Regression approaches

Regression relates a response variable (e.g. presence–absence, abundance, biomass) to a set of pre-selected environmental predictors (e.g. climate, land use, resource).

The predictors can be used as untransformed environmental variables or, in order to prevent multicollinearity (phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy) in the data, as orthogonal components derived from the environmental variables through multivariate analyses (e.g., PCA).

Generalized Linear Models (GLMs) and Generalized Additive Models (GAMs) are commonly used regression approaches. The major difference between GLMs, and their extensions (e.g. GAMs) lies in the choice of model-driven versus data-driven response shapes. To properly use a GLM, one should have some expectation regarding the shape of the response variable along the predictors. When a highly limiting factor is expected, a linear relationship could be sufficient. Data-driven approaches, such as GAM, are slightly more flexible in this regard.

Generalized Linear Models

GLMs can handle Gaussian (e.g. biomass), Poisson (species abundance, species richness), binomial (e.g. presence–absence), or gamma distributions.

In our case, a binomial distribution is the obvious choice.

If the response shape is not a linear function of predictors, a transformed (higher-order polynomial) term of the latter can be included in the model. This type of regression is called a polynomial regression.

Simple linear regression:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Polinomial regression, second order:

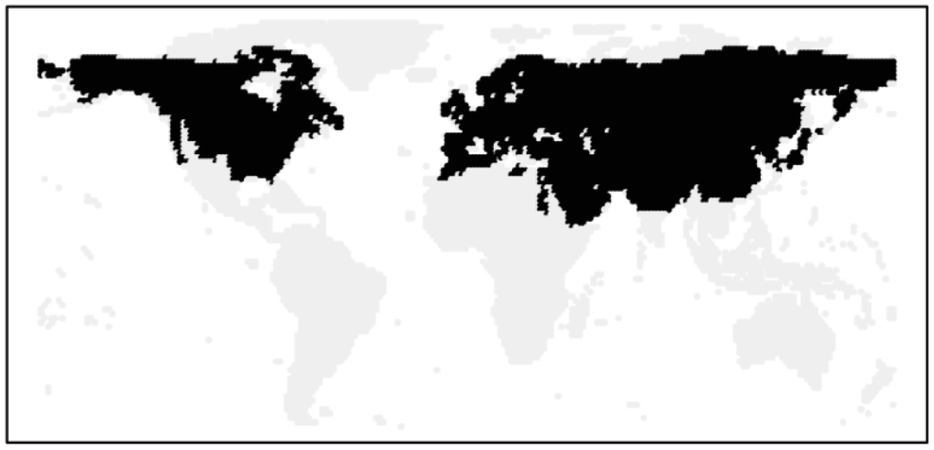
$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

Second order polynomial regressions simulate unimodal symmetric responses (e.g. a hypothetical bell-shaped relationship between species abundance and a given environmental variable), whereas third-order or higher terms make it possible to simulate skewed and bimodal responses, or even a combination of both.

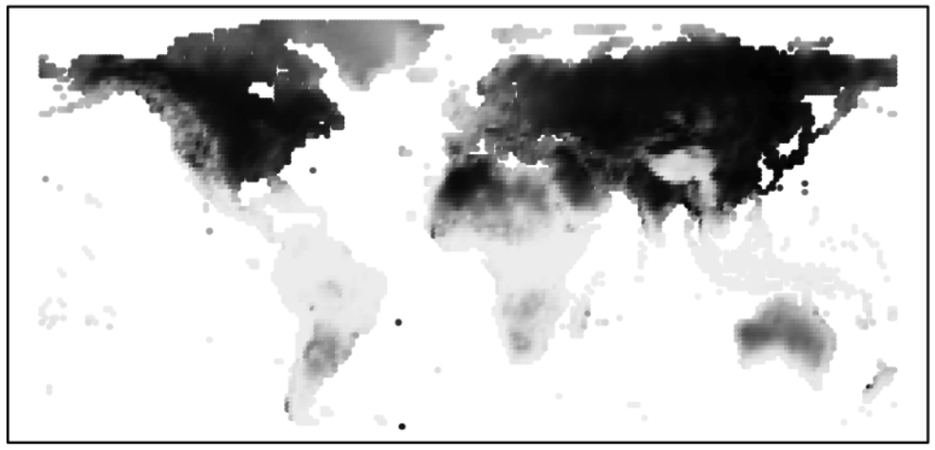
Let’s switch to R,
and make an example

Modelling approaches

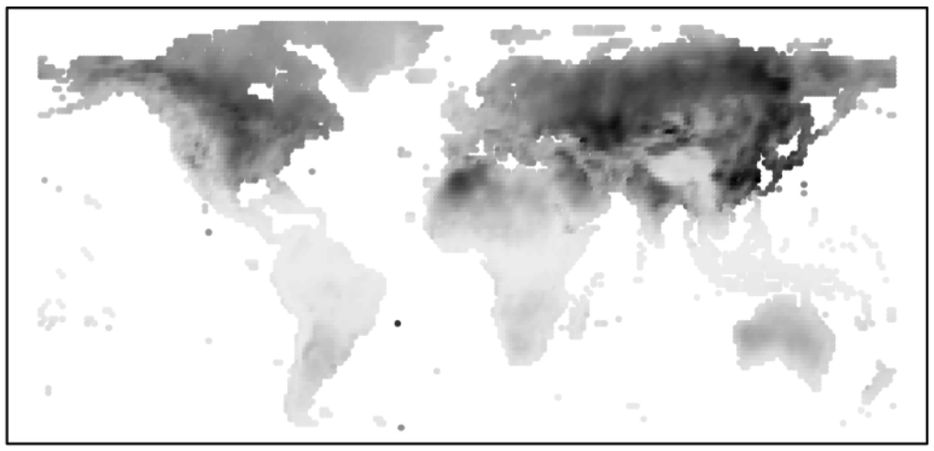
Original data



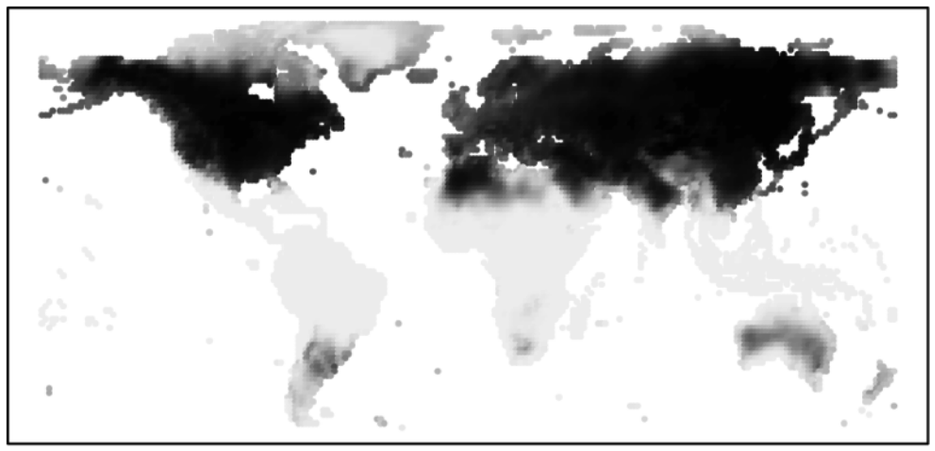
GLM with linear terms, binomial distribution



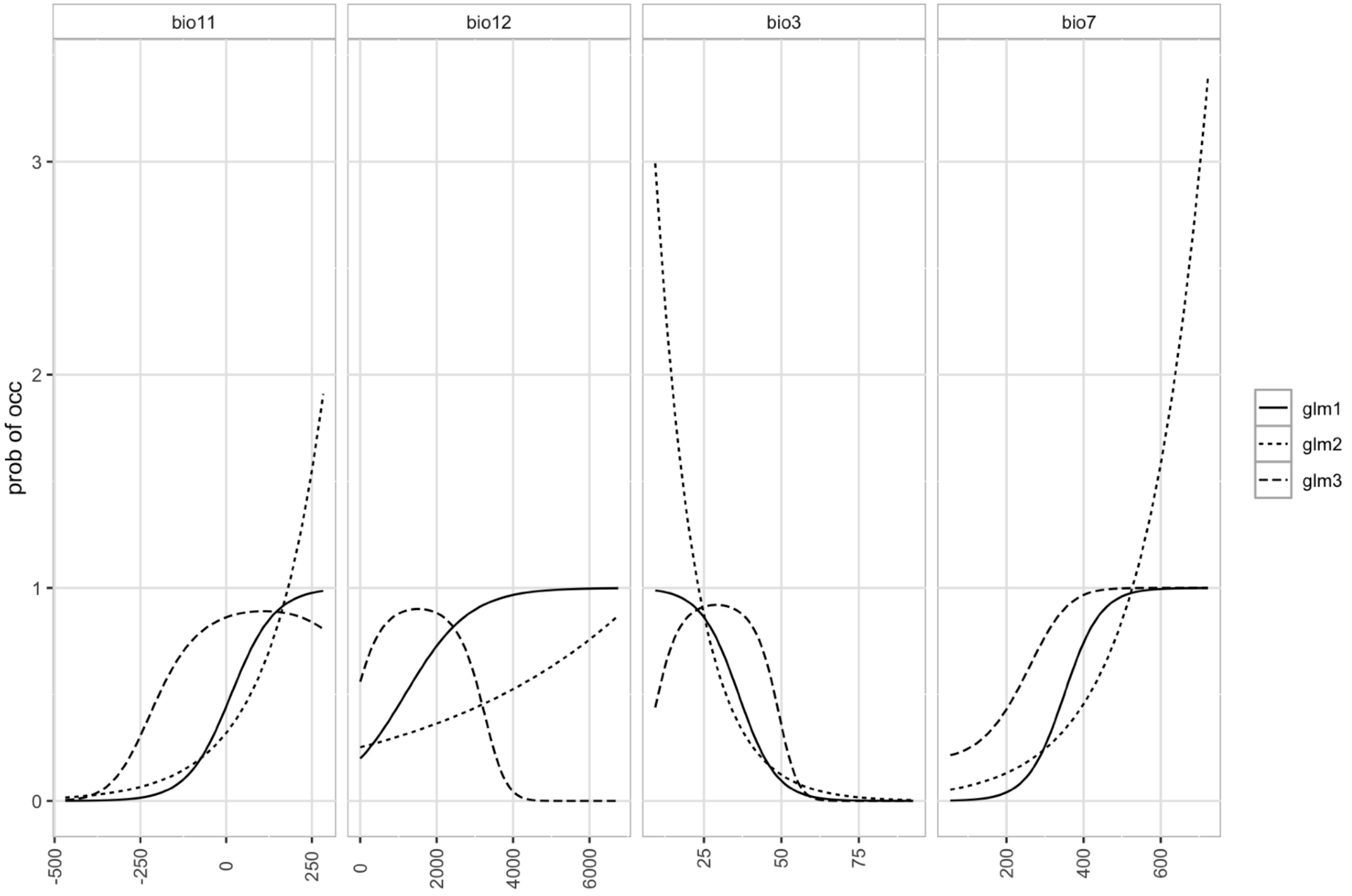
GLM with linear terms, Poisson distribution



GLM with quadratic terms, binomial distribution



Modelling approaches



The model which uses the Poisson distribution is clearly biased, since we are not using abundance data, but presence/absence data.

The two models *glm1* and *glm3* differ in terms of the hypotheses used regarding the shape of the relationship between all variables and the presence of the species. In *glm1*, we assume that linear predictors are sufficient, in *glm2 we* expect quadratic relationships, (i.e. non-symmetric, unimodal or sigmoidal relationships).

While the spatial distributions of the probability of occurrence from *glm1* and *glm3* appear rather similar, the modeled responses differ in environmental space, as shown in the response curves of the species along the environmental gradients fitted in the models.

For building the predicted response curves, n-1 variables are set as constants to a fixed value (mean, median, min or max, i.e. fixed.var.metric argument) and only the remaining one (remaining two for three-dimensional response plots) varies across its whole range (given by data), showing the sensibility of the model to that specific variable.

Generalized Additive Models

GAMs are techniques designed to capitalize on the strengths of GLMs but which do not require postulating a shape for the response curve from a specific parametric function. GAMs use algorithms called “smoothers” that automatically fit response curves “as closely as possible” to the data given the permitted level of smoothing. GAMs are therefore useful when the relationship between the variables is expected to be of a more complex form, not easily fitted with standard parametric functions of the predictors (e.g. GLM with a linear or quadratic response), or where there is no a priori reason for using a particular shape.

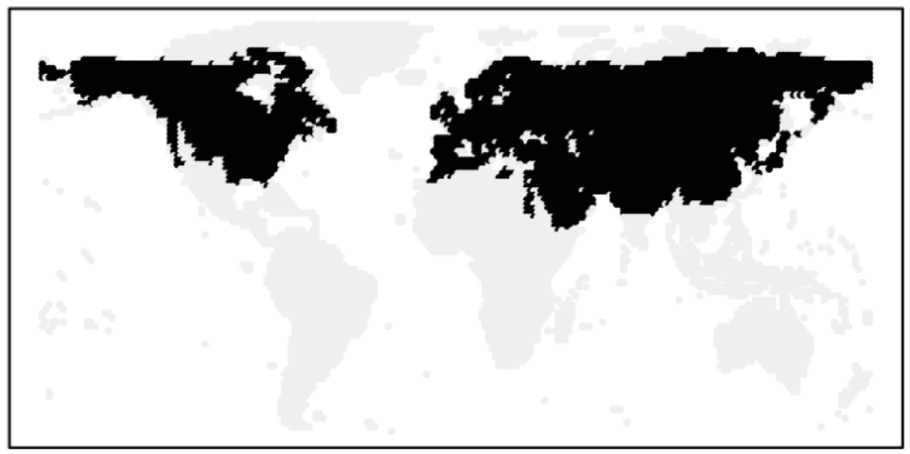
There are now several packages, which can be used to fit GAMs in R (e.g. *gam*, *mgcv*, *gamair*, *GAMBoost*). There are different smoothers available, but the most commonly used is the **cubic-spline** smoother, a collection of polynomials of degree less than or equal to 3. A separate polynomial model is fitted in each neighborhood (using a moving window algorithm), thus enabling the fitted curve to connect all the points.

Nevertheless, the user has to predetermine the degree of smoothing applied when fitting the curve (or select it through cross-validation). In the SDMs field, researchers have generally used degrees lower than 4, which corresponds roughly to a polynomial of degree 3. The degree of smoothing can change across the variables in a model.

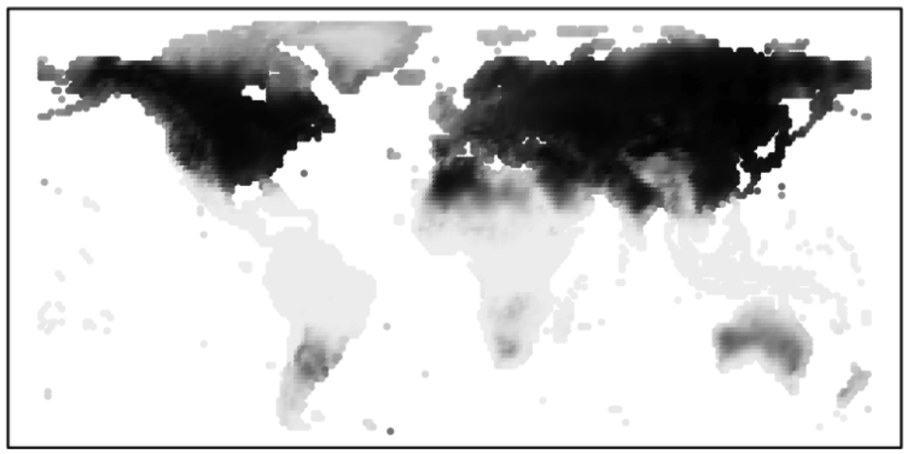
Let’s switch to R,
and make an example

Modelling approaches

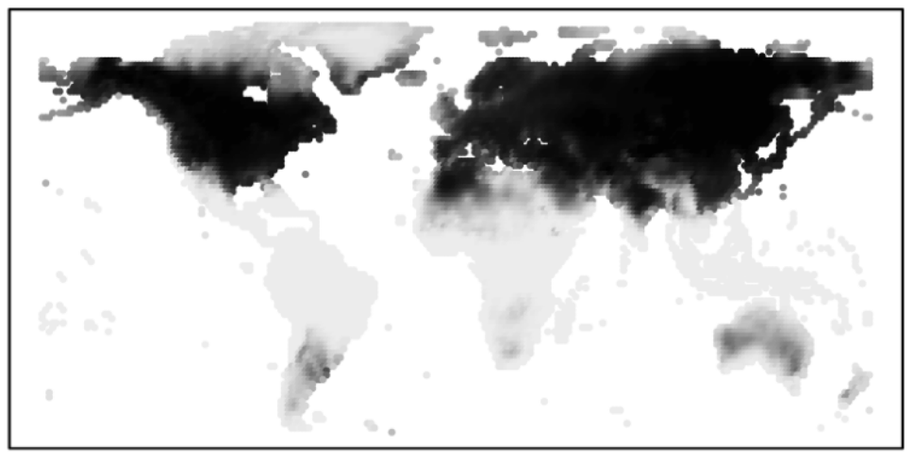
Original data



GAM with smooth level 2



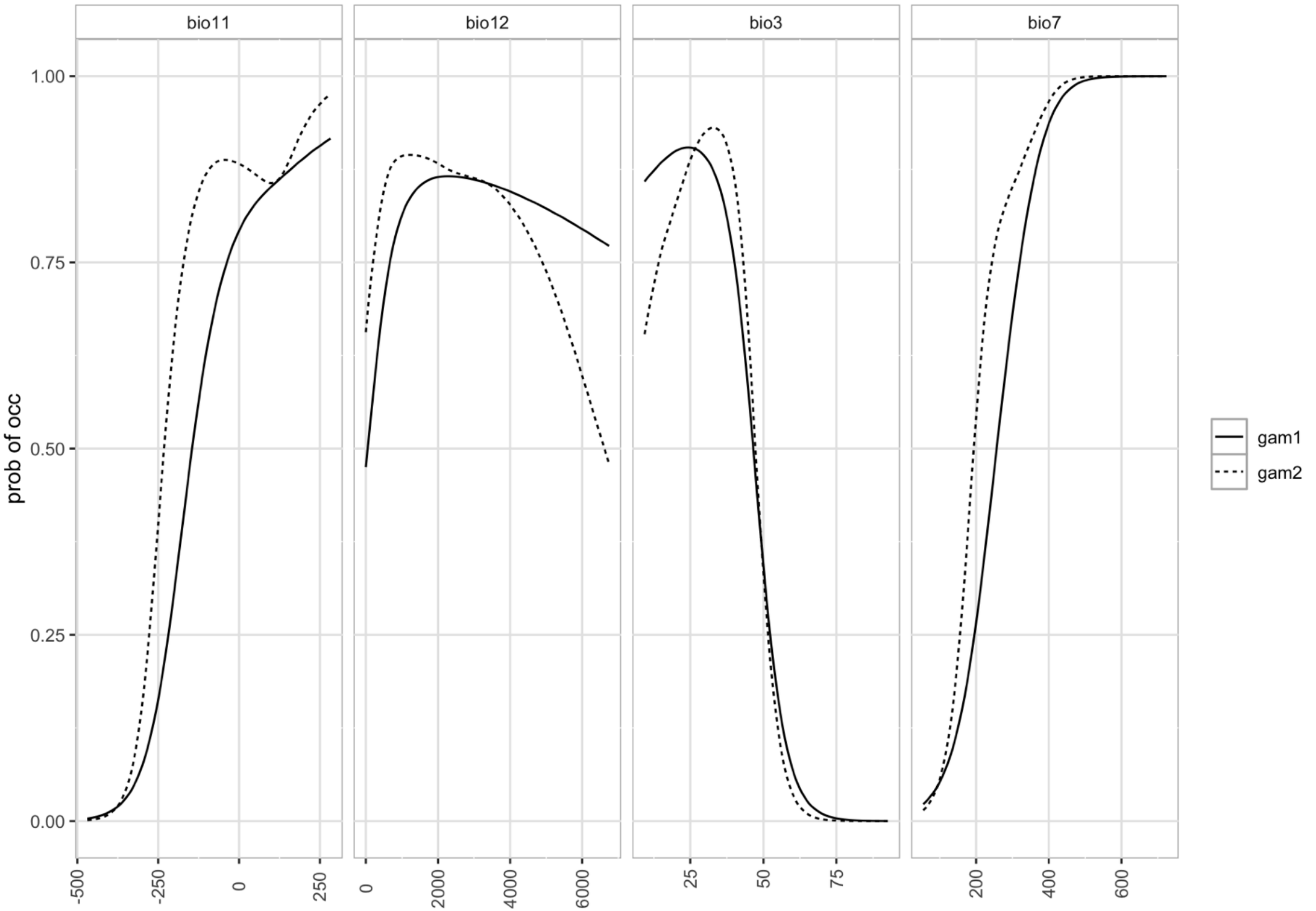
GAM with smooth level 4



Projections, and response curves are similar to GLMs. Degree of smoothing has a relatively small effect in this case.

GAMs are data-driven and thus prone to overfitting when highly complex smoothers are used. Thus, when modeling species distributions for predictive purposes, degrees of smoothing higher than 4 should not be adopted.

Modelling approaches



Recursive Partitioning

Among the different techniques generally categorized as classification approaches, recursive partitioning is one of the most interesting for habitat suitability modeling.

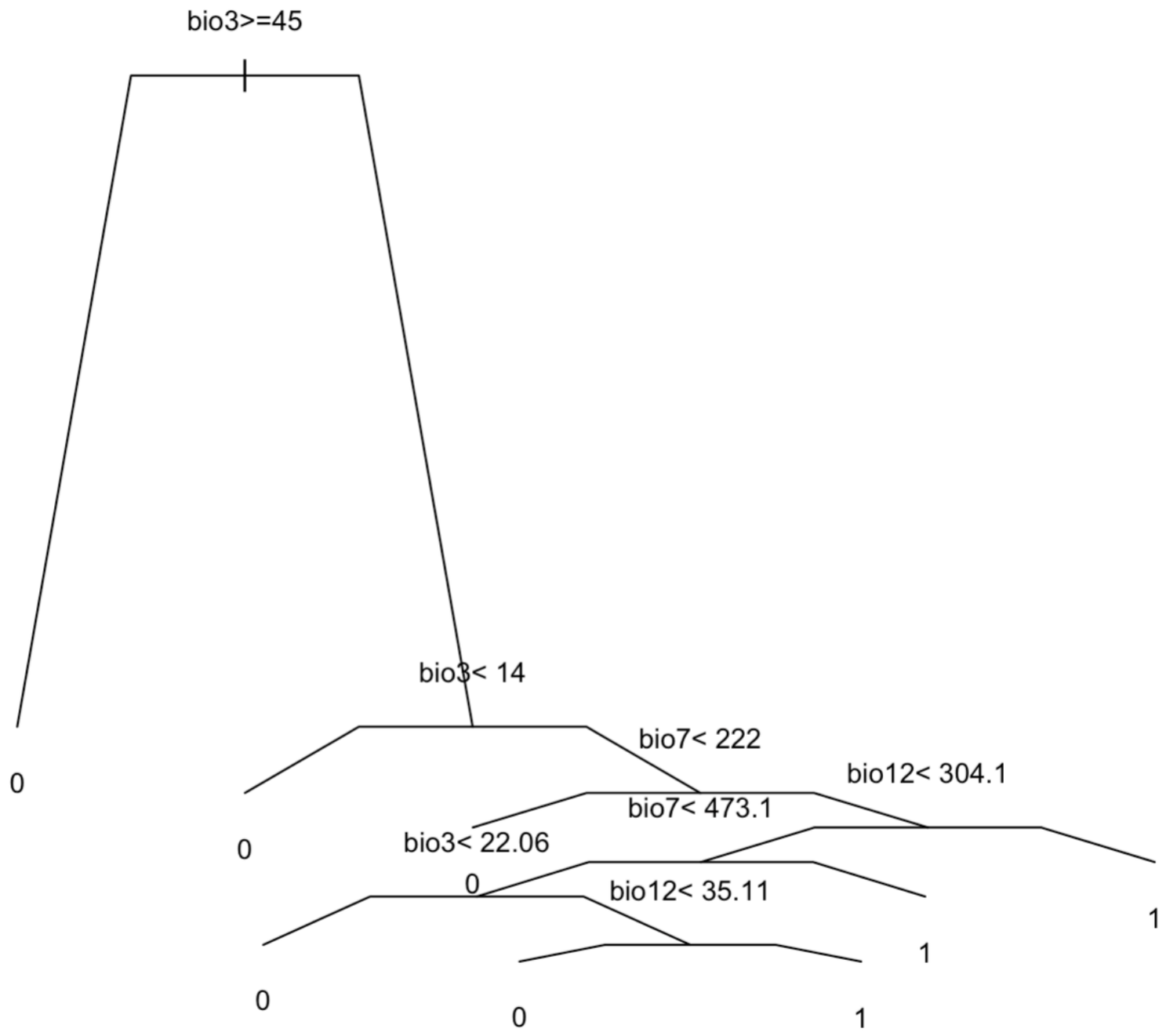
Recursive partitioning techniques also form the basis of more complex and powerful techniques such as bagging or boosting (e.g., RandomForests).

Recursive partitioning (RP) approaches are meant to explain the variation for a single response variable (e.g. species presence–absence, biomass, abundance) with one or more explanatory variables. The response variable can be either discrete (**classification trees**) or continuous (regression trees). Specifying a binary response (e.g. presence–absence) as a factor will lead to a classification tree, which is grown by repeatedly splitting the data, defined at each split (node) by a rule based on a single explanatory variable. At each split the data is partitioned into two mutually exclusive groups. The criteria for segmenting the data are based on either minimizing the classification error rate in the case of a classification tree, or maximizing the inter-class variance in the case of a regression tree.

The key trade-off is to partition the response into homogeneous groups, but also to keep the tree reasonably small in order to avoid overfitting the data through a very complex model. Furthermore, a complete tree will predict each data point perfectly, but will have limited power to predict outside of the training data.

Let’s switch to R,
and make an example

Modelling approaches



Data-splitting is first performed until an overly large tree is grown (the maximum possible size equals the number of samples, or sites). This complex tree is then pruned back to the desired size using specific rules to reduce overfitting. This pruning of the tree is the trickiest part of RP. The goal is to reduce the tree to an optimal size while maintaining enough predictive power to ensure accurate predictions. There are several algorithms for defining rules for pruning. The most common rely on cross-validation, where data-splitting is performed on a subset of data and then the predictive power is evaluated on the remaining data.

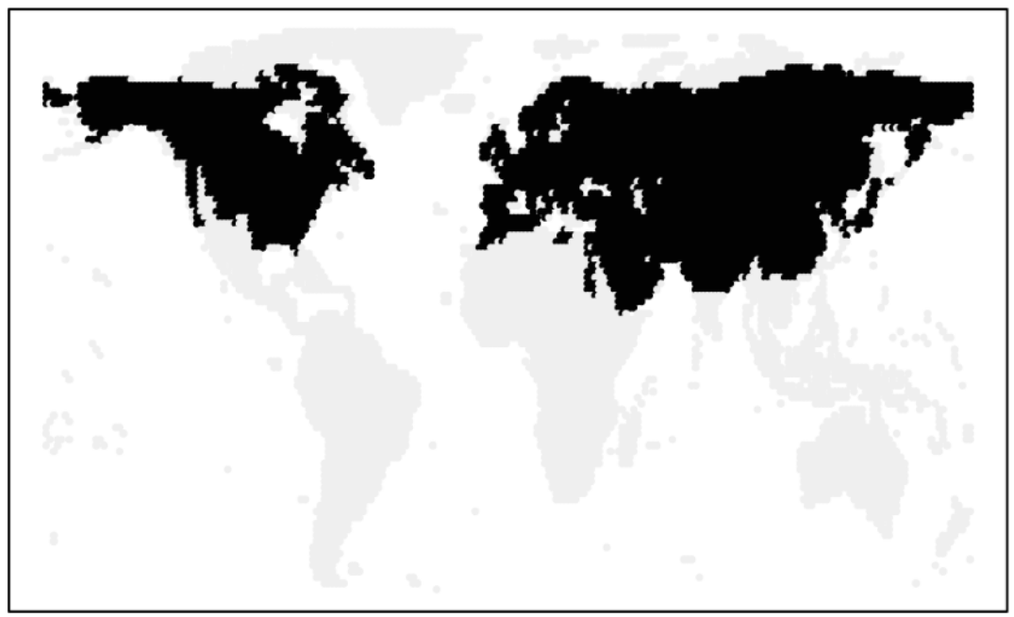
Each final leaf (or terminal node) corresponds to one, or a group of observations, and is predicted by the values of the explanatory variables that define the nodes along the path to the terminal leaf.

Obviously, the way the splits are defined depends on the type of the predictor variables. For continuous variables, a split is defined using values of less than, or greater than, a chosen splitting value.

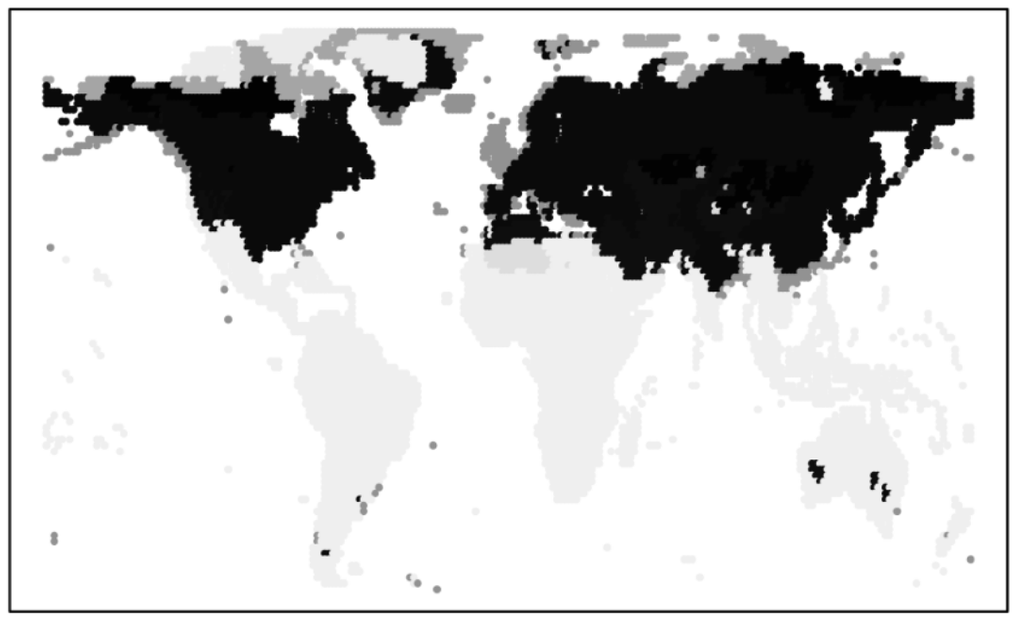
One advantage of RP is that it does not rely on assumptions about the relationship between the explanatory variable and the response variable of interest. Also, it does not expect the dependent variable to follow any specific distribution (as in GLM, or GAM models). The approach is thus entirely data-driven.

Let’s switch to R,
and continue this example

Original data



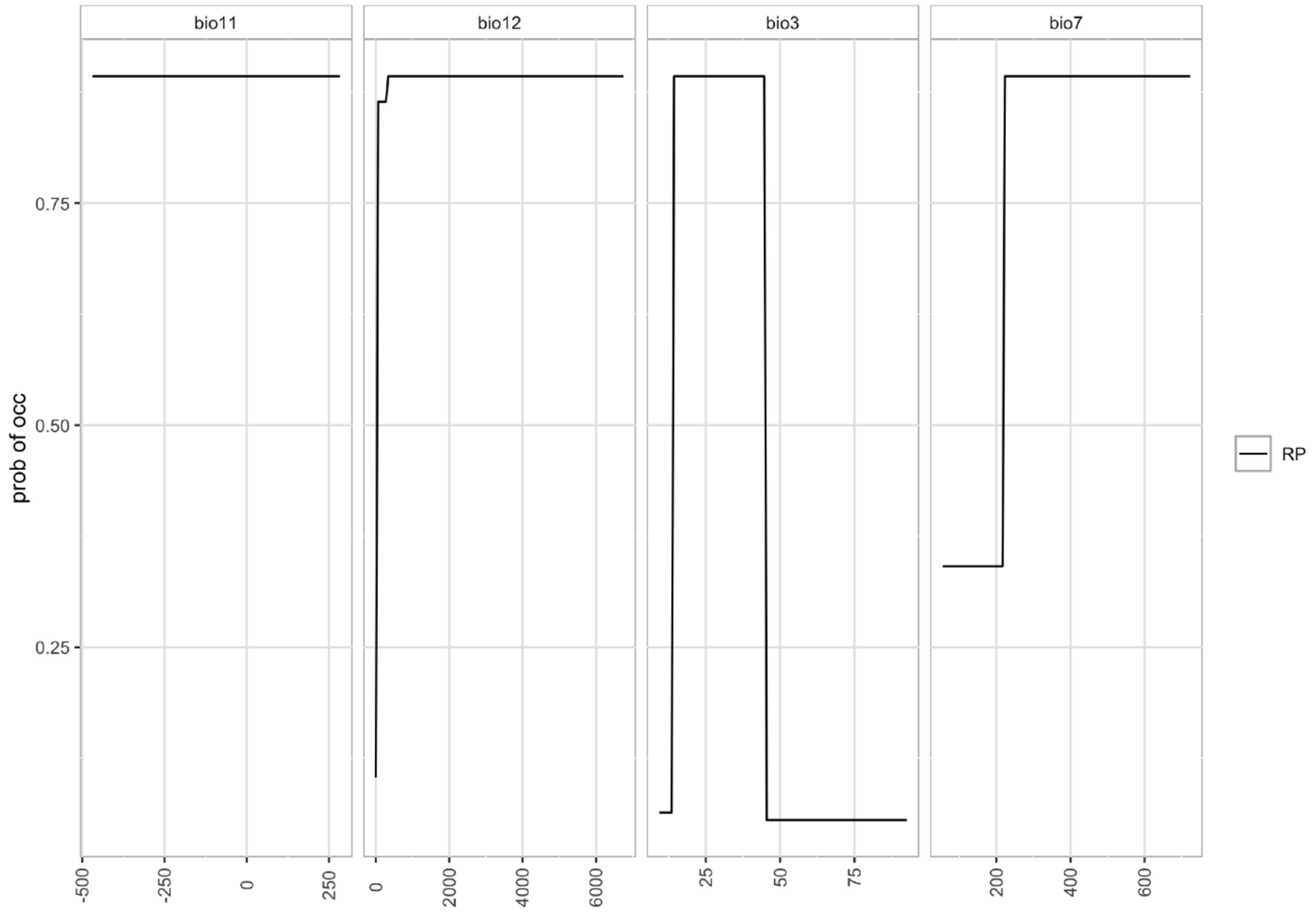
Recursive partitioning



Similarly to other techniques (e.g. GAM, GLM), the spatial prediction of an RP model can be easily obtained using the *predict*() function.

Note that *rpart* provides both the presence–absence values, and the probabilities of presence. Using the prob arguments makes it possible to extract the probabilities of presence.

Modelling approaches

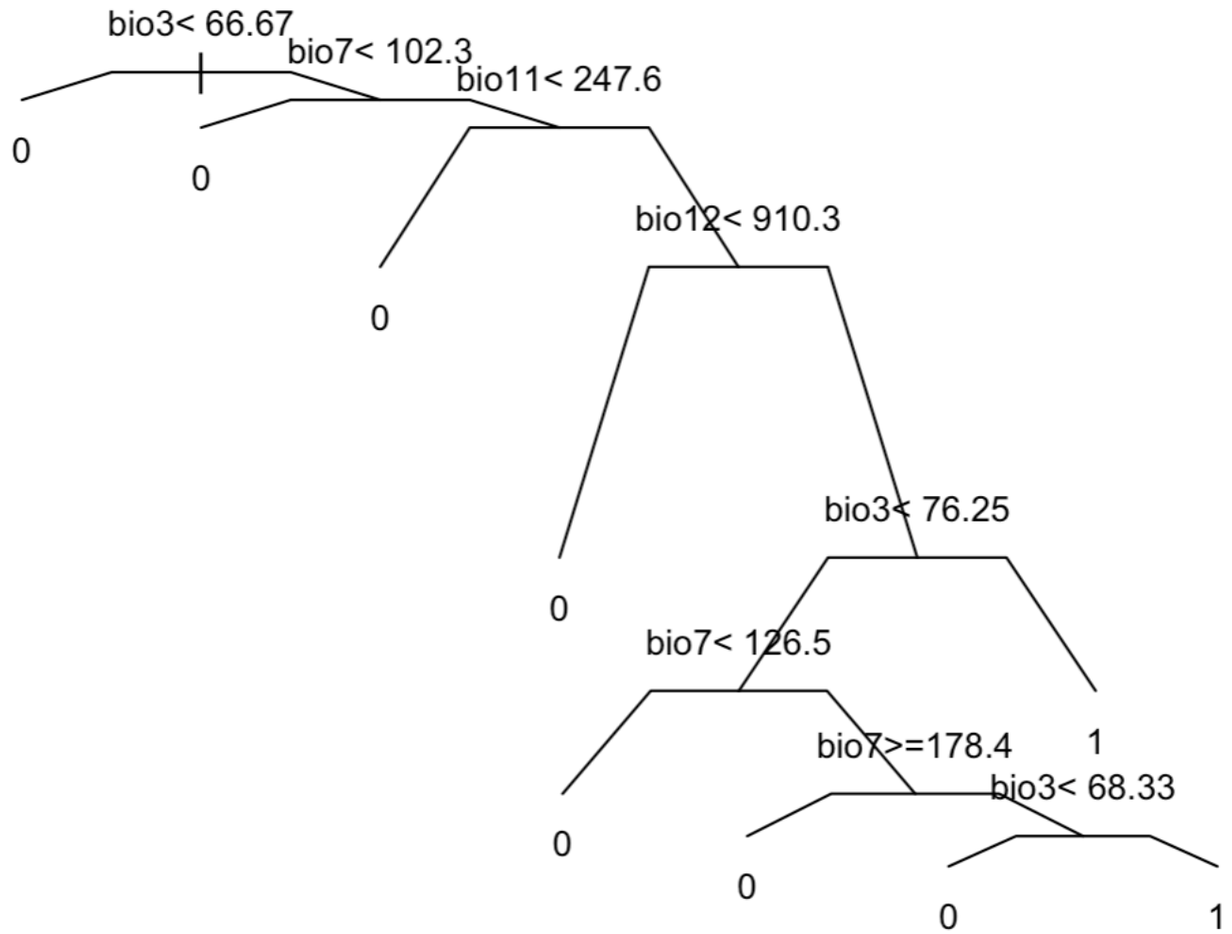


Response curves can also be extracted for RP in the same way as for GLM or GAM

Random Forests

One of the trickier aspects of RP is that it is a high-variance process. Small changes in the chosen variables or small changes in the dataset could lead to very different selected trees. The optimal tree size is also difficult to select.

As an example, let's select another species from our dataset. We will use the jaguar (*Panthera onca*). The fitted tree is slightly more complicated than for *V. vulpes.*



Ten splits have been selected in the optimized model. How does this value change with different cross-validations runs, for instance? How robust is it to noisy data or small perturbations in the input data? These are fundamental questions one should preferably ask when applying RP approaches, instead of taking the first decision tree as given.

The idea of **bagging** is to fit several trees to different resampling of the original dataset and then to average the trees from the different subsamples.

In order to understand the benefit of this approach, it is interesting to look at the structure of the multiple trees. A simple use of the *table*() function allows us to see which variable has been selected for a set of nodes.

We can see that through the 50 bootstraps, *bio3* is always for the first split. When going down the trees, it becomes clear that all the variables could have been selected for a given split. The further we go down the tree, the higher the variability of the selected variables.

Another advantage of the bootstrap approach is that it is possible to extract the averaged probability (and the variance) of occurrences across all bootstrap samples.

Random forests have been developed to check for overfitting by adding some stochasticity to the process of building the trees, but also at each node of each tree. Let's assume that we have *N* plots or sites and *X* explanatory variables, each tree is grown based on the follow procedure:

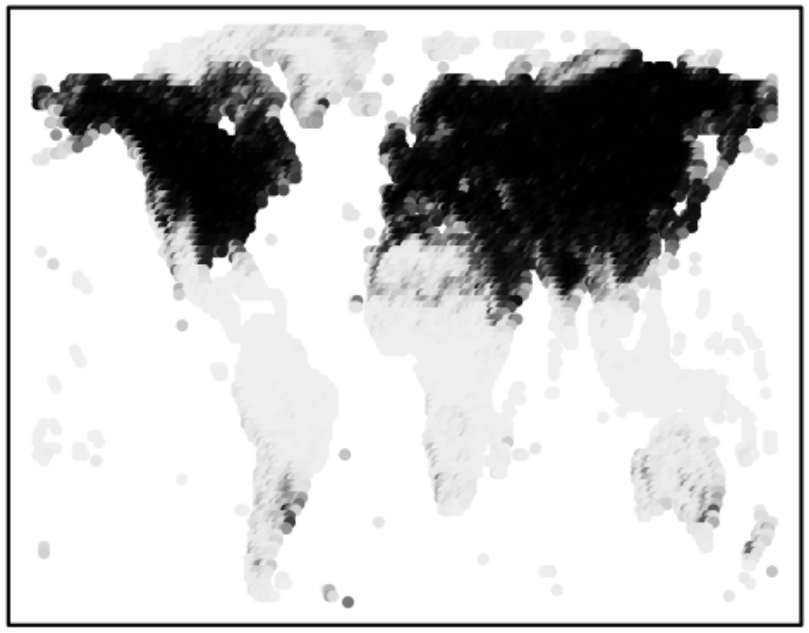
1. Take a bootstrapped sample of *N* sites at random with replacement. This sample represents the training set for growing the tree.
2. At each node, select *x* candidate variables randomly out of all *X* predictors and evaluate the best split based on one of these *x* variable for the node. The value of *x* has to be selected beforehand and is kept constant during the forest growing.
3. Each tree is grown to the largest possible extent. There is no pruning.

The number of candidate variables taken randomly at each node is one of the few adjustable parameters to which random forests are somewhat sensitive.

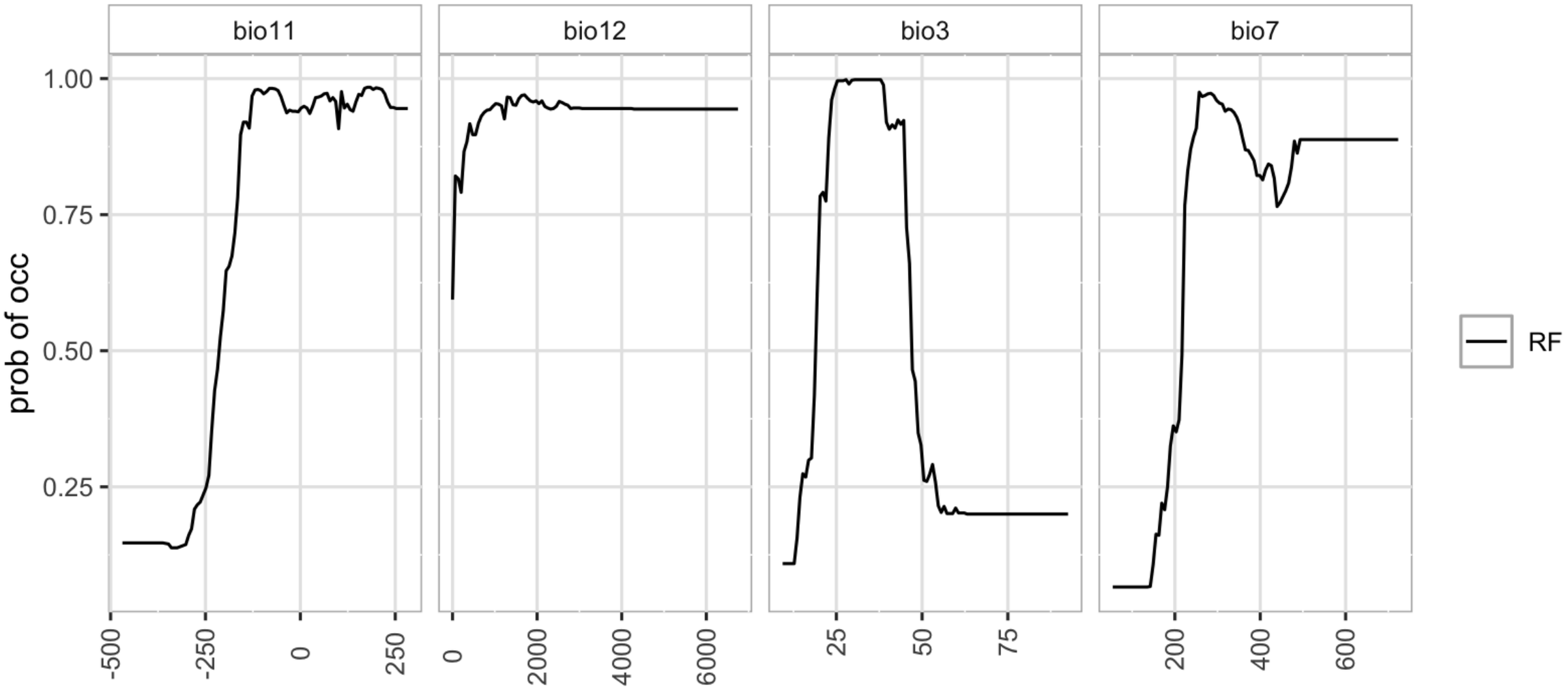
Let’s switch to R,
and see an example

Original data

RF



Modelling approaches



Maximum entropy (MaxEnt)

The application of the maximum entropy formalism to species distribution modeling was first introduced by Phillips et al. (2004), and is now well-developed in the standalone package Maxent.

Although it is not formally implemented in R, it is possible to run it from R, so that the Maxent results can be compared with those from other modeling techniques and approaches. Both *dismo* and *biomod2* can be used to run Maxent in a batch mode.

In addition, a maximum entropy R package is currently in development.

Conceptually, Maxent contrasts observed presence data (y = 1) to the available environment in a given region (named z, a vector of environmental predictors).

f(z) is the probability density of predictors across the region and f1(z) is the probability density of covariates across locations within the region where the species occurs.

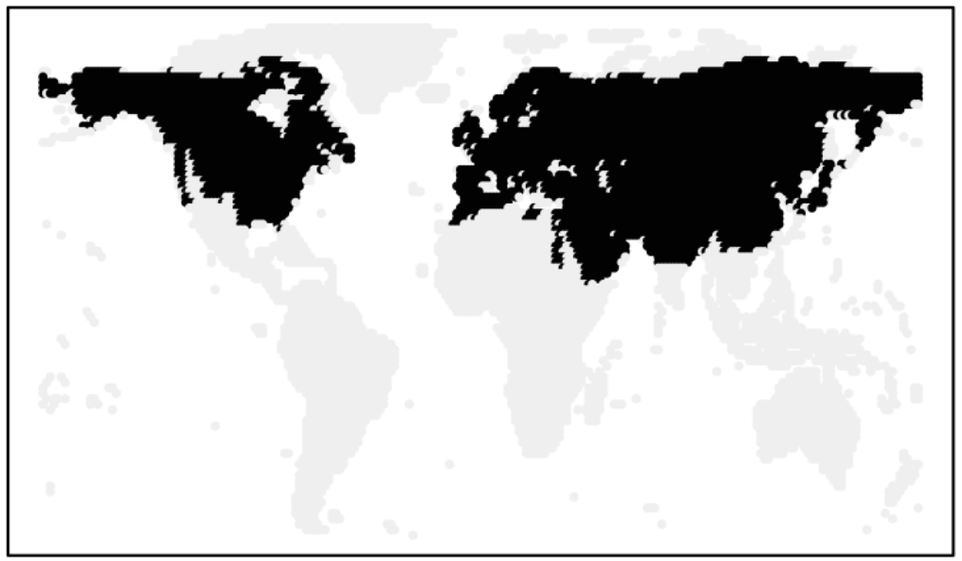
MaxEnt uses the predictors from the occurrence and the background sample to estimate the ratio f1(z)/f(z). The optimization algorithm looks for f1(z) that minimizes the distance from f(z).

f(z) is here seen as a null model for f1(z) since there is no reason to expect the species to prefer any particular environmental conditions in the absence of occurrence data. In the latter case, the best prediction is that the species occupies environmental conditions proportionally to their availability in the region. In MaxEnt, this distance from f(z) is taken to be the relative entropy of f1(z) with respect to f(z).

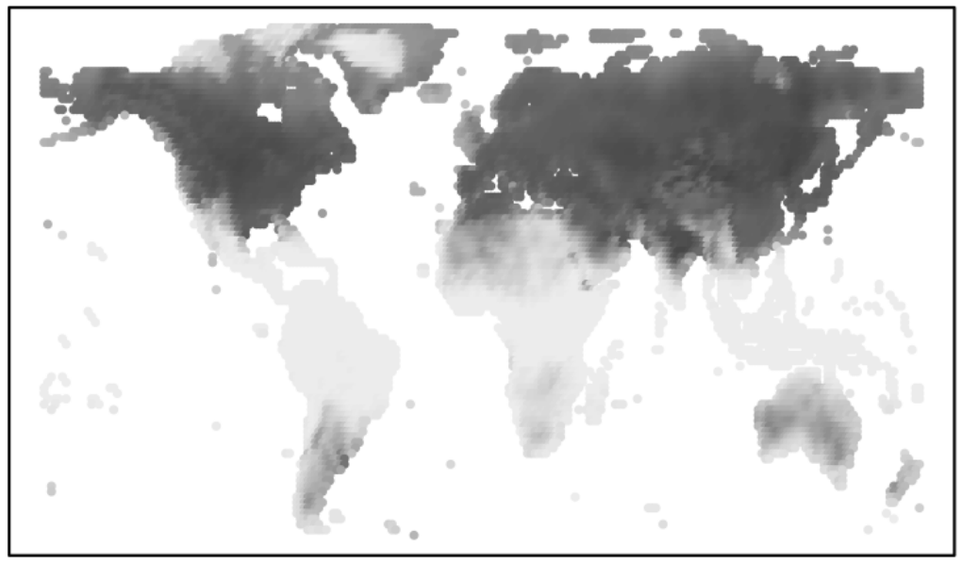
Maxent does not specifically model presence data but rather the density of used environmental conditions.

Let’s switch to R,
and see an example

Original data

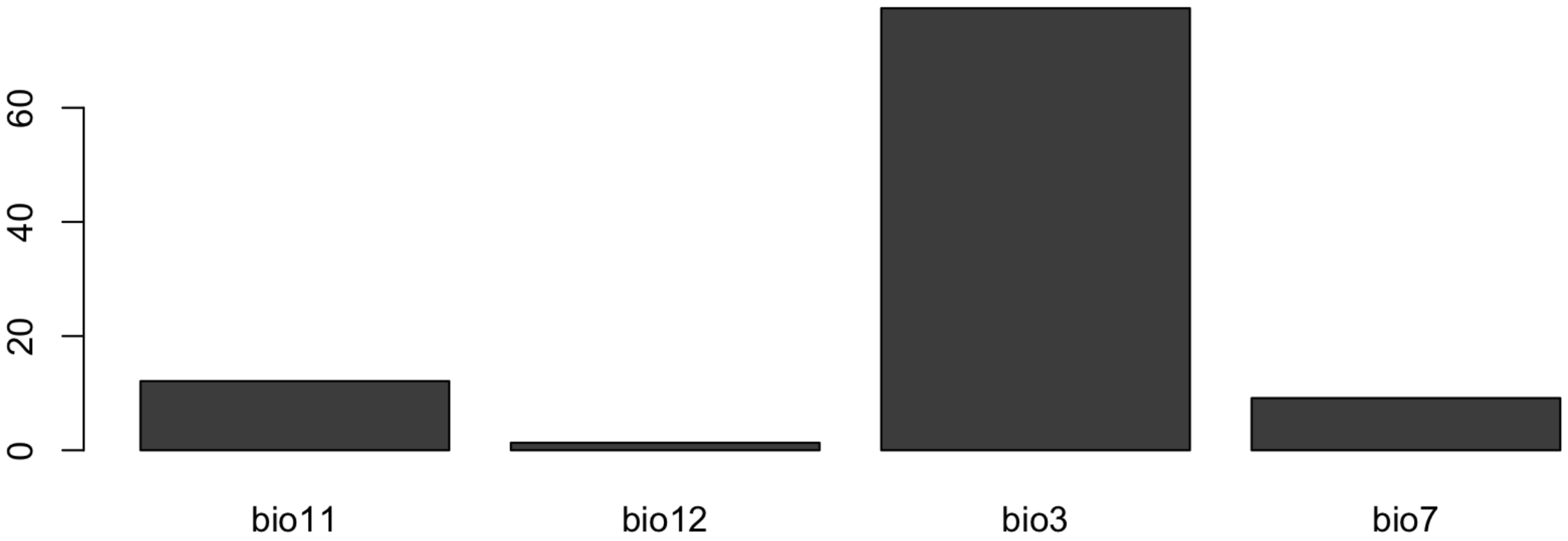


MAXENT

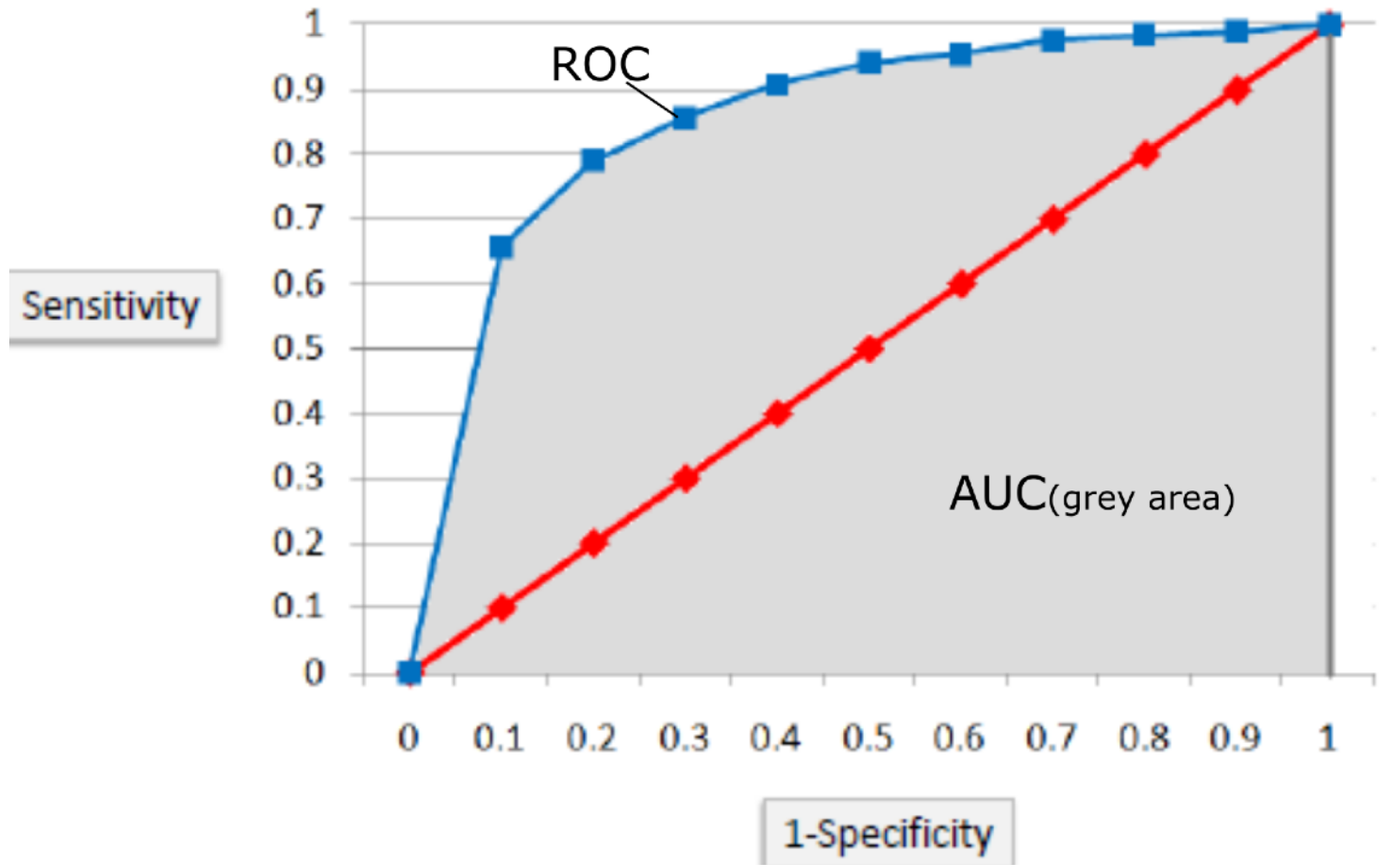


Observed (black = presence, light gray = absence) and (b) potential distribution of the red fox modeled using Maxent in batch mode from R. The gray scale of predictions shows habitat suitability values between 0 (unsuitable) and 1 (highly suitable).

Modelling approaches



Heuristic estimate of relative contributions of the four environmental variables to the Maxent model using a jackknife procedure.
The jackknife procedure estimates a parameter by systematically leaving out each observation from a dataset, calculating the estimate, and finally finding the average of the calculations.



AUC is the **A**rea **U**nder the ROC **C**urve (grey in the graph), calculated simply with integal of the function (ROC).
Measuring this you can discriminate performance of the model:

AUC = 0,5 (line of non discrimination, red in the graph) model no better then random
AUC > 0,8 good model

Model evaluation: contingency table

Real (occurrence)

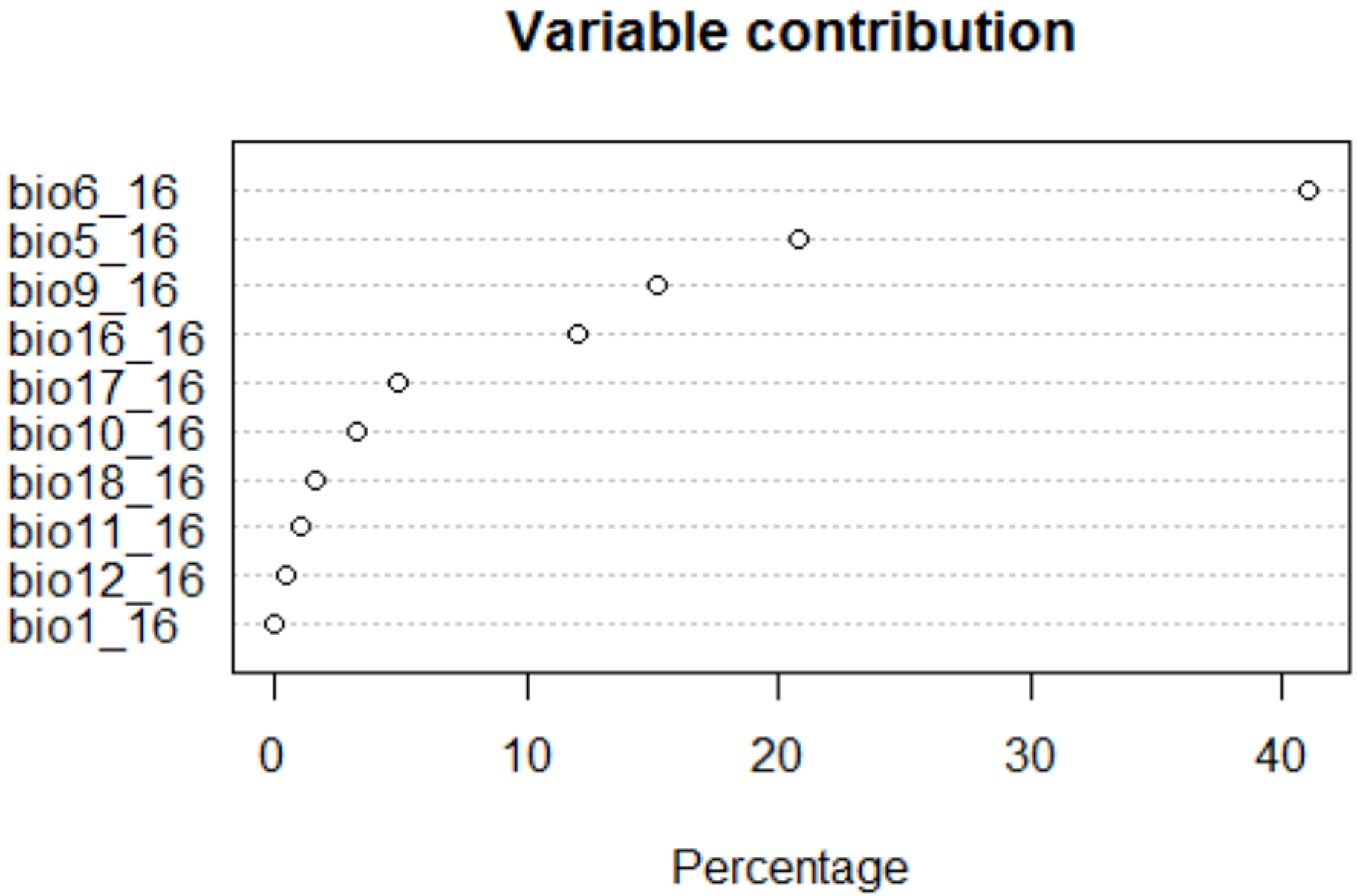
Predicted (model)	**presence +**	**absence -**
presence +	true +	false +
absence -	false -	true -

Sensitivity= (**T+**) rate

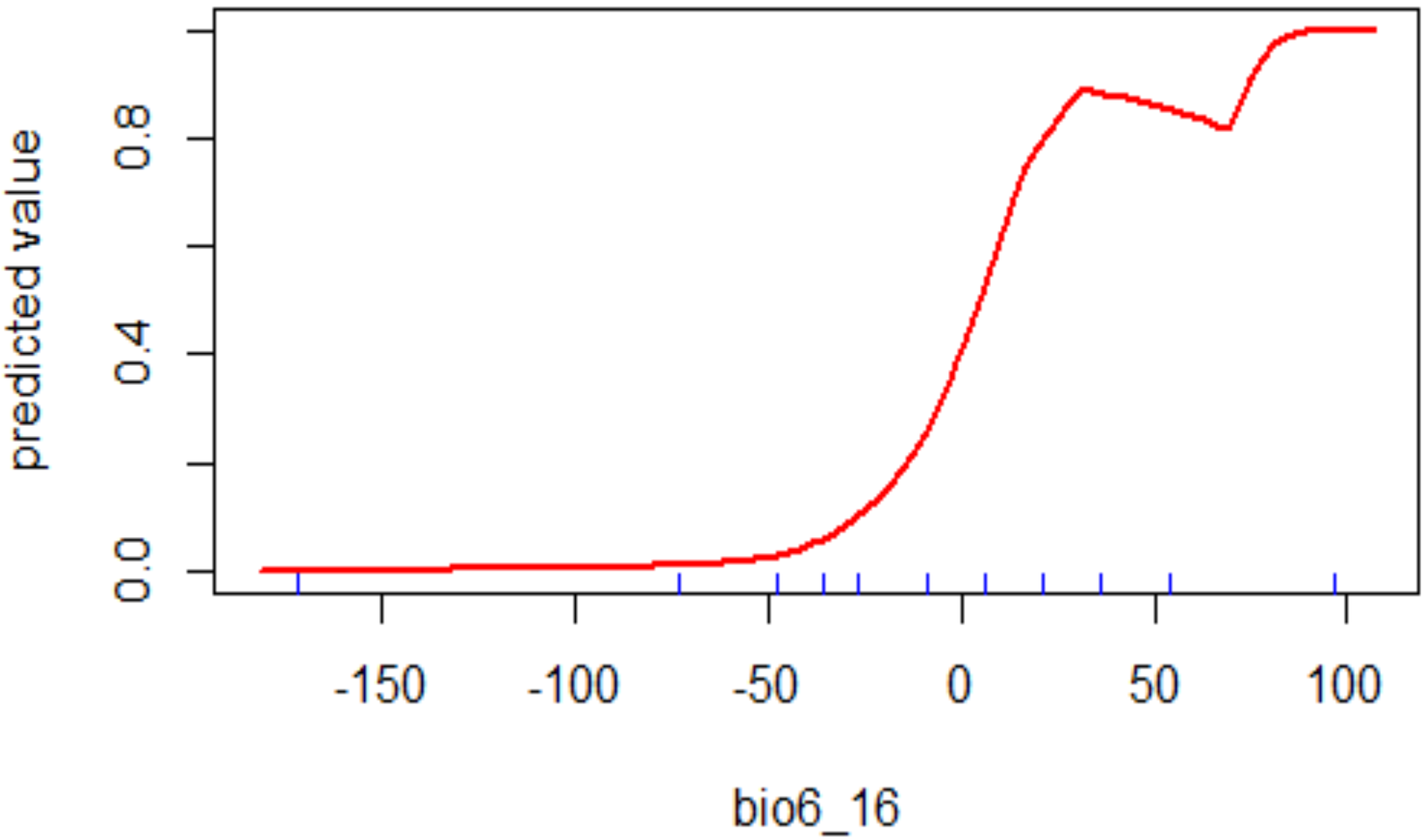
Specificity= (**T-**) rate

Let’s have another example with MaxEnt

<u>Analysis of predictors' contribution</u>: how much predictors contribute to fit the model? In each iteration, Maxent tracks how much the model gain when single variable value is modified. At the end of the run, all these small gains are summed for each variables and converted to percentage.

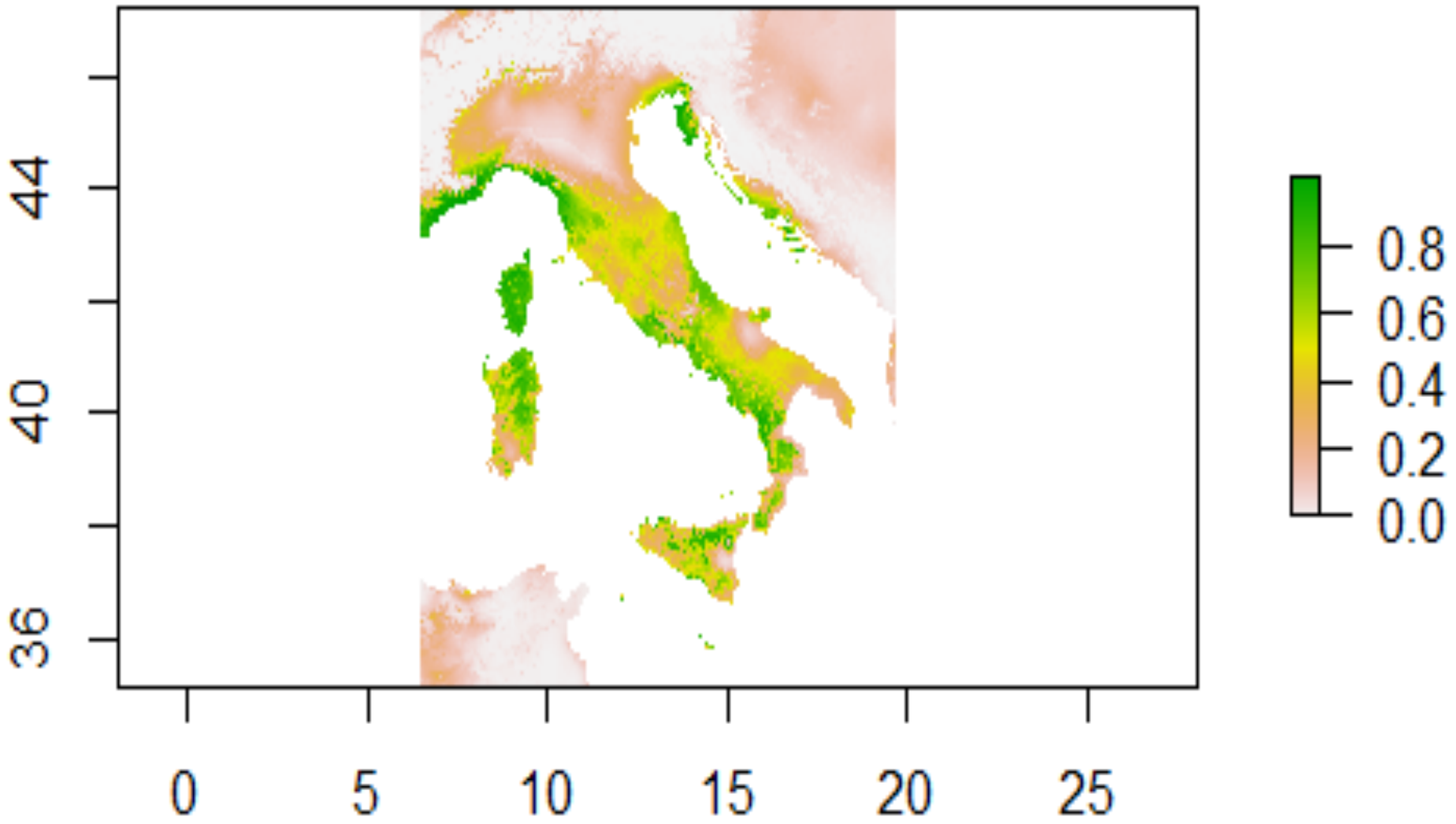


<u>Response curves</u>: shows how variables influence the probablity of occurrrence.
Y= predicted probability of presence
X= values of the variable



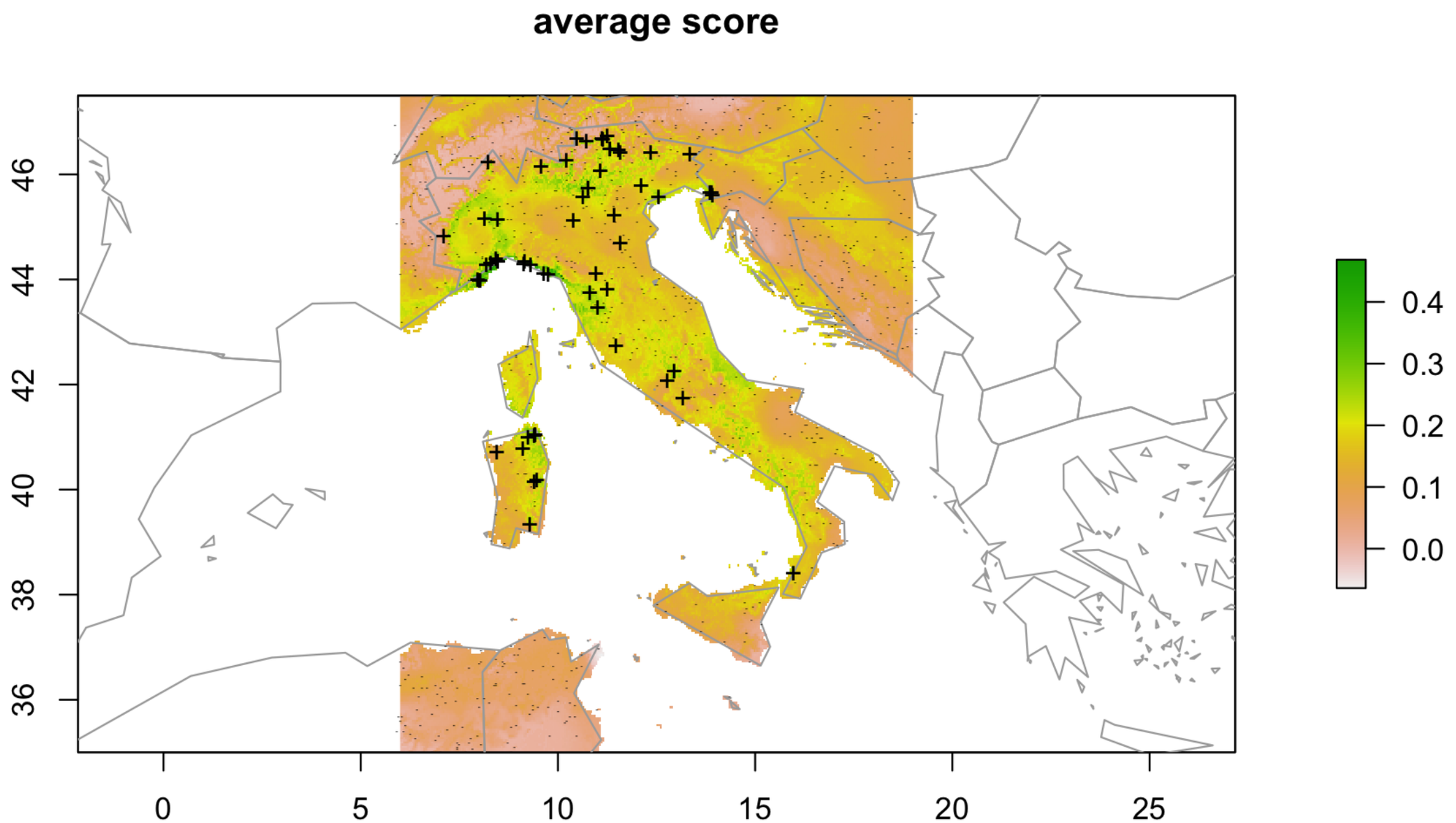
The curves display how the predicted probability of presence changes when a predictor varies, holding all the other predictors fixed to their median value.

<u>Suitability map</u>: the study area is here represented with scaled colours based on the probability of presence for each cell: 0= non suitable, 1= highly suitable.



Ensemble

Let’s switch to R again, and wrap up a little


average score
46
44
42
40
38
36
0
5
10
15
20
25
0.4
0.3
0.2
0.1
0.0

A first model