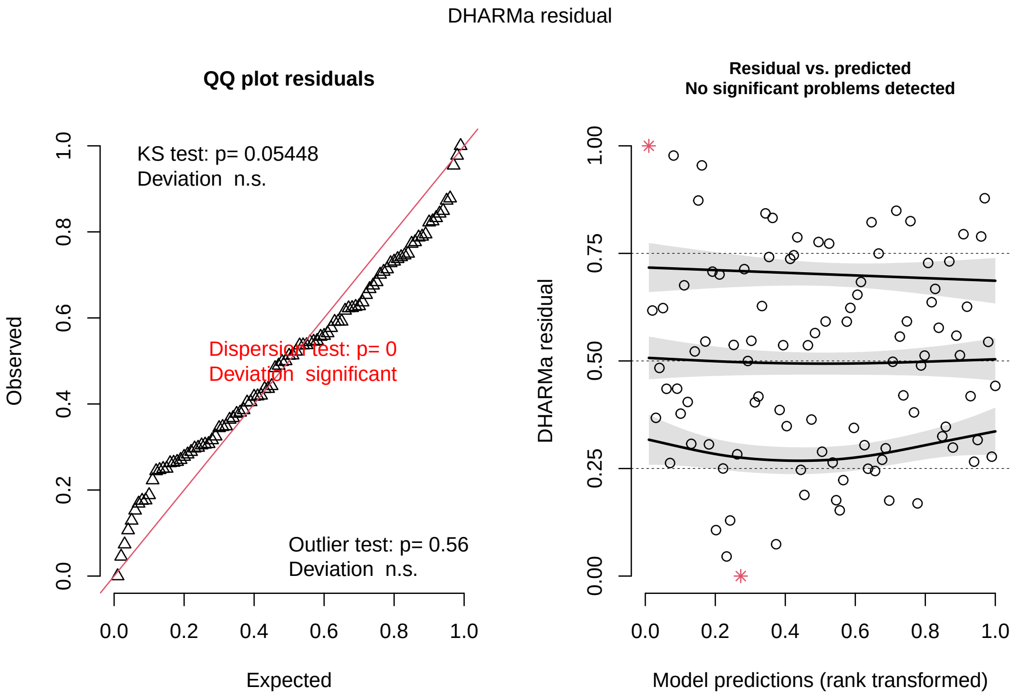
For Poisson GLMs, there is one further assumption that we have not encountered before. If data follow a Poisson distribution, then the mean of the distribution is equal to the variance. Accordingly, a Poisson distribution is represented by just one parameter , which describes both the mean and the variance of the distribution.

Count data in ecology are often **overdispersed**, where the variance is greater than the mean. This violates the assumption of a Poisson GLM, and means that any statistics that we calculate from the model may be unreliable.

We can get an indication of whether a model is over-dispersed by inspecting the model summary. As a rule of thumb, if the response variable conforms to a true Poisson distribution, we expect the residual deviance to be approximately equal to the residual degrees of freedom. If the deviance is much greater than the degrees of freedom, this indicates over-dispersion. This is the case in our models.

To check the model assumptions in a GLM is not as straight forward as with a linear model. This is because classical residuals are not expected to behave in the same way for GLMs. We can use the DHARMa package in R for working with GLMs, which uses a simulation-based approach to compare the residuals from the actual model with the expectation if the model is behaving normally:

```r
# Simulate residuals
simResids <- DHARMa::simulateResiduals(birdModel1)

# Generate plots to compare the model residuals to expectations
plot(simResids)
```



These plots show us that this model is not behaving as we would expect in terms of homogeneity of variance and distribution of residuals. A follow up to this would be to try alternatives to deal with over-dispersed count data in GLMs such as fit a quasi-Poisson GLM or a negative binomial GLM. Unfortunately we do not have time to continuou in this exercise.

3.4.7 Question 4

- Is the number of bird species affected by forest management type and the forest structural diversity?

3.4.7.1 Fitting a Poisson GLM in R

Fitting a Poisson GLM in R is very similar to fitting an analysis of covariance (or linear model), except that now we need to use the glm function. To run a GLM, we need to provide one extra piece of information beyond that needed for a linear model: the family of model we want to use. In this case, we want a Poisson family.

We could start by only looking at how the forest structural diversity affects the number Bryophytes species that we have in a plot. You can do this by creating a GLM model which has `birdNumObs` as response variable and `GiniDBH` as explanatory variable. For that you will use the function `glm` in R and use the `family = poisson`. You can see how to create and see this model here:

```
birdModel1 <- glm(birdNumObs ~ GiniDBH,
                  family = poisson,
                  data = observations)

summary(birdModel1)
```

```

Call:
glm(formula = birdNumObs ~ GiniDBH, family = poisson, data = observations)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  2.63827    0.05636  46.809   <2e-16 ***
GiniDBH      0.42725    0.19030   2.245   0.0248 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 59.277  on 98  degrees of freedom
Residual deviance: 54.280  on 97  degrees of freedom
AIC: 511.57

Number of Fisher Scoring iterations: 4
```

Here you can see that the coefficient table produced by a GLM is very similar to a linear model. The intercept tells us the estimated value of the response variable when the continuous explanatory variables (here just Gini index) have a value of 0. We then also have coefficients describing the slope of the relationship with our continuous explanatory variables. We can see here that the bird numbers appears to show a positive relationship with Gini index, which means that increasing structural diversity in trees diameter sizes has a positive relationship with the number of bird species in the plot.

We are also interested in understanding the relationship with of the number of observed bird species and forest management, we can now try to add this variable into the model.

```r
birdModel2 <- glm(birdNumObs ~ forestManagementType,
                  family = poisson,
                  data = observations)

summary(birdModel2)
```

```

Call:
glm(formula = birdNumObs ~ forestManagementType, family = poisson,
    data = observations)

Coefficients:
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                              2.83688    0.06052  46.873   <2e-16
forestManagementTypesimple clearcutting -0.11744    0.06850  -1.714   0.0865
forestManagementTypeunmanaged           -0.06429    0.08338  -0.771   0.4407

(Intercept)                             ***
forestManagementTypesimple clearcutting .
forestManagementTypeunmanaged
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 59.277  on 98  degrees of freedom
Residual deviance: 56.205  on 96  degrees of freedom
AIC: 515.49

Number of Fisher Scoring iterations: 4
```

The intercept here tells us the estimated value of the response variable when the reference groups in our grouping (categorical) variables (here for retention clear-cutting). We then also have coefficients describing the slope of the relationship with our continuous explanatory variables, and coefficients giving the estimated difference in the response variable for non-reference groupings. We can see here that number of bird species appears to show a negative relationship with simple clearcutting, unmanaged and a positive relationship with retention clear-cutting.

The type simple clearcutting and unmanaged it appears to be no significant, which only tell us about the pairwise differences between the levels. To test whether the categorical predictor, as a whole, is significant is equivalent to testing whether there is any heterogeneity in the means of the levels of the predictor. When there are no other predictors in the model, this is a classical ANOVA problem.

3.4.7.2 Explanatory Power of the model

When we ran linear models, we used the coefficient of determination, or R^2 to assess how much of the variability in our response variable is explained by a given model. R^2 is based on the sums of squares of our model, and so cannot be calculated for GLMs. Instead, we can calculate the the deviance explained by our model:

```r
# Extract the null and residual deviance from the model
dev.null <- birdModel1$null.deviance
dev.resid <- birdModel1$deviance

# Calculate the deviance explained by the model
dev.explained <- (dev.null - dev.resid)/dev.null

# Round to 3 decimal places
dev.explained <- round(dev.explained, 3)

dev.explained
```

```
[1] 0.084
```

Variability in forest structure (Gini index) explain 15% of the variation in bryophita species richness in this study system. That is an ok explanatory power for a very simple model of a complex ecological system (many factors determine the species richness for bryophitas and we are attempting to explain everything with one variable).

3.4.7.3 Model Assumptions

For Poisson GLMs, there is one further assumption that we have not encountered before. If data follow a Poisson distribution, then the mean of the distribution is equal to the

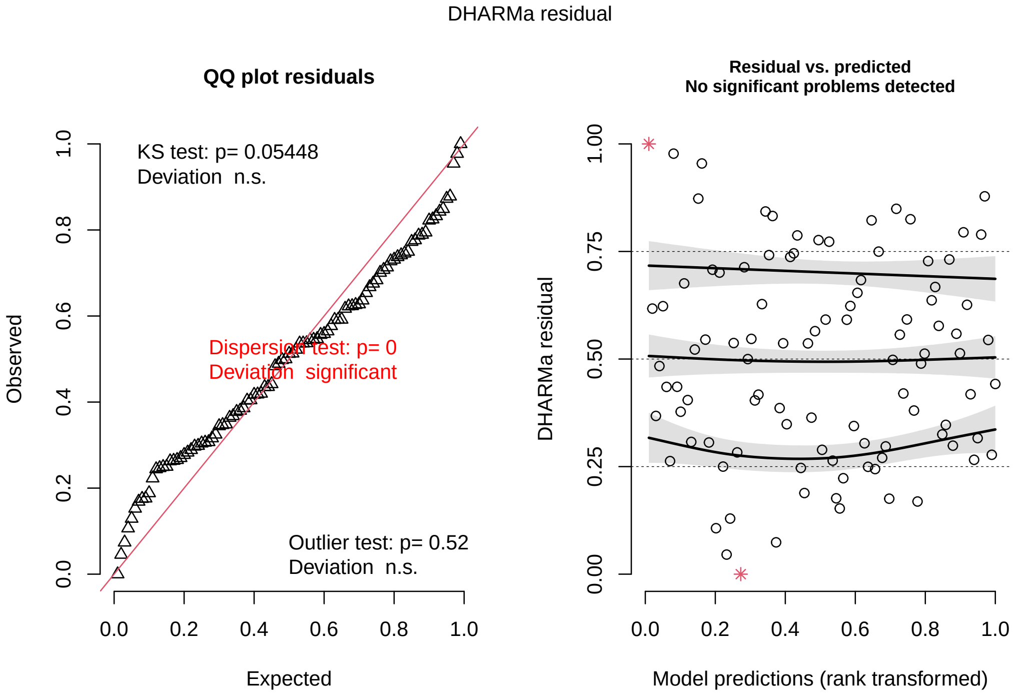
variance. Accordingly, a Poisson distribution is represented by just one parameter ,
which describes both the mean and the variance of the distribution.

Count data in ecology are often ***overdispersed***, where the variance is greater than the mean. This violates the assumption of a Poisson GLM, and means that any statistics that we calculate from the model may be unreliable.

We can get an indication of whether a model is over-dispersed by inspecting the model summary. As a rule of thumb, if the response variable conforms to a true Poisson distribution, we expect the residual deviance to be approximately equal to the residual degrees of freedom. If the deviance is much greater than the degrees of freedom, this indicates over-dispersion. This is the case in our models.

To check the model assumptions in a GLM is not as straight forward as with a linear model. This is because classical residuals are not expected to behave in the same way for GLMs. We can use the DHARMa package in R for working with GLMs, which uses a simulation-based approach to compare the residuals from the actual model with the expectation if the model is behaving normally.

```
# Simulate residuals
simResids <- DHARMa::simulateResiduals(birdModel1)

# Generate plots to compare the model residuals to expectations
plot(simResids)
```



These plots show us that this model is not behaving as we would expect in terms of homogeneity of variance and distribution of residuals. A follow up to this would be to try alternatives to deal with over-dispersed count data in GLMs such as fit a quasi-Poisson GLM or a negative binomial GLM. Unfortunately we do not have to continuou in this exercise.

3.4.8 Question 5

- Is the presence of the Great spotted woodpecker affected by forest density?

3.4.8.1 Fitting a BRT in R

In this case we want to assess the occurrence of certain species across the plots. In other words, we want to assess what is the probability of a certain species with biodiversity interest to be present in a plot based on the variables that describe the forest of that plot. In this case the response variable is `dendrocoposMajor` that represents if the Great spotted woodpecker has been observed in this plot or not.

Then you need to select some variables of interest, after you have explored the data you can decide which variables you want to use to fit this model. We are proposing to select the following variables:

- `latitude` as proxy for plot location or/and climate
- `forestManagementType` to assess if different management types have different impact in the presence / absence of Parus major.
- `volAllha` that is the total volume in the plot, as a proxy of how dense the plot is. Higher volumes will mean that the forest is more dense.
- `GiniDBH` showing how homogeneous the plot is in trees diameters. A value closer to 1 will mean that indicate more structural heterogeneity, lower values indicate more homogeneous plots.
- `Birds_rich` are the species richness value calculated for all bird species species, higher value indicate higher species richness for birds.

You can create a vector `selVar` in which you add the names of the selected variables. Then you only take those variables from the data that you will use to create the model.

```
# Select variables from the dataset for the model
selVar <- c("dendrocoposMajor", "latitude","forestManagementType",
            "volAllha", "GiniDBH",  "ShannonIndexTreeSpp", "Birds_rich")

# Filter the dataset to the selected variables
modelDataSel <- observations[, colnames(observations) %in% selVar]
```

Unfortunately the amount of that we have in this dataset it is not enough to fit a BRT model for these variables. We are going to do an obviously wrong thing for the shake of being able to demonstrate how to fit a BRT model. In the next code you are going to repeat the same dataset multiple times:

```
modelDataSel <- rbind(modelDataSel, modelDataSel, modelDataSel, modelDataSel,
                      modelDataSel)
```

Now it is important to assess if the variables have the right categories. Variables should be type numeric or factor.

```
summary(modelDataSel)
```

```
    latitude     forestManagementType    volAllha          GiniDBH
 Min.   :48.65   Length:495           Min.   :  1.681   Min.   :0.08209
 1st Qu.:49.31   Class :character     1st Qu.:319.920   1st Qu.:0.13684
 Median :49.40   Mode  :character     Median :434.729   Median :0.20358
 Mean   :49.49                        Mean   :425.694   Mean   :0.25683
 3rd Qu.:50.19                        3rd Qu.:558.355   3rd Qu.:0.37368
 Max.   :50.34                         Max.   :777.882   Max.   :0.52852
 ShannonIndexTreeSpp   Birds_rich     dendrocoposMajor
 Min.   :0.000       Min.   :0.1694   Min.   :0.0000
 1st Qu.:0.060       1st Qu.:0.3995   1st Qu.:0.0000
 Median :0.280       Median :0.4679   Median :1.0000
 Mean   :0.392       Mean   :0.4698   Mean   :0.7071
 3rd Qu.:0.680       3rd Qu.:0.5089   3rd Qu.:1.0000
 Max.   :1.450       Max.   :0.8360   Max.   :1.0000
```

```
# Two variables are character, we assign to factor instead:
modelDataSel$forestManagementType <- as.factor(modelDataSel$forestManagementType)
```

In the next step you can see how you can run the model with the selected variables and model parameters. You have a description of the models parameters in the Section 3.3.2 . In this example we are going to use the default parameters for the calibration, where learning rate = 0.01 and tree complexity = 1 and cross-validation = 10-fold. However, the bag fraction is changed from the default value, 0.75, to 0.5. As a family we used the Bernoulli family, because we are predicting presence/absence per plot. These data have 495 plots, comprising 350 presence records for the Great spotted woodpecker. You can check these numbers by doing:

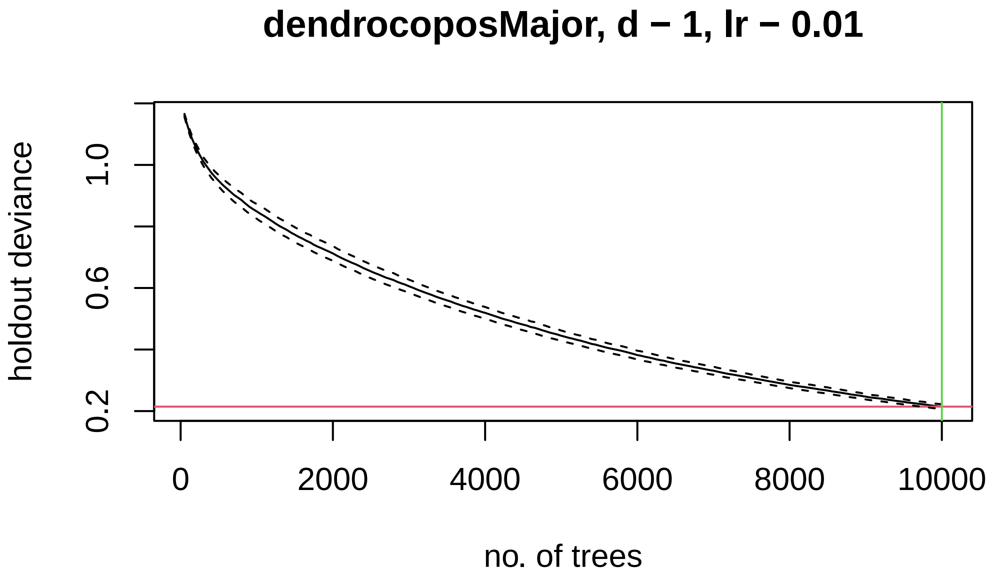
```
table(modelDataSel$dendrocoposMajor)
```

```

  0   1 
145 350 
```

As a first guess you could decide there are enough data to model interactions of reasonable complexity, and a lr of about 0.01 could be a reasonable starting point. You can use the model creation function that steps forward and identifies the optimal number of trees (nt) by doing this:

```
family <- "bernoulli"

    tc = 1    # tree complexity
    lr = 0.01 # learning rate-shrinkage
    bag = 0.5 # bag fraction

    modelBRT <- dismo::gbm.step(data = modelDataSel,
                          #indices of predictor variables in data
                          gbm.x = 1:6,
                          #index of response variable in data:
```

```
                          gbm.y = 7,
                          family = family,
                          tree.complexity = tc,
                          learning.rate = lr,
                          bag.fraction = bag)
```



Running a model such as that described above writes progress reports to the screen, makes a graph, and returns an object containing a number of components. The R console results reports a brief model summary all the values are also retained in the model object.

The model is built with the default 10-fold cross-validation (CV). In the plotted graph the solid black curve is the mean, and the dotted curves 1 standard error, for the changes in predictive deviance (i.e., as measured on the excluded folds of the CV). The red line shows the minimum of the mean, and the green line the number of trees at which that occurs. The final model that is returned in the model object is built on the full data set, using the number of trees identified as optimal.

Ideally, you should invest time in modifying the parameters and finr the parameters that provide the models with the minimum deviance resulting from the best combination of bag, tree complexity and learning rate values. For the shake of limited timing, we will only test here the default values.

3.4.8.2 Model behaviour

You can summarized the model parameters used and the cross validation statistics from the fitted model by doing this:

```
# We make a table with the summary statistics
 results <- data.frame(

# Model parameters
Tree.Complexity = modelBRT$gbm.call$tree.complexity,
Learning.Rate = modelBRT$gbm.call$learning.rate,
                 Bag.Fraction = modelBRT$gbm.call$bag.fraction,
                 Interaction.depth = modelBRT$interaction.depth,
                 Shrinkage = modelBRT$shrinkage,
                 N.trees = modelBRT$n.trees,

# Cross validation statistics

## mean total deviance
Deviance = modelBRT$self.statistics$mean.resid, # mean residual deviance

AUC = modelBRT$self.statistics$discrimination, # training data AUC score

Corr = modelBRT$self.statistics$correlation,   # training data correlation

## Cross Validation statistics

# We calculate each statistic within each fold (at the identified optimal number
# of trees that is calculated on the mean change in predictive deviance over all folds),
#then present here the mean and standard error of those fold-based statistics.

devianceCV = modelBRT$cv.statistics$deviance.mean,  # estimated cv deviance
devianceCVse = modelBRT$cv.statistics$deviance.se,  # estimated cv deviance se

CorrCV = modelBRT$cv.statistics$correlation.mean,  #cv correlation
CorrCVse = modelBRT$cv.statistics$correlation.se,  #cv correlation se

AUCcv = modelBRT$cv.statistics$discrimination.mean, # cv AUC score
AUCcvSE = modelBRT$cv.statistics$discrimination.se)  # cv AUC score se


print(t(results))
```

```
                          [,1]
Tree.Complexity   1.000000e+00
Learning.Rate     1.000000e-02
Bag.Fraction      5.000000e-01
Interaction.depth 1.000000e+00
Shrinkage         1.000000e-02
N.trees           1.000000e+04
Deviance          1.767121e-01
AUC               1.000000e+00
Corr              9.836141e-01
devianceCV        2.143476e-01
devianceCVse      7.822666e-03
CorrCV            9.726135e-01
CorrCVse          3.554647e-03
AUCcv             1.000000e+00
AUCcvSE           0.000000e+00
```

3.4.8.3 Model output analysis

We can look at the relative contribution of each of the predictor variables. The measures are based on the number of time the variable is selected for splitting, weighted by the improvement of the model as a result of each split averaged across all trees. The relative contribution of each of the variables is scaled so the sum is 100%, with higher numbers indicating stronger influence in the response.

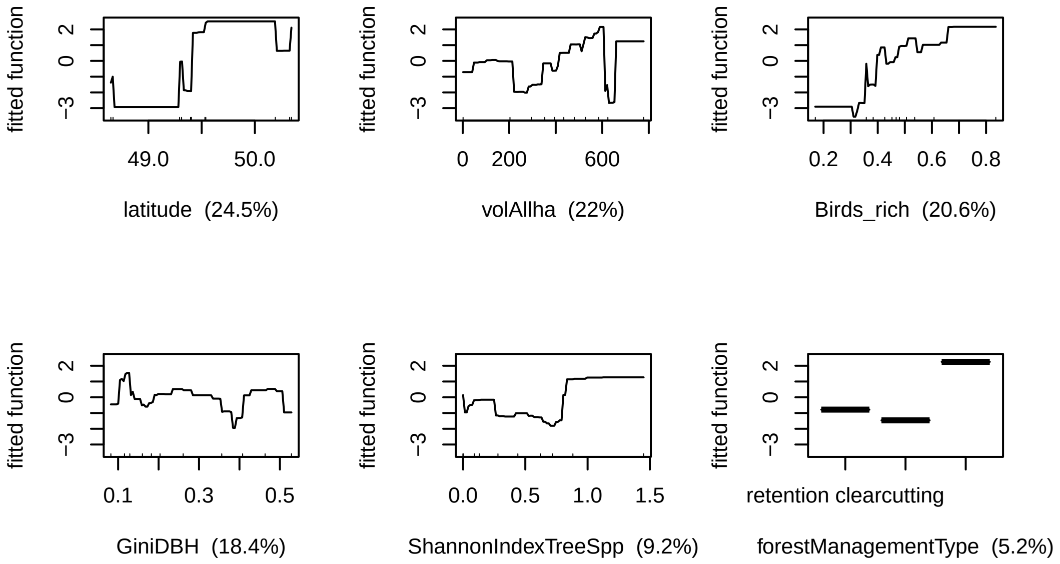
```
# Variables contribution
modelBRT$contributions
```

```
                                       var   rel.inf
latitude                          latitude 24.453824
volAllha                          volAllha 22.017043
Birds_rich                      Birds_rich 20.611369
GiniDBH                            GiniDBH 18.445519
ShannonIndexTreeSpp   ShannonIndexTreeSpp  9.223416
forestManagementType forestManagementType  5.248830
```

Here we can see that the two variables with the highest influence in the response are `latitude`, and `volAllhsa`.

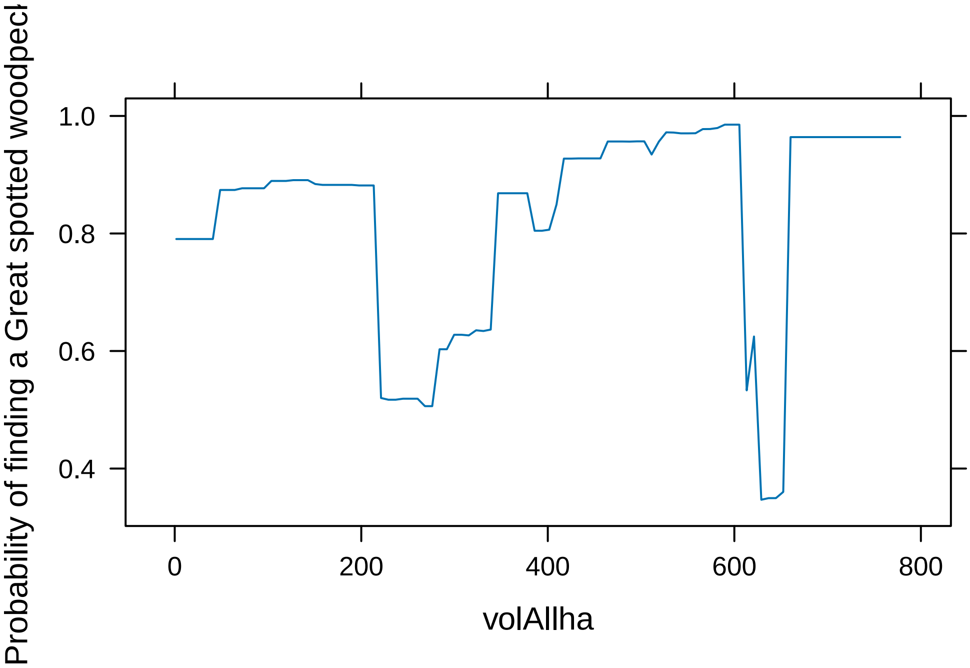
Now we can evaluate the model behavior via partial dependence plots, showing the effect of each of the variables on the response by accounting for the average effects of all other predictors in the model:

```
dismo::gbm.plot(modelBRT, n.plots = 6,
                plot.layout = c(2, 3), write.title = F)
```



In this partial dependence plots the predictions are on the scale of f(x). In this case, for the Bernoulli loss the returned value is on the log odds scale. You can see how this plot will look by plotting with the function from the package `gbm` and using the type "response". Since we are interesting in finding out if the forest density has an impact in the presence of the Great spotted woodpecker we can have a look to the plot density variable `volAllha` :

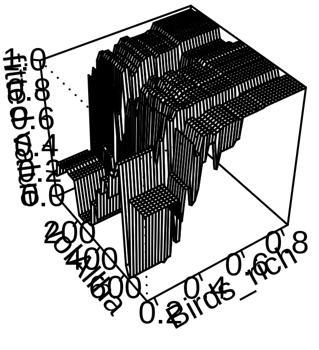
```
gbm::plot.gbm(modelBRT, i.var = 3, type = "response", ylab = "Probability of finding a G
```



It seems that there is an increase in probability of finding a Great spotted woodpecker with higher forest densities, but the trend it is not very clear. We could also analyse the interaction effects, of density and for example overall bird richness. The model predictions can be obtained for each pair of predictor variables, setting all other predictors to their means.

To plot this pairwise interactions we have to do:

```
dismo::gbm.perspec(modelBRT, 3, 6)
```



Here we can see that both increasing bird diversity and forest density provides the highest probabilities for finding the Great spotted woodpecker.

3.5 Question 6

- Is the presence of the Great spotted woodpecker affected by forest diversity?

3.5.0.1 Fitting a BRT in R

In this case we want to assess the occurrence of certain species across the plots. In other words, we want to assess what is the probability of a certain species with biodiversity interest to be present in a plot based on the variables that describe the forest of that plot. In this case the response variable is `dendrocoposMajor` that represents if the Great spotted woodpecker has been observed in this plot or not.

Then you need to select some variables of interest, after you have explored the data you can decide which variables you want to use to fit this model. We are proposing to select the following variables:

- `latitude` as proxy for plot location or/and climate

- forestManagementType to assess if different management types have different impact in the presence / absence of Parus major.
- volAllha that is the total volume in the plot, as a proxy of how dense the plot is. Higher volumes will mean that the forest is more dense.
- GiniDBH showing how homogeneous the plot is in trees diameters. A value closer to 1 will mean that indicate more structural heterogeneity, lower values indicate more homogeneous plots.
- Birds_rich are the species richness value calculated for all bird species species, higher value indicate higher species richness for birds.

You can create a vector selVar in which you add the names of the selected variables. Then you only take those variables from the data that you will use to create the model.

```
# Select variables from the dataset for the model
selVar <- c("dendrocoposMajor", "latitude","forestManagementType",
            "volAllha","GiniDBH",  "ShannonIndexTreeSpp", "Birds_rich")

# Filter the dataset to the selected variables
modelDataSel <- observations[, colnames(observations) %in% selVar]
```

Unfortunately the amount of that we have in this dataset it is not enough to fit a BRT model for these variables. We are going to do an obviously wrong thing for the shake of being able to demonstrate how to fit a BRT model. In the next code you are going to repeat the same dataset multiple times:

```
modelDataSel <- rbind(modelDataSel, modelDataSel, modelDataSel, modelDataSel,
                      modelDataSel)
```

Now it is important to assess if the variables have the right categories. Variables should be type numeric or factor.

```
summary(modelDataSel)
```

```
    latitude     forestManagementType    volAllha          GiniDBH
 Min.   :48.65   Length:495           Min.   :  1.681   Min.   :0.08209
 1st Qu.:49.31   Class :character     1st Qu.:319.920   1st Qu.:0.13684
 Median :49.40   Mode  :character     Median :434.729   Median :0.20358
 Mean   :49.49                        Mean   :425.694   Mean   :0.25683
 3rd Qu.:50.19                        3rd Qu.:558.355   3rd Qu.:0.37368
 Max.   :50.34                        Max.   :777.882   Max.   :0.52852
 ShannonIndexTreeSpp   Birds_rich     dendrocoposMajor
 Min.   :0.000       Min.   :0.1694   Min.   :0.0000  
 1st Qu.:0.060       1st Qu.:0.3995   1st Qu.:0.0000  
 Median :0.280       Median :0.4679   Median :1.0000  
 Mean   :0.392       Mean   :0.4698   Mean   :0.7071  
 3rd Qu.:0.680       3rd Qu.:0.5089   3rd Qu.:1.0000  
 Max.   :1.450       Max.   :0.8360   Max.   :1.0000  
```

```r
# Two variables are character, we assign to factor instead:
modelDataSel$forestManagementType <- as.factor(modelDataSel$forestManagementType)
```

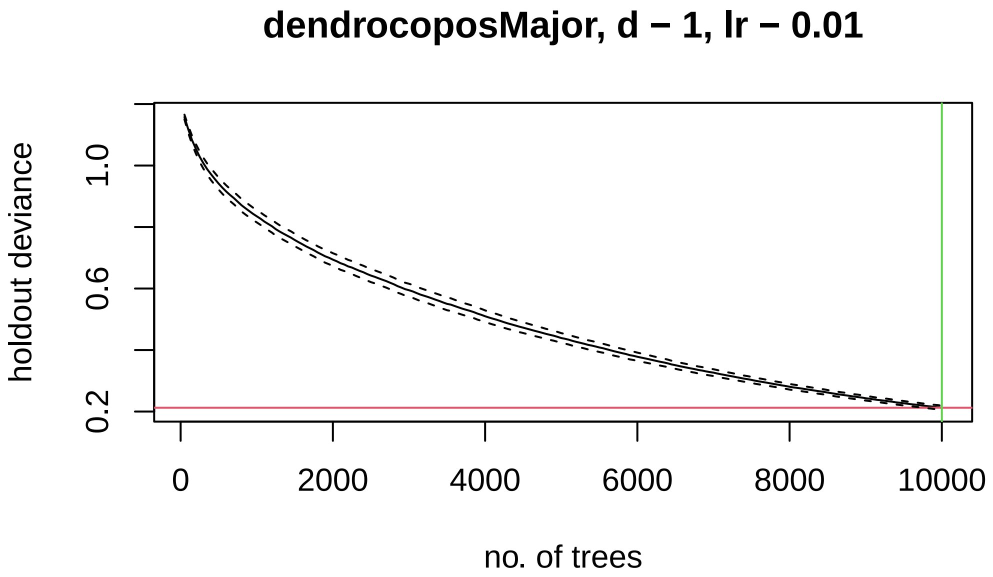
In the next step you can see how you can run the model with the selected variables and model parameters. You have a description of the models parameters in the Section 3.3.2 . In this example we are going to use the default parameters for the calibration, where learning rate = 0.01 and tree complexity = 1 and cross-validation = 10-fold. However, the bag fraction is changed from the default value, 0.75, to 0.5. As a family we used the Bernoulli family, because we are predicting presence/absence per plot. These data have 495 plots, comprising 350 presence records for the Great spotted woodpecker. You can check these numbers by doing:

```r
table(modelDataSel$dendrocoposMajor)
```

```

  0   1 
145 350 
```

As a first guess you could decide there are enough data to model interactions of reasonable complexity, and a lr of about 0.01 could be a reasonable starting point. You can use the model creation function that steps forward and identifies the optimal number of trees (nt) by doing this:

```r
family <- "bernoulli"

    tc = 1    # tree complexity
    lr = 0.01 # learning rate-shrinkage
    bag = 0.5 # bag fraction

    modelBRT <- dismo::gbm.step(data = modelDataSel,
                          #indices of predictor variables in data
                          gbm.x = 1:6,
                          #index of response variable in data:
                          gbm.y = 7,
                          family = family,
                          tree.complexity = tc,
                          learning.rate = lr,
                          bag.fraction = bag)
```



Running a model such as that described above writes progress reports to the screen, makes a graph, and returns an object containing a number of components. The R console results reports a brief model summary all the values are also retained in the model object.

The model is built with the default 10-fold cross-validation (CV). In the plotted graph the solid black curve is the mean, and the dotted curves 1 standard error, for the changes in predictive deviance (i.e., as measured on the excluded folds of the CV). The red line shows the minimum of the mean, and the green line the number of trees at which that occurs. The final model that is returned in the model object is built on the full data set, using the number of trees identified as optimal.

Ideally, you should invest time in modifying the parameters and finr the parameters that provide the models with the minimum deviance resulting from the best combination of bag, tree complexity and learning rate values. For the shake of limited timing, we will only test here the default values.

3.5.0.2 Model behaviour

You can summarized the model parameters used and the cross validation statistics from the fitted model by doing this:

```
# We make a table with the summary statistics
 results <- data.frame(

# Model parameters
Tree.Complexity = modelBRT$gbm.call$tree.complexity,
Learning.Rate = modelBRT$gbm.call$learning.rate,
                 Bag.Fraction = modelBRT$gbm.call$bag.fraction,
                 Interaction.depth = modelBRT$interaction.depth,
                 Shrinkage = modelBRT$shrinkage,
                 N.trees = modelBRT$n.trees,

# Cross validation statistics

## mean total deviance
Deviance = modelBRT$self.statistics$mean.resid, # mean residual deviance

AUC = modelBRT$self.statistics$discrimination, # training data AUC score

Corr = modelBRT$self.statistics$correlation,   # training data correlation

## Cross Validation statistics

# We calculate each statistic within each fold (at the identified optimal number
# of trees that is calculated on the mean change in predictive deviance over all folds),
#then present here the mean and standard error of those fold-based statistics.

devianceCV = modelBRT$cv.statistics$deviance.mean,  # estimated cv deviance
devianceCVse = modelBRT$cv.statistics$deviance.se,  # estimated cv deviance se

CorrCV = modelBRT$cv.statistics$correlation.mean,  #cv correlation
CorrCVse = modelBRT$cv.statistics$correlation.se,  #cv correlation se

AUCcv = modelBRT$cv.statistics$discrimination.mean, # cv AUC score
AUCcvSE = modelBRT$cv.statistics$discrimination.se)  # cv AUC score se


print(t(results))
```

```
                           [,1]
Tree.Complexity   1.000000e+00
Learning.Rate     1.000000e-02
Bag.Fraction      5.000000e-01
Interaction.depth 1.000000e+00
Shrinkage         1.000000e-02
N.trees           1.000000e+04
Deviance          1.758584e-01
AUC               1.000000e+00
Corr              9.841562e-01
devianceCV        2.122778e-01
devianceCVse      6.677707e-03
CorrCV            9.749094e-01
CorrCVse          2.810924e-03
AUCcv             1.000000e+00
AUCcvSE           0.000000e+00
```

3.5.0.3 Model output analysis

We can look at the relative contribution of each of the predictor variables. The measures are based on the number of time the variable is selected for splitting, weighted by the improvement of the model as a result of each split averaged across all trees. The relative contribution of each of the variables is scaled so the sum is 100%, with higher numbers indicating stronger influence in the response.

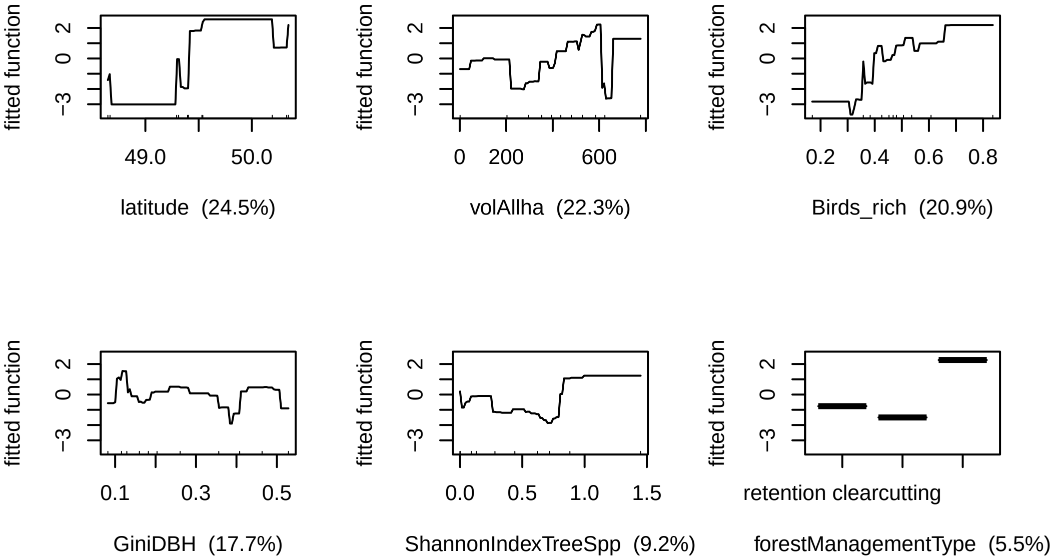
```
# Variables contribution
modelBRT$contributions
```

```
                                       var   rel.inf
latitude                          latitude 24.451265
volAllha                          volAllha 22.281773
Birds_rich                      Birds_rich 20.874933
GiniDBH                            GiniDBH 17.686307
ShannonIndexTreeSpp   ShannonIndexTreeSpp  9.222782
forestManagementType forestManagementType  5.482939
```

Here we can see that the two variables with the highest influence in the response are `latitude`, and `volAllhsa`.

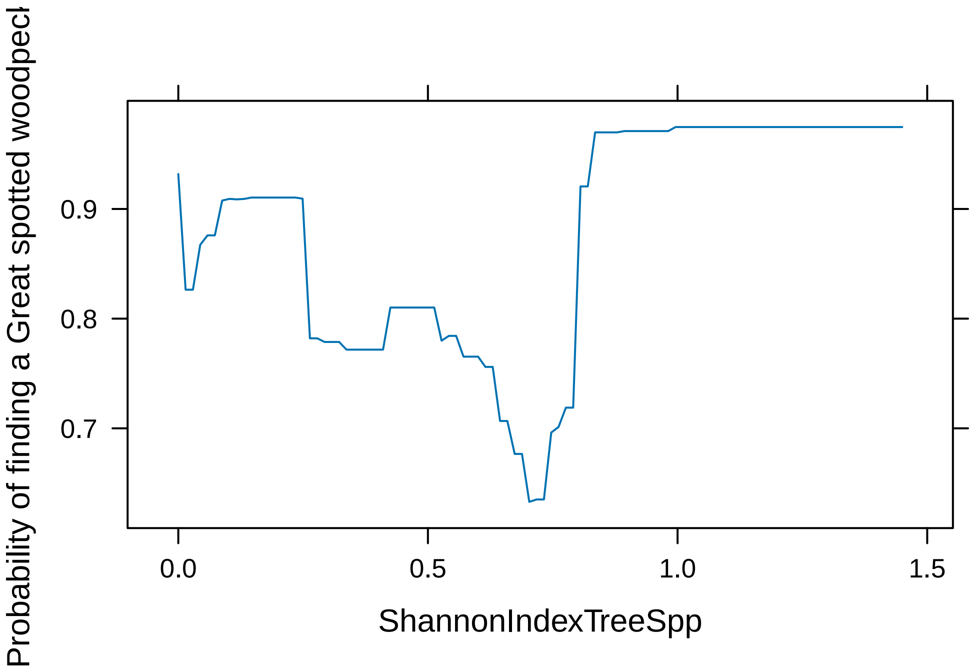
Now we can evaluate the model behavior via partial dependence plots, showing the effect of each of the variables on the response by accounting for the average effects of all other predictors in the model:

```
    dismo::gbm.plot(modelBRT, n.plots = 6,
             plot.layout = c(2, 3), write.title = F)
```



In this partial dependence plots the predictions are on the scale of f(x). In this case, for the Bernoulli loss the returned value is on the log odds scale. You can see how this plot will look by plotting with the function from the package `gbm` and using the type "response". Since we are interesting in finding out if the forest diversity has an impact in the presence of the Great spotted woodpecker we can have a look to the plot density variable `volAllha` :

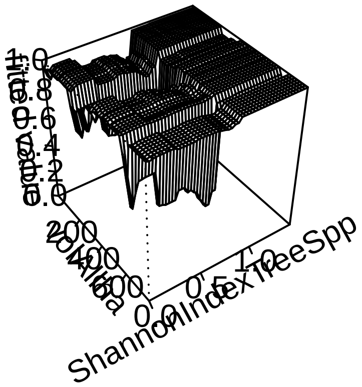
```
gbm::plot.gbm(modelBRT, i.var = 5, type = "response", ylab = "Probability of finding a Gr
```



It seems that there is an increase in probability of finding a Great spotted woodpecker with higher forest densities, but the trend it is not very clear. We could also analyse the interaction effects, of forest diversity and for example forest density. The model predictions can be obtained for each pair of predictor variables, setting all other predictors to their means.

To plot this pairwise interactions we have to do:

```
dismo::gbm.perspec(modelBRT, 3, 5)
```



Here we can not see a very clear combined behavior between forest diversity and forest structural diversity in respect to the probabilities for finding the Great spotted woodpecker.

3.5.1 Question 7

- Is the presence of the Eurasian treecreeper affected by forest density?

3.5.1.1 Fitting a BRT in R

In this case we want to assess the occurrence of certain species across the plots. In other words, we want to assess what is the probability of a certain species with biodiversity interest to be present in a plot based on the variables that describe the forest of that plot. In this case the response variable is `phoenicurus` that represents if the Great spotted woodpecker has been observed in this plot or not.

Then you need to select some variables of interest, after you have explored the data you can decide which variables you want to use to fit this model. We are proposing to select the following variables:

- `latitude` as proxy for plot location or/and climate

- forestManagementType to assess if different management types have different impact in the presence / absence of Parus major.
- volAllha that is the total volume in the plot, as a proxy of how dense the plot is. Higher volumes will mean that the forest is more dense.
- GiniDBH showing how homogeneous the plot is in trees diameters. A value closer to 1 will mean that indicate more structural heterogeneity, lower values indicate more homogeneous plots.

You can create a vector selVar in which you add the names of the selected variables. Then you only take those variables from the data that you will use to create the model.

```
# Select variables from the dataset for the model
selVar <- c("certhia", "latitude", "forestManagementType",
            "volAllha", "GiniDBH",  "ShannonIndexTreeSpp",
            "Birds_rich")

# Filter the dataset to the selected variables
modelDataSel <- observations[, colnames(observations) %in% selVar]
```

Unfortunately the amount of that we have in this dataset it is not enough to fit a BRT model for these variables. We are going to do an obviously wrong thing for the shake of being able to demonstrate how to fit a BRT model. In the next code you are going to repeat the same dataset multiple times:

```
modelDataSel <- rbind(modelDataSel, modelDataSel, modelDataSel, modelDataSel,
                      modelDataSel)
```

Now it is important to assess if the variables have the right categories. Variables should be type numeric or factor.

```
summary(modelDataSel)
```

```
    latitude     forestManagementType    volAllha          GiniDBH
 Min.   :48.65   Length:495           Min.   :  1.681   Min.   :0.08209
 1st Qu.:49.31   Class :character     1st Qu.:319.920   1st Qu.:0.13684
 Median :49.40   Mode  :character     Median :434.729   Median :0.20358
 Mean   :49.49                        Mean   :425.694   Mean   :0.25683
 3rd Qu.:50.19                        3rd Qu.:558.355   3rd Qu.:0.37368
 Max.   :50.34                        Max.   :777.882   Max.   :0.52852
 ShannonIndexTreeSpp   Birds_rich        certhia
 Min.   :0.000       Min.   :0.1694   Min.   :0.0000
 1st Qu.:0.060       1st Qu.:0.3995   1st Qu.:0.0000  
 Median :0.280       Median :0.4679   Median :1.0000  
 Mean   :0.392       Mean   :0.4698   Mean   :0.5859  
 3rd Qu.:0.680       3rd Qu.:0.5089   3rd Qu.:1.0000  
 Max.   :1.450       Max.   :0.8360   Max.   :1.0000  
```

```
# Two variables are character, we assign to factor instead:
modelDataSel$forestManagementType <- as.factor(modelDataSel$forestManagementType)
```

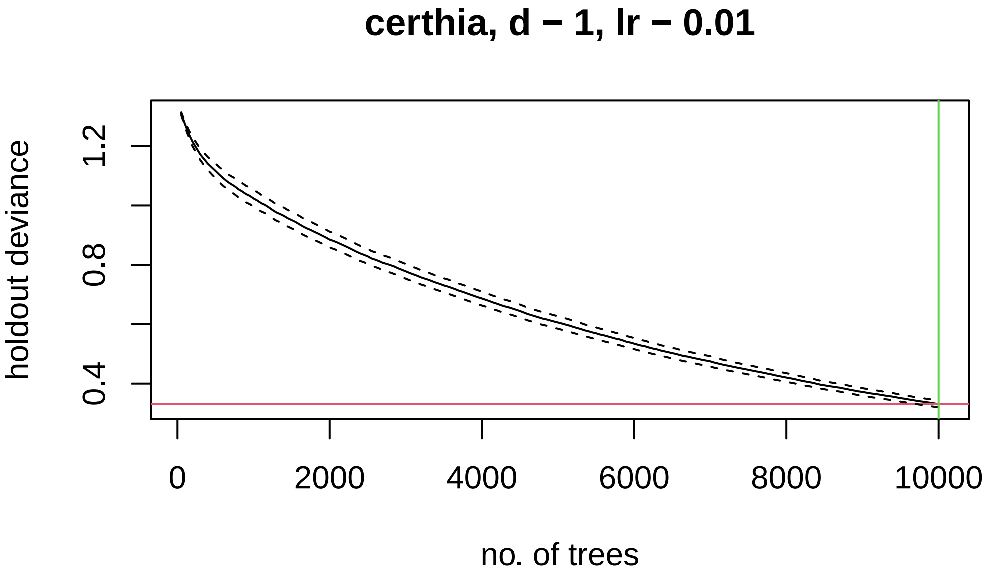
In the next step you can see how you can run the model with the selected variables and model parameters. You have a description of the models parameters in the Section 3.3.2 . In this example we are going to use the default parameters for the calibration, where learning rate = 0.01 and tree complexity = 1 and cross-validation = 10-fold. However, the bag fraction is changed from the default value, 0.75, to 0.5. As a family we used the Bernoulli family, because we are predicting presence/absence per plot. These data have 495 plots, comprising 290 presence records for the Eurasian treecreeper. You can check these numbers by doing:

```
table(modelDataSel$certhia)
```

```

  0   1 
205 290 
```

As a first guess you could decide there are enough data to model interactions of reasonable complexity, and a lr of about 0.01 could be a reasonable starting point. You can use the model creation function that steps forward and identifies the optimal number of trees (nt) by doing this:

```
family <- "bernoulli"

    tc = 1    # tree complexity
    lr = 0.01 # learning rate-shrinkage
    bag = 0.5 # bag fraction

    modelBRT <- dismo::gbm.step(data = modelDataSel,
                          #indices of predictor variables in data
                          gbm.x = 1:6,
                          #index of response variable in data:
                          gbm.y = 7,
                          family = family,
                          tree.complexity = tc,
                          learning.rate = lr,
                          bag.fraction = bag)
```



Running a model such as that described above writes progress reports to the screen, makes a graph, and returns an object containing a number of components. The R console results reports a brief model summary all the values are also retained in the model object.

The model is built with the default 10-fold cross-validation (CV). In the plotted graph the solid black curve is the mean, and the dotted curves 1 standard error, for the changes in predictive deviance (i.e., as measured on the excluded folds of the CV). The red line shows the minimum of the mean, and the green line the number of trees at which that occurs. The final model that is returned in the model object is built on the full data set, using the number of trees identified as optimal.

Ideally, you should invest time in modifying the parameters and finr the parameters that provide the models with the minimum deviance resulting from the best combination of bag, tree complexity and learning rate values. For the shake of limited timing, we will only test here the default values.

3.5.1.2 Model behaviour

You can summarized the model parameters used and the cross validation statistics from the fitted model by doing this:

```
# We make a table with the summary statistics
 results <- data.frame(

# Model parameters
Tree.Complexity = modelBRT$gbm.call$tree.complexity,
Learning.Rate = modelBRT$gbm.call$learning.rate,
                 Bag.Fraction = modelBRT$gbm.call$bag.fraction,
                 Interaction.depth = modelBRT$interaction.depth,
                 Shrinkage = modelBRT$shrinkage,
                 N.trees = modelBRT$n.trees,

# Cross validation statistics

## mean total deviance
Deviance = modelBRT$self.statistics$mean.resid, # mean residual deviance

AUC = modelBRT$self.statistics$discrimination, # training data AUC score

Corr = modelBRT$self.statistics$correlation,   # training data correlation

## Cross Validation statistics

# We calculate each statistic within each fold (at the identified optimal number
# of trees that is calculated on the mean change in predictive deviance over all folds),
#then present here the mean and standard error of those fold-based statistics.

devianceCV = modelBRT$cv.statistics$deviance.mean,  # estimated cv deviance
devianceCVse = modelBRT$cv.statistics$deviance.se,  # estimated cv deviance se

CorrCV = modelBRT$cv.statistics$correlation.mean,  #cv correlation
CorrCVse = modelBRT$cv.statistics$correlation.se,  #cv correlation se

AUCcv = modelBRT$cv.statistics$discrimination.mean, # cv AUC score
AUCcvSE = modelBRT$cv.statistics$discrimination.se)  # cv AUC score se


print(t(results))
```

```
                          [,1]
Tree.Complexity   1.000000e+00
Learning.Rate     1.000000e-02
Bag.Fraction      5.000000e-01
Interaction.depth 1.000000e+00
Shrinkage         1.000000e-02
N.trees           1.000000e+04
Deviance          2.742447e-01
AUC               1.000000e+00
Corr              9.728129e-01
devianceCV        3.308735e-01
devianceCVse      1.112925e-02
CorrCV            9.543224e-01
CorrCVse          5.398620e-03
AUCcv             9.993100e-01
AUCcvSE           6.900000e-04
```

3.5.1.3 Model output analysis

We can look at the relative contribution of each of the predictor variables. The measures are based on the number of time the variable is selected for splitting, weighted by the improvement of the model as a result of each split averaged across all trees. The relative contribution of each of the variables is scaled so the sum is 100%, with higher numbers indicating stronger influence in the response.

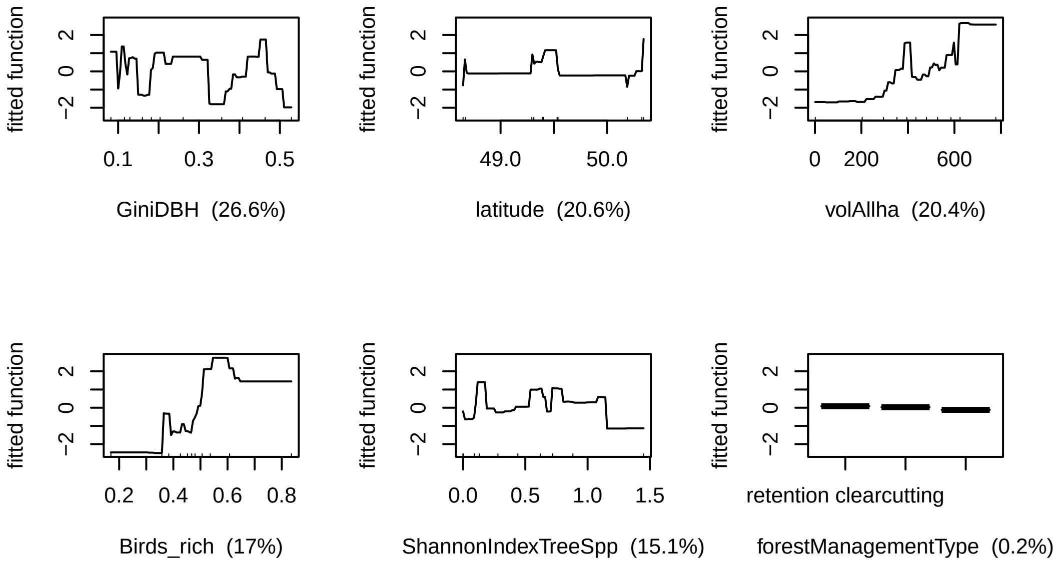
```
# Variables contribution
modelBRT$contributions
```

```
                                      var    rel.inf
GiniDBH                           GiniDBH 26.6331987
latitude                         latitude 20.5912754
volAllha                         volAllha 20.4227241
Birds_rich                     Birds_rich 16.9979177
ShannonIndexTreeSpp   ShannonIndexTreeSpp 15.1170258
forestManagementType forestManagementType  0.2378582
```

Here we can see that the two variables with the highest influence in the response are `GiniDBH` and `volAllhsa`.

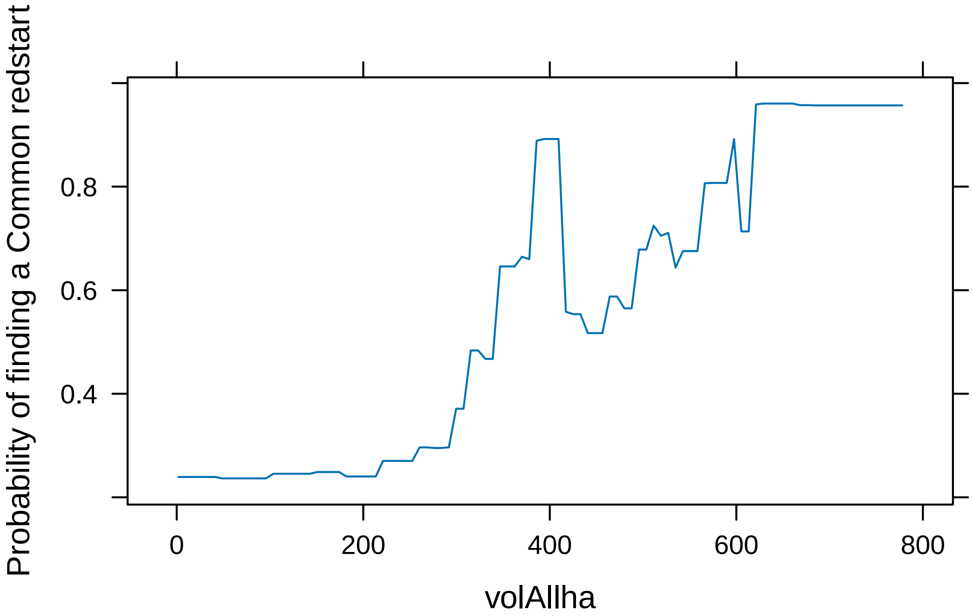
Now we can evaluate the model behavior via partial dependence plots, showing the effect of each of the variables on the response by accounting for the average effects of all other predictors in the model:

```
dismo::gbm.plot(modelBRT, n.plots = 6,
                plot.layout = c(2, 3), write.title = F)
```



In this partial dependence plots the predictions are on the scale of f(x). In this case, for the Bernoulli loss the returned value is on the log odds scale. You can see how this plot will look by plotting with the function from the package `gbm` and using the type "response". Since we are interesting in finding out if the forest density has an impact in the presence of the Eurasian treecreeper affected we can have a look to the plot density variable `volAllha` :

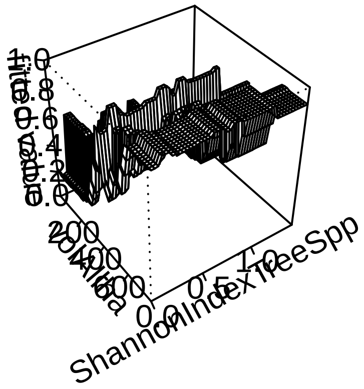
```
gbm::plot.gbm(modelBRT, i.var = 3, type = "response", ylab = "Probability of finding a Co
```



It seems that there is an increase in probability of finding a Eurasian treecreeper with higher forest densities. We could also analyse the interaction effects, of forest diversity and for example forest density. The model predictions can be obtained for each pair of predictor variables, setting all other predictors to their means.

To plot this pairwise interactions we have to do:

```
dismo::gbm.perspec(modelBRT, 3, 5)
```



It seems that the highes chances to see a Eurasian treecreeper is in dense forests with highest tree diversity.

3.5.2 Question 8

- Is the presence of the Eurasian treecreeper affected by forest management?

3.5.2.1 Fitting a BRT in R

In this case we want to assess the occurrence of certain species across the plots. In other words, we want to assess what is the probability of a certain species with biodiversity interest to be present in a plot based on the variables that describe the forest of that plot. In this case the response variable is `phoenicurus` that represents if the Great spotted woodpecker has been observed in this plot or not.

Then you need to select some variables of interest, after you have explored the data you can decide which variables you want to use to fit this model. We are proposing to select the following variables:

- `latitude` as proxy for plot location or/and climate
- `forestManagementType` to assess if different management types have different impact in the presence / absence of Parus major.

- `volAllha` that is the total volume in the plot, as a proxy of how dense the plot is. Higher volumes will mean that the forest is more dense.
- `GiniDBH` showing how homogeneous the plot is in trees diameters. A value closer to 1 will mean that indicate more structural heterogeneity, lower values indicate more homogeneous plots.

You can create a vector `selVar` in which you add the names of the selected variables. Then you only take those variables from the data that you will use to create the model.

```
# Select variables from the dataset for the model
selVar <- c("certhia", "latitude", "forestManagementType",
            "volAllha", "GiniDBH",  "ShannonIndexTreeSpp",
            "Birds_rich")

# Filter the dataset to the selected variables
modelDataSel <- observations[, colnames(observations) %in% selVar]
```

Unfortunately the amount of that we have in this dataset it is not enough to fit a BRT model for these variables. We are going to do an obviously wrong thing for the shake of being able to demonstrate how to fit a BRT model. In the next code you are going to repeat the same dataset multiple times:

```
modelDataSel <- rbind(modelDataSel, modelDataSel, modelDataSel, modelDataSel,
                      modelDataSel)
```

Now it is important to assess if the variables have the right categories. Variables should be type numeric or factor.

```
summary(modelDataSel)
```

```
    latitude     forestManagementType    volAllha          GiniDBH
 Min.   :48.65   Length:495           Min.   :  1.681   Min.   :0.08209
 1st Qu.:49.31   Class :character     1st Qu.:319.920   1st Qu.:0.13684
 Median :49.40   Mode  :character     Median :434.729   Median :0.20358
 Mean   :49.49                        Mean   :425.694   Mean   :0.25683
 3rd Qu.:50.19                        3rd Qu.:558.355   3rd Qu.:0.37368
 Max.   :50.34                        Max.   :777.882   Max.   :0.52852
 ShannonIndexTreeSpp   Birds_rich        certhia
 Min.   :0.000       Min.   :0.1694   Min.   :0.0000
 1st Qu.:0.060       1st Qu.:0.3995   1st Qu.:0.0000
 Median :0.280       Median :0.4679   Median :1.0000
 Mean   :0.392       Mean   :0.4698   Mean   :0.5859
 3rd Qu.:0.680       3rd Qu.:0.5089   3rd Qu.:1.0000
 Max.   :1.450       Max.   :0.8360   Max.   :1.0000
```

```
# Two variables are character, we assign to factor instead:
modelDataSel$forestManagementType <- as.factor(modelDataSel$forestManagementType)
```

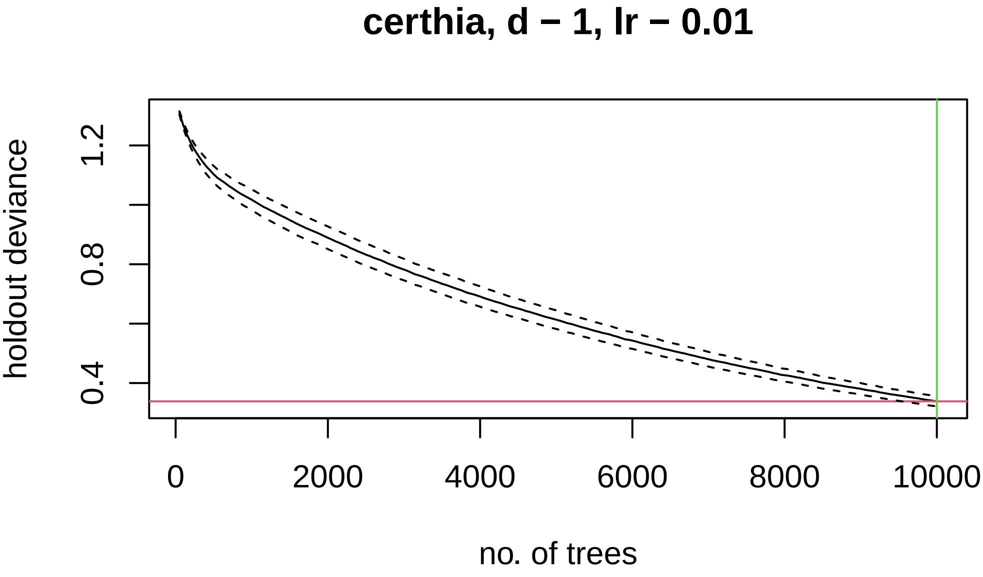
In the next step you can see how you can run the model with the selected variables and model parameters. You have a description of the models parameters in the Section 3.3.2 . In this example we are going to use the default parameters for the calibration, where learning rate = 0.01 and tree complexity = 1 and cross-validation = 10-fold. However, the bag fraction is changed from the default value, 0.75, to 0.5. As a family we used the Bernoulli family, because we are predicting presence/absence per plot. These data have 495 plots, comprising 290 presence records for the Eurasian treecreeper. You can check these numbers by doing:

```
table(modelDataSel$certhia)
```

```
  0   1 
205 290 
```

As a first guess you could decide there are enough data to model interactions of reasonable complexity, and a lr of about 0.01 could be a reasonable starting point. You can use the model creation function that steps forward and identifies the optimal number of trees (nt) by doing this:

```
family <- "bernoulli"

    tc = 1    # tree complexity
    lr = 0.01 # learning rate-shrinkage
    bag = 0.5 # bag fraction

    modelBRT <- dismo::gbm.step(data = modelDataSel,
                          #indices of predictor variables in data
                          gbm.x = 1:6,
                          #index of response variable in data:
                          gbm.y = 7,
                          family = family,
                          tree.complexity = tc,
                          learning.rate = lr,
                          bag.fraction = bag)
```



Running a model such as that described above writes progress reports to the screen, makes a graph, and returns an object containing a number of components. The R console results reports a brief model summary all the values are also retained in the model object.

The model is built with the default 10-fold cross-validation (CV). In the plotted graph the solid black curve is the mean, and the dotted curves 1 standard error, for the changes in predictive deviance (i.e., as measured on the excluded folds of the CV). The red line shows the minimum of the mean, and the green line the number of trees at which that occurs. The final model that is returned in the model object is built on the full data set, using the number of trees identified as optimal.

Ideally, you should invest time in modifying the parameters and finr the parameters that provide the models with the minimum deviance resulting from the best combination of bag, tree complexity and learning rate values. For the shake of limited timing, we will only test here the default values.

3.5.2.2 Model behaviour

You can summarized the model parameters used and the cross validation statistics from the fitted model by doing this:

```
# We make a table with the summary statistics
 results <- data.frame(

# Model parameters
Tree.Complexity = modelBRT$gbm.call$tree.complexity,
Learning.Rate = modelBRT$gbm.call$learning.rate,
                 Bag.Fraction = modelBRT$gbm.call$bag.fraction,
                 Interaction.depth = modelBRT$interaction.depth,
                 Shrinkage = modelBRT$shrinkage,
                 N.trees = modelBRT$n.trees,

# Cross validation statistics

## mean total deviance
Deviance = modelBRT$self.statistics$mean.resid, # mean residual deviance

AUC = modelBRT$self.statistics$discrimination, # training data AUC score

Corr = modelBRT$self.statistics$correlation,   # training data correlation

## Cross Validation statistics

# We calculate each statistic within each fold (at the identified optimal number
# of trees that is calculated on the mean change in predictive deviance over all folds),
#then present here the mean and standard error of those fold-based statistics.

devianceCV = modelBRT$cv.statistics$deviance.mean,  # estimated cv deviance
devianceCVse = modelBRT$cv.statistics$deviance.se,  # estimated cv deviance se

CorrCV = modelBRT$cv.statistics$correlation.mean,  #cv correlation
CorrCVse = modelBRT$cv.statistics$correlation.se,  #cv correlation se

AUCcv = modelBRT$cv.statistics$discrimination.mean, # cv AUC score
AUCcvSE = modelBRT$cv.statistics$discrimination.se)  # cv AUC score se


print(t(results))
```

```
                          [,1]
Tree.Complexity   1.000000e+00
Learning.Rate     1.000000e-02
Bag.Fraction      5.000000e-01
```

```
Interaction.depth 1.000000e+00
Shrinkage         1.000000e-02
N.trees           1.000000e+04
Deviance          2.741671e-01
AUC               1.000000e+00
Corr              9.724280e-01
devianceCV        3.385295e-01
devianceCVse      1.715425e-02
CorrCV            9.504394e-01
CorrCVse          6.592752e-03
AUCcv             9.995100e-01
AUCcvSE           4.900000e-04
```

3.5.2.3 Model output analysis

We can look at the relative contribution of each of the predictor variables. The measures are based on the number of time the variable is selected for splitting, weighted by the improvement of the model as a result of each split averaged across all trees. The relative contribution of each of the variables is scaled so the sum is 100%, with higher numbers indicating stronger influence in the response.

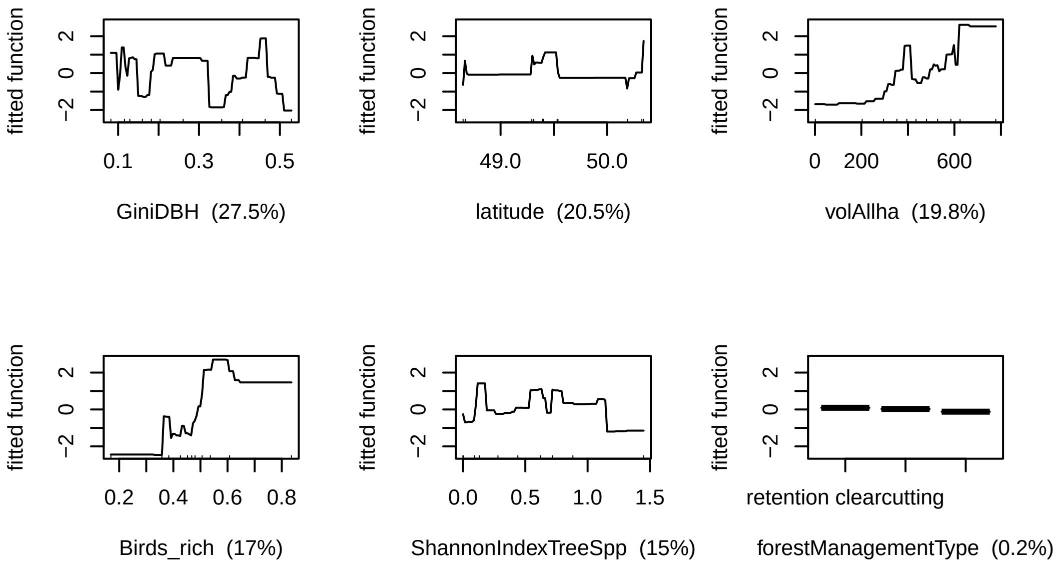
```
# Variables contribution
modelBRT$contributions
```

```
                                      var    rel.inf
GiniDBH                           GiniDBH 27.4611761
latitude                         latitude 20.5241487
volAllha                         volAllha 19.8156056
Birds_rich                     Birds_rich 17.0460334
ShannonIndexTreeSpp   ShannonIndexTreeSpp 14.9509203
forestManagementType forestManagementType  0.2021159
```

Here we can see that the two variables with the highest influence in the response are `GiniDBH` and `volAllhsa`.

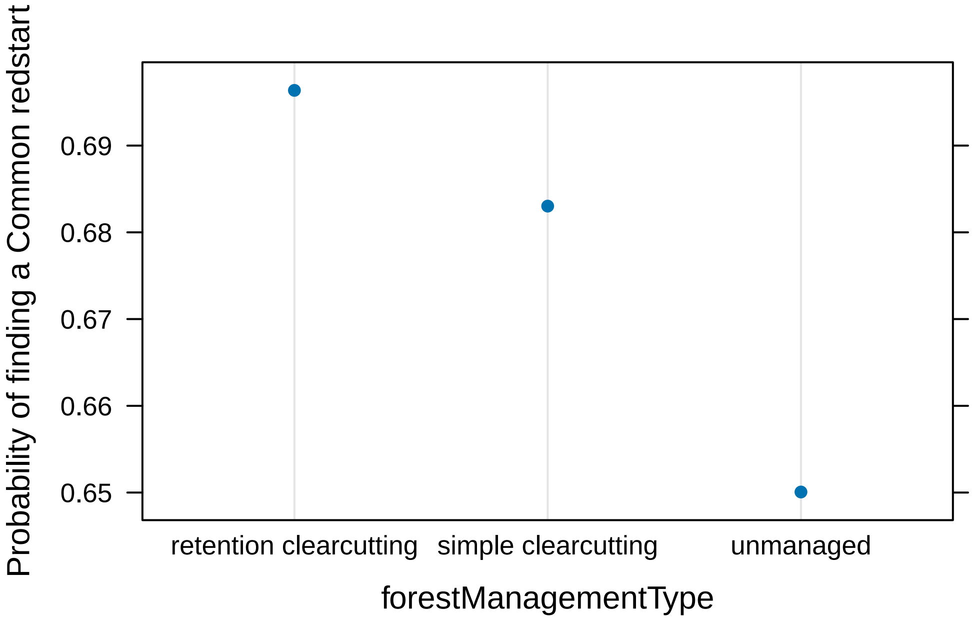
Now we can evaluate the model behavior via partial dependence plots, showing the effect of each of the variables on the response by accounting for the average effects of all other predictors in the model:

```
    dismo::gbm.plot(modelBRT, n.plots = 6,
             plot.layout = c(2, 3), write.title = F)
```



In this partial dependence plots the predictions are on the scale of f(x). In this case, for the Bernoulli loss the returned value is on the log odds scale. You can see how this plot will look by plotting with the function from the package `gbm` and using the type "response". Since we are interesting in finding out if the forest density has an impact in the presence of the Eurasian treecreeper affected we can have a look to the plot density variable `volAllha` :

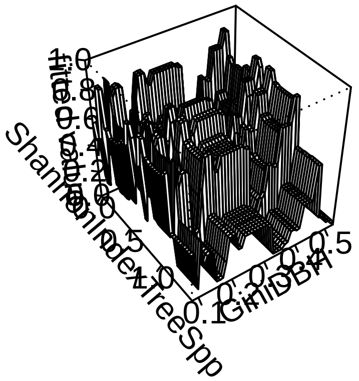
```
gbm::plot.gbm(modelBRT, i.var = 2, type = "response", ylab = "Probability of finding a Co
```



It seems that there is a small difference in the probability of finding the Eurasian treecreeper different management strategies. We could also analyse the interaction effects, of forest diversity and forest structual diversity. The model predictions can be obtained for each pair of predictor variables, setting all other predictors to their means.

To plot this pairwise interactions we have to do:

```r
dismo::gbm.perspec(modelBRT, 5, 4)
```



It seems that there are not clear pattenrs in the combined effect of forest structure diversity and forest tree species diversity.

References

Elith, J., J. R. Leathwick, and T. Hastie. 2008. "A Working Guide to Boosted Regression Trees." *Journal of Animal Ecology* 77 (4): 802–13. https://doi.org/10.1111/j.1365-2656.2008.01390.x.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. n.d. "ADDITIVE LOGISTIC REGRESSION: A STATISTICAL VIEW OF BOOSTING."

Hošek, Jan. n.d. "Forest Structure and Dead Wood Properties in Six Representative Forest Areas in the Czech Republic."