Species distribution models

Pinus sylvestris

Prof. Dr. Loic Pellissier

August 2, 2016

Contents

1 Introduction[1]

Scots pine (*Pinus sylvestris*) is a species of pine that is native to Eurasia, ranging from Western Europe to Eastern Siberia, south to the Caucasus Mountains and Anatolia, and north to well inside the Arctic Circle in Scandinavia. In the north of its range, it occurs from sea level to 1,000 m, while in the south of its range it is a high altitude mountain tree, growing at 1,200-2,600 metres (3,900-8,500 ft) altitude. It is readily identified by its combination of fairly short, blue-green leaves and orange-red bark.

[1]Introduction text and images taken from wikipedia https://en.wikipedia.org/wiki/Scots_pine

Figure 1: Native Scots pine at Glenmuick, Scotland

1.1 Distribution

Scots pine is the only pine native to northern Europe, forming either pure forests or alongside Norway spruce, common juniper, silver birch, European rowan, Eurasian aspen and other hardwood species. In central and southern Europe, it occurs with numerous additional species, including European black pine, mountain pine, Macedonian pine, and Swiss pine. In the eastern part of its range, it also occurs with Siberian pine among other trees.

1.2 Study Area

In this practical the distribution of *Pinus sylvestris* in Switzerland will be modeled. In Switzerland, where the height above sea level ranges from 193 to 4,634 meters (633 to 15,203 ft), a large variety of climates are found, and as a consequence, many diverse forest communities.

In most inhabited regions of Switzerland, at lower altitudes, the weather is generally moderate. On the Plateau, freezing temperatures generally occur during December to early March with an average temperature of 9 °C (48.2 °F) for elevations between 500-600 meters (1,640-1,969 ft). On the Plateau the average precipitation is 1,000 millimeters (39 in) with a range of about 800-1,300 millimeters (31.5-51.2 in). The higher elevations of the Jura and the Alps naturally cause lower temperatures and in the high Alps glaciers exist. The Jura and foothills (both north and south of the Alps) typically have more precipitation with an average of 1,200-1,600 millimeters (47.2-63.0 in) while the high Alps may have over 2,500 millimeters (98.4 in). Ticino, on the south side of the Alps, has sub-tropical vegetation and is usually 2-4 °C (3.6-7.2 °F) warmer, and wetter than the Swiss Plateau.

2 Uploading data in R

Question 2.0.1 *Which environmental factor would you say best explains the distribution of the Pinus sylvestris species? Think about the factors that might limit the growth of a tree.*

We will model the distribution of the Pinus sylvestris species using climatic variables. We now load the relevant data.

2.1 Response variable: species' data

We will now load a sample of locations where the presence of Pinus sylvestris was corroborated.

Load the occurrence data:

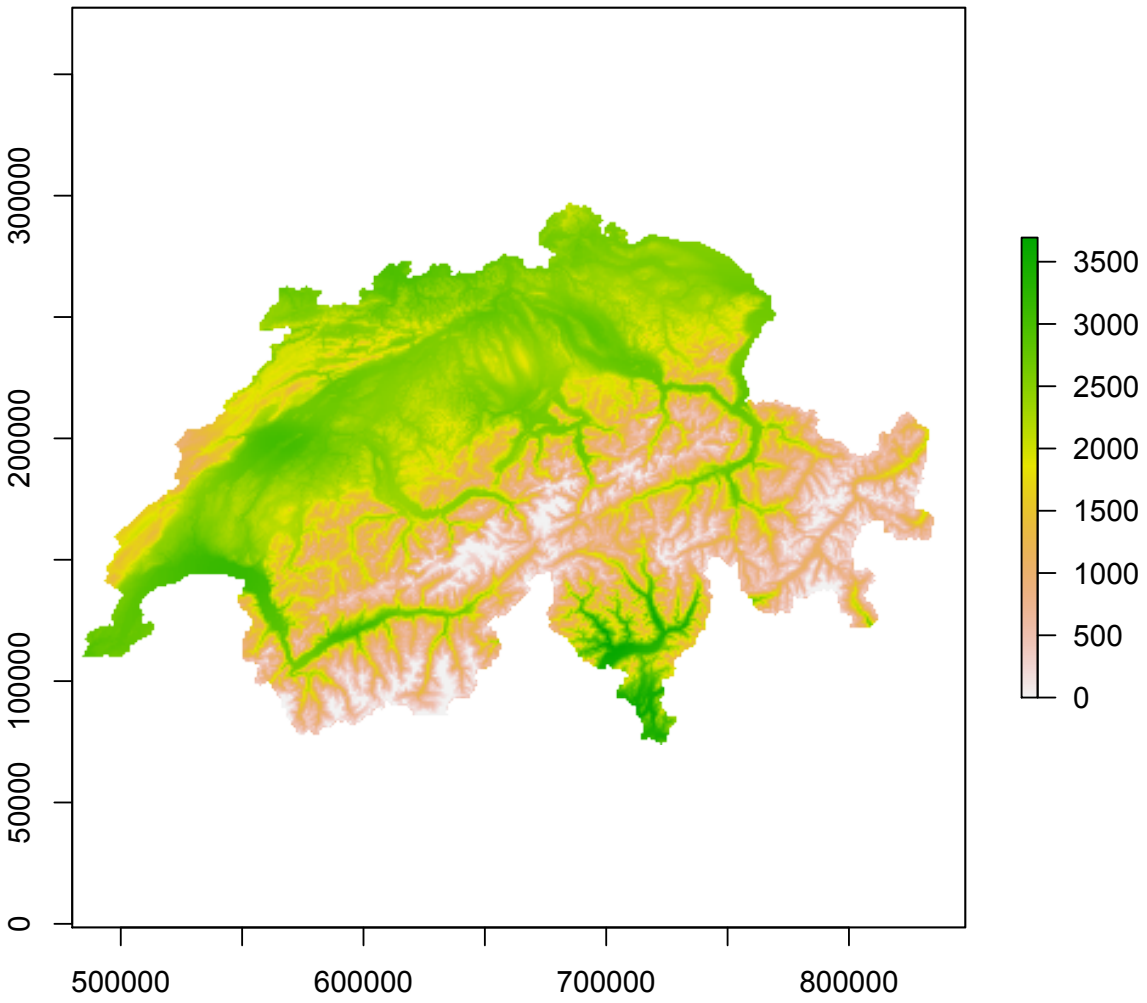

Figure 2: Study area in Switzerland with temperature as background representing the energy available for tree growth. The climatic maps were obtained by combining meteorological records from the Swiss meteorological stations and a digital elevation model mapping the elevation across Switzerland. Because temperature decrease linearly with elevation, it is possible to extrapolate the meteorological information from the station to the entire Switzerland.

```
Occurrences <- read.csv("../data/occurrences.csv", h = T)
```

Vizualise the data:

```
pander(head(Occurrences), caption = "This table contains the location of tree occurrences. Botanists hav
```

number	x	y	elevation	exposition	slope	cover	presence
342	491900	116150	390	NW	65	3	1
340	493250	118150	405	SW	25	90	1
341	494200	116600	410	NE	12	62.5	1
3046	499220	127150	427	–	0	0.5	1
2554	5e+05	85000	700	N	36	2.5	1
10602	502920	141980	620	SE	7	0.5	1

Table 1: This table contains the location of tree occurrences. Botanists have conducted over 12,000 forest inventories accross Switzerland, recording the cover of all tree species. Here, the forest plots containing Pinus sylvestris are indicated with a "1", and the cover of the species in the plot is provided. The plots without Pinus sylvestris are indicated with a "0". The coordinates of the plots is provided in the Swiss coordinate system "CH1903". Finally, a few environmental predictors, including elevation, slope and cover are provided.

2.2 Predictors: climatic data

Temperature is a major physiological limitation for many different species including trees. For instance only a few specialized species have the adaptations necessary to tolerate the cold temperatures observed at higher elevation. We will use degree-days, the number of degrees above a threshold of 5°C, as a proxy for energy available. Solar radiation is another relevant measure of energy, complementary to temperature. This is because temperature measured by meteorological stations corresponds to standardized **shade** temperature. Trees rely on water to survive so it is likely that a good predictor of pinus sylvestris presence at a location will be water availability. We will use the moisture index in the month of July, the driest and of greatest significance to species occurrence, as a proxy for water availability .

The three climatic rasters are *degree-days*, *moisture index* and *solar radiation*:

- DDEG is the number of degrees above a threshold of 5°C, a proxy for energy available.
- MIND is the moisture index in the month of July, computed as the difference between precipitation and potential evapotranspiration, and
- SRAD is solar radiation,

We will now load all three climatic rasters for Switzerland. These rasters were obtained using climatic observations at meteorological stations and using them to model the desired variables at all raster cell locations for Switzerland. Other variables such as elevation and slope which are available at all locations were used as predictors for the desired climatic variables. Interpolation models were also used to model spatial dependence not included in the predictor variables.

Import the library containing required functions:

```
library(raster)
```

The `raster` function from the R package `raster` creates a raster object in one of several ways, including by reading a file that can be interpreted by the *gdal* driver, which is how we use it here. A raster object consists primarily of:

- A grid of cells,
- A coordinate reference system (CRS) for the grid and its cells so that we know the location to which the grid refers,
- A variable of interest for which each cell in the grid has a value, and,
- Other information relating to the GRS, projection, resolution, etc.

Load the three climatic rasters using raster function:

```
DDEG <- raster("../data/ddeg_1km")
MIND <- raster("../data/mind_1km")
SRAD <- raster("../data/srad_1km")
```

Vizualise the climatic maps that you have loaded:

```
par(mar = c(0.1, 0.1, 0.1, 0.1), mfrow = c(3, 1))
library(RColorBrewer)
par(mar = c(3, 0.1, 1, 0.1), mfrow = c(2, 2))
plot(DDEG, col = rev(heat.colors(20)), box = F, axes = F, legend = F)
plot(DDEG, legend.only = T, horizontal = T, add = T, col = heat.colors(20),
    smallplot = c(0.25, 0.75, 0.08, 0.1))
plot(MIND, col = brewer.pal(9, "PuBu"), box = F, axes = F, legend = F)
plot(MIND, legend.only = T, horizontal = T, add = T, col = brewer.pal(9,
    "PuBu"), smallplot = c(0.25, 0.75, 0.08, 0.1))
plot(SRAD, col = brewer.pal(9, "YlOrRd"), box = F, axes = F,
    legend = F)
plot(SRAD, legend.only = T, horizontal = T, add = T, col = brewer.pal(9,
    "YlOrRd"), smallplot = c(0.25, 0.75, 0.14, 0.16))
```

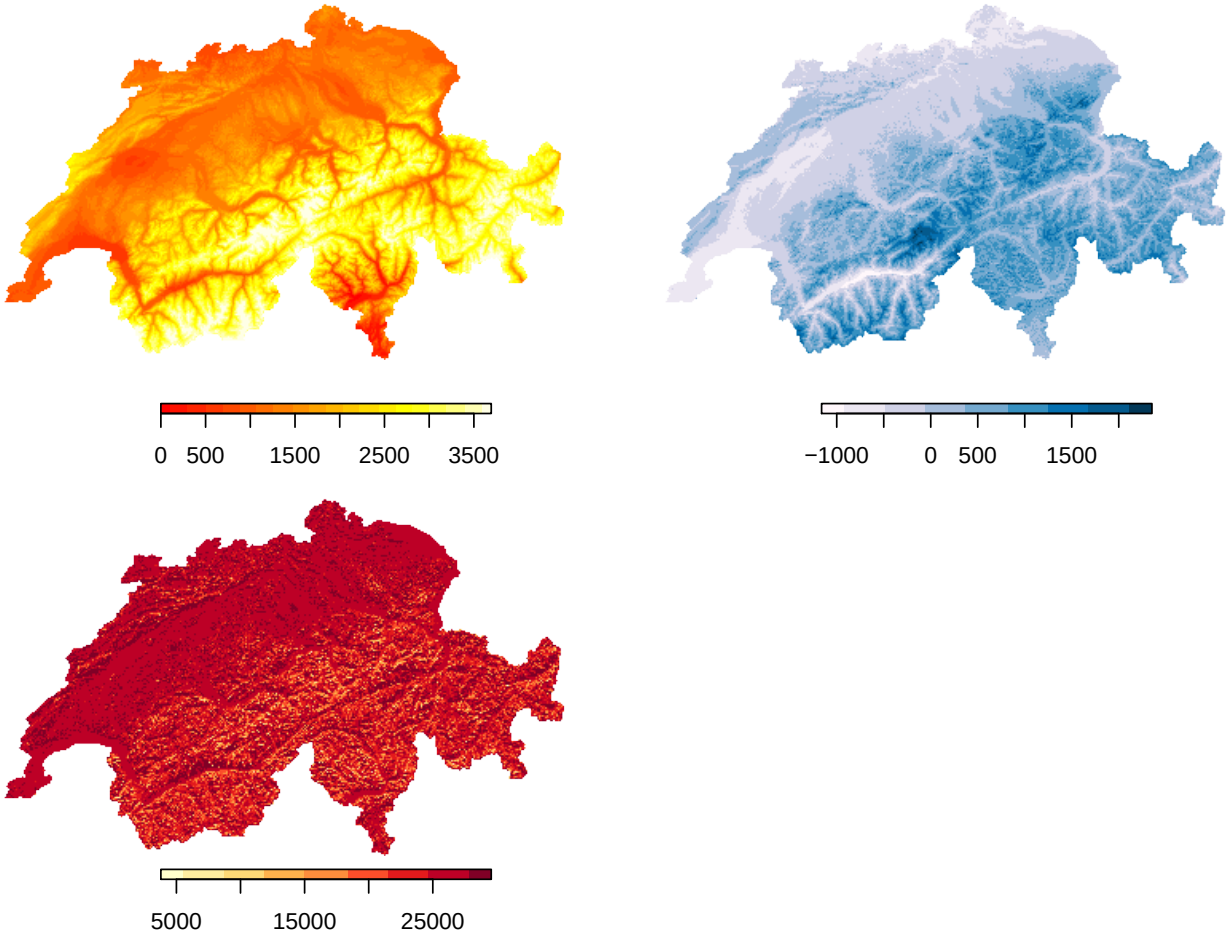

Figure 3: Maps of degree days (top left), moisture index (top right) and radiation (bottom left) for Switzerland (rasters).

Question 2.2.1 *Observe the differences in climate across Switzerland. Why do you think that degree-days, moisture and solar radiation are relevant to explain the biology and distribution of the Pinus sylvestris species in Switzerland?*

3 Link response variable to climatic predictors

Before fitting the species distribution model we need to link explanatory variables, in this case our three climatic predictors, to *occurrence* data points. The climatic data is in the form of rasters while the occurrence data is in dataframe format. This step is essential to link the information of your species presence/absence to the variables explaining the distribution of the species.

The `extract` function frome the R package `raster` is used to obtain the values of a raster object at the locations of other spatial data. The first argument is a raster object and the second, in this case, are the x-y coordinates we are interested in.

Create a dataframe containing climate variables corresponding to each data point in the Occurrences dataframe by using the function extract.

```
Climate <- cbind(DDEG = extract(DDEG, Occurrences[, c("x", "y")]),
    MIND = extract(MIND, Occurrences[, c("x", "y")]), SRAD = extract(SRAD,
        Occurrences[, c("x", "y")]))
```

We now have a dataframe with the degree-day, moisture index and solar radiation variables at the locations where we recorded the presence or absence of Pinus sylvestris.

Append climatic data to occurrences data and discard incomplete observations:

```
Occurrences <- cbind(Occurrences, Climate)
Occurrences <- na.omit(Occurrences)
```

Plot Climate raster together with absences and presences (random sample because otherwise we can't see map very well):

```
set.seed(1)
table(Occurrences$presence)/nrow(Occurrences) * 100
```

```
##
##        0        1
## 79.84136 20.15864
```

```
indx.oc <- sample(which(Occurrences$presence == 1), round(1000 *
    sum(Occurrences$presence == 1)/nrow(Occurrences)))
indx.ab <- sample(1:nrow(Occurrences), round(1000 * sum(Occurrences$presence ==
    0)/nrow(Occurrences)))
par(mfrow = c(2, 1), mar = c(0.1, 0.1, 0.1, 0.1))
plot(DDEG, col = rev(heat.colors(20)), box = F, axes = F)
points(Occurrences$x[indx.oc], Occurrences$y[indx.oc], col = "green4",
    pch = 3, cex = 0.5)
plot(DDEG, col = rev(heat.colors(20)), box = F, axes = F)
points(Occurrences$x[indx.ab], Occurrences$y[indx.ab], col = "blue",
    pch = 3, cex = 0.5)
legend("bottomright", legend = c("present", "absent"), pch = c(3,
    3), col = c("green4", "blue"))
```

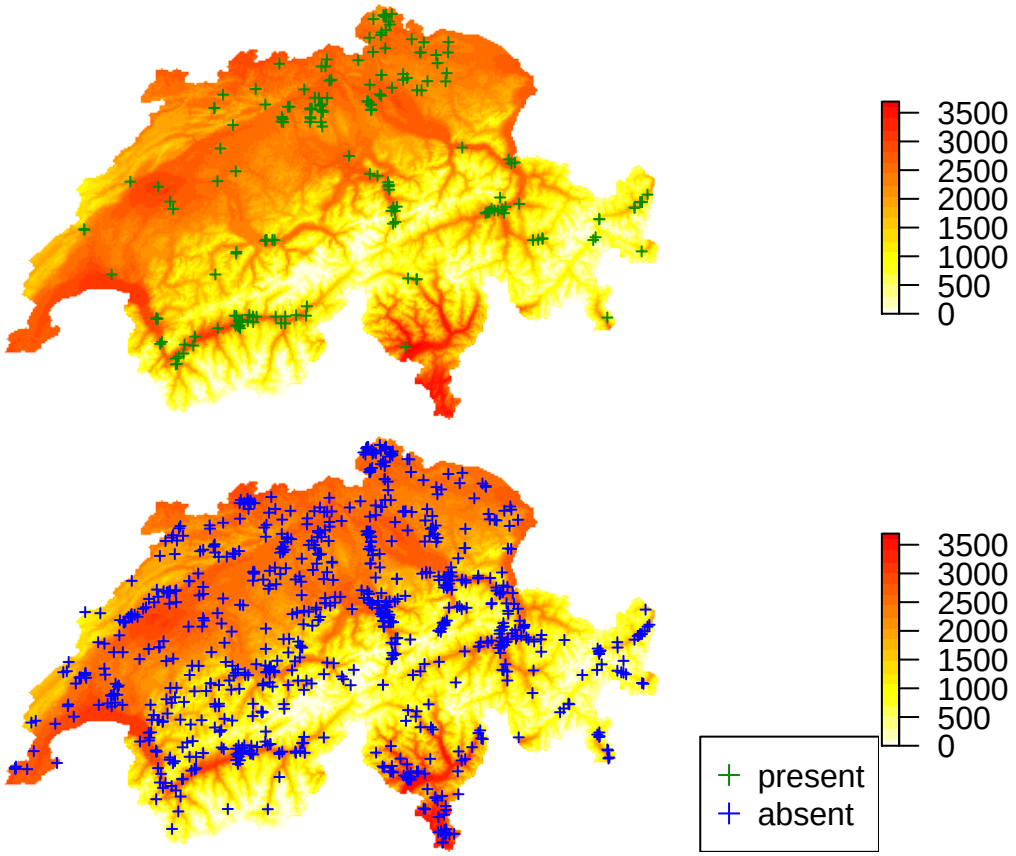

Figure 4: Map of degree-day (raster) with pinus sylvestris presences (top) and absences (bottom) overlayed.

Question 3.0.2 *Observe to distribution points of Pinus sylvestris over Switzerland. What can you conclude on the ecology of this species? Is it a specialist or a generalist species? What environmental factors could be limiting its distribution?*

Since we now have available the response variable and predictors at all sampled locations we can fit a model to describe the relationship that the former has with the latter.

4 Modeling Pinus sylvestris' distribution

We will use Generalized Linear Models (GLMs) to model the relationship between the distribution of Pinus sylvestris and climatic factors. We first briefly review some of the theory involved.

4.1 Generalized Linear Model (GLM) theory

The classic linear model can be described by the following equation:

$$y_i = \mathbb{E}[y_i|x_i] + \epsilon_i = \beta^T x_i + \epsilon_i \tag{1}$$

where:

- y is the response variable,
- x is a vector of predictors, and
- ϵ is the error term, a random variable such that:
  - mean is zero: $\mathbb{E}[\epsilon] = 0$,
  - variance is constant: $Var[\epsilon] = \sigma^2$,
  - independent identically distributed,
  - unbounded support,
  - typically we assume $y_i \sim N(\mu_i, \sigma^2)$

Although this is a useful model it has two important restrictions:

1. The range of y must not be restricted since the range of $\mathbb{E}[y|x] = \beta^T x \in (-\infty, \infty)$ is not. This is why y and ϵ are assumed to have a distribution with unbounded support such as the normal distribution.
2. The variance of y must be constant.

Here our aim is modeling pinus sylvestris presence/absence which is a binary variable and so, does not correspond to the properties of the classic linear regression model.

GLMs address both these issues. Like the linear model, a generalized linear model consists of a linear predictor $f(x) = \beta^T x$ and, additionally, two functions:

- A **link** function $g(\mu)$ that describes how the mean depends on the linear predictor:

$$g(\mu) = g(\mathbb{E}[y|x]) = \beta^T x \tag{2}$$

  This allows us to have a response variable with a restricted range. The function g could, *a priori*, be any function such that if $\mathbb{E}[y|x] \in (a, b)$ then $g(\mathbb{E}[y|x]) \in (-\infty, \infty)$.
- A **variance** function $V(\mu)$ that describes how the variance depends on the mean:

$$Var[y_i] = \phi V(\mathbb{E}[y|x]) = \phi V(\mu) \tag{3}$$

  where ϕ is a constant.

4.1.1 Normal response

Assume our data is distributed normally such that:

- $y_i \sim N(\mu_i, \sigma^2)$
- $\mu_i = \mathbb{E}[y_i|x_i] = \beta^T x_i$
- $Var[y_i|x_i] = \sigma^2$

Since $\mu_i \in (-\infty, \infty)$, a valid choice for the link function is:

$$g(\mu) = \mu \in (-\infty, \infty) \tag{4}$$

Also since $Var[y_i|x_i] = \sigma^2$ we have that $V(\mu) = 1$. Since a normal variable y can be written as the sum of its expectation μ and a zero mean normal with the same variance σ^2 as y we see that for this choice of link function we recover the classic linear model:

$$y_i = \mu_i + \epsilon_i = g^{-1}(\beta^T x_i) + \epsilon_i = \beta^T x_i + \epsilon_i \tag{5}$$

where $\epsilon \sim N(0, \sigma^2)$.

4.1.2 Bernoulli response

Assume our data is distributed Bernoulli such that:

- $y_i \sim Bernoulli(p_i)$
- $p_i = \mathbb{E}[y_i|x_i] = \beta^T x_i$
- $Var[y_i|x_i] = p_i(1 - p_i)$

Since $p_i \in (0, 1)$, a valid choice for the link function is the *log-odds ratio* or *logit* function:

$$g(p) = \log \frac{p}{1-p} \in (-\infty, \infty) \tag{6}$$

Also since $Var[y_i|x_i] = p_i(1 - p_i)$ we have that $V(p) = p(1 - p)$. For this choice of link function we obtain the **logistic** regression model:

$$\mathbb{E}[y_i|x_i] = p_i = g^{-1}(\beta^T x_i) = \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \tag{7}$$

$$Var[y_i|x_i] = p_i(1 - p_i) \tag{8}$$

Another valid choice for the link function is the *probit* function:

$$g(p) = \Phi^{-1}(p) \in (-\infty, \infty) \tag{9}$$

where $\Phi(p)$ is the normal cumulitive distribution function. This choice of link function leads to the **probit** regression model.

For species distribution models, where the response variable indicates the presence ($y_i = 1$) or absence ($y_i = 0$) of a species at sampled location a bernoulli response variable is appropriate and, *a priori* both the *logit* or *probit* functions are valid choices for the link function. Here we will use the *logit* link functon.

4.1.3 Poisson response

Assume our data is distributed Poisson such that:

- $y_i \sim Poisson(\lambda_i)$

- $\lambda_i = \mathbb{E}[y_i|x_i] = \beta^T x_i$
- $Var[y_i|x_i] = \lambda_i$

Since $\lambda_i \in (0, \infty)$, a valid choice for the link function is the *log* function:

$$g(\lambda) = \log \lambda \in (-\infty, \infty) \tag{10}$$

Also since $Var[y_i|x_i] = \lambda_i$ we have that $V(\lambda) = \lambda$. For this choice of link function we obtain the **Poisson** regression model:

$$\mathbb{E}[y_i|x_i] = \lambda_i = g^{-1}(\beta^T x_i) = e^{\beta^T x_i} \tag{11}$$

$$Var[y_i|x_i] = \lambda_i = g^{-1}(\beta^T x_i) = e^{\beta^T x_i} \tag{12}$$

For species richness models where the response variable indicates the number of different species ($y_i = 0, 1, 2, ...$) of a species at a sampled location a poisson response variable is appropriate and, the *log* function is a valid choice for the link function.

4.1.4 Residual analysis and diagnostics

In the classic linear regression setting the residuals are a *natural* quantity to study owing to the fact that the response variable can be expressed as a location model:

$$y_i = \mathbb{E}[y_i|x_i] + \epsilon = \mu_i + \epsilon \tag{13}$$

meaning that the residuals $r_i = y_i - \hat{\mu}_i$ are estimators of the error term ϵ. As we have seen in subsection **??** in the linear regression setting we can look at certain plots of the residuals to verify model assumptions:

- The Tukey-Anscombe plot to check the unbiasedness of the model: $\mathbb{E}[r_i] = 0$
- The Scale-location plot to check the homoscedasticity of errors: $Var[r_i] = k$
- The normal Q-Q plot helps check that error term is distributed normally
- Cook and leverage plots help identify outliers: observations with atypical values for predictors (high leverage) and which have high influence on model estimation (high Cook's distance)..

For general linear models, in general we don't know how the residual term $r_i = y_i - \hat{\mu}_i$ is distributed since non-normal response variables y cannot, in general, be expressed as a location model $y = \mu + \epsilon$. We do however know that if the variance of the response variable is non-constant, as in the Bernoulli and Poisson case, then the residuals will also have non-constant variance. We also know that the residuals need not be distributed normally so the Scale-location and normal Q-Q plots do not apply in the GLM setting.

In order to check for the unbiasedness of the model we look at a modified Tukey-Anscombe plot constructed with *Pearson* residuals which are the residuals standardized by dividing by the estimated standard deviation of the response variable. Standardizing the residuals helps us to visualize whether the expected value of the residuals is zero for any given level of the linear predictor. The Cook and leverage plots to help identify outliers also applies for GLMs.

4.1.5 GLMs in R

We use the `glm` function from the `stats` package to fit our generalized linear model. Like the `lm` function the `glm` function needs at least *formula* and *data* parameters where:

- **formula** could, for example, be y~poly(x,2)+ I(z^2) + I(w==0) + I(w>0):I(x<0) + I(w>0)*I(z<0), where,
  - the I() term is used to transform existing variables,
  - the poly(x,n) term, creates a polynomial of degree n: $x + x^2 + ... + x^n$,
  - I(w>0):I(x<0) term, creates an interaction term between the indicator functions $1_{w>0}(w)$ and $1_{x<0}(x)$, and
  - I(w>0)*I(z<0) term, creates the indicator functions $1_{w>0}(w)$ and $1_{z<0}(x)$ and additionally their interaction.
- **data** indicates the data.frame where the x,y,z,w variables are located.

Additionally `glm` needs a *family* parameter which indicates the type of dependent variable and the link function to be used. For example:

- family= "binomial" or family=binomial("logit") indicates a binary dependent variable with logit link function,
- family= binomial("probit") indicates a binary dependent variable with probit link function, and
- family= poisson indicates a poisson dependent variable (values in 0,1,2,...) with a log link function.

As in the case with other R fitting functions which create *model* objects, the glm objects created by the fitting function `glm` has available certain standard functions:

- `summary`: gives a summary of estimated parameters, their significance and the overall significance of the model,
- `plot`: helps visualize he fitted model and/or gives model diagnostics to assess the validity of model assumptions, and
- `predict`: applies the model to new data points, provided that the model predictors are available, to predict the dependent variable (and probabilites in the case of glm objects).

- `coef`: provides model coefficients (the β's in the case of linear and generalized linear models).
- `fitted`: provides fitted values (the $\hat{y}$'s)
- `residuals`: provides model residuals.

4.2 Model estimation

We will now fit logistic regression models based on climatic variables to explain the distribution of Pinus Sylvestris in Switzerland.

Run a GLM for each predictor (degree-days and moisture index) separately. Use a polynomial of order two for each corresponding predictor. The model is thus:

$$g(\mathbb{E}[y]) = g(p) = f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_n x^n \tag{14}$$

where

- y is the dependent variable, a binary variable that takes the value 1 or 0, indicating the presence or absence of the species,
- p is the expected value of y, in this case, the probability that $y = 1$,
- $g(p)$ is a link function, in this case log-odds ratio $\frac{p}{1-p}$,
- x is the predictor, for example, temperature or precipitation,
- $f(x)$ is a polynomial in x, and
- n is the order of the polynomial used, in this case $n = 2$.

Fit a General Linear Model (GLM) based on **degree-days** only:

```
glm.uni <- glm(presence ~ poly(DDEG, 2), data = Occurrences,
    family = binomial, maxit = 100)
pander(summary(glm.uni)$coefficients, caption = "Summary of GLM model based on degree-days.")
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.525	0.02497	-61.06	0
poly(DDEG, 2)1	69.91	3.76	18.59	3.664e-77
poly(DDEG, 2)2	-60.72	3.937	-15.42	1.117e-53

Table 2: Summary of GLM model based on degree-days.

In this GLM, a polynomial relationship is fitted which includes both a linear and a quadratic term in the relationship between the predictors and the response variable. This is provided by the *poly(var, 2)* argument.

To assess the contribution of individual predictors in a given model, one may examine the significance of the Wald statistic. The Wald statistic, analogous to the t-test in linear regression, is used to assess the significance of coefficients.

Since the coefficient corresponding to the linear and quadratic terms have opposite signs it is difficult to interpret the relationship between degree-days and probability of Pinus sylvestris presence in a simple way. We plot a degree-days vs. predicted response curve to visualize the relationship.

Plot a curve showing the relationship between degree-days and predicted probability of presence. Add observed response variable: presences in red and absences in blue:

```
par(mfrow = c(1, 1))
data_plot <- data.frame(cbind(Occurrences$DDEG, predict(glm.uni,
    type = "response")))
sort1 <- na.omit(data_plot[order(data_plot[, 1], decreasing = FALSE),
    ])
data_plot <- data.frame(cbind(sort1[, 1], sort1[, 2]))
plot(data_plot[, 1], data_plot[, 2], xlab = expression("Degree" -
    days ~ degree ~ C), ylab = "Probability of occurrence", frame.plot = F,
    type = "l", ylim = c(0, 1))
# Plot the presences in red and absences in blue
points(Occurrences$DDEG[Occurrences$presence == 1], Occurrences$presence[Occurrences$presence ==
    1], col = "red")
points(Occurrences$DDEG[Occurrences$presence == 0], Occurrences$presence[Occurrences$presence ==
    0], col = "blue")
```

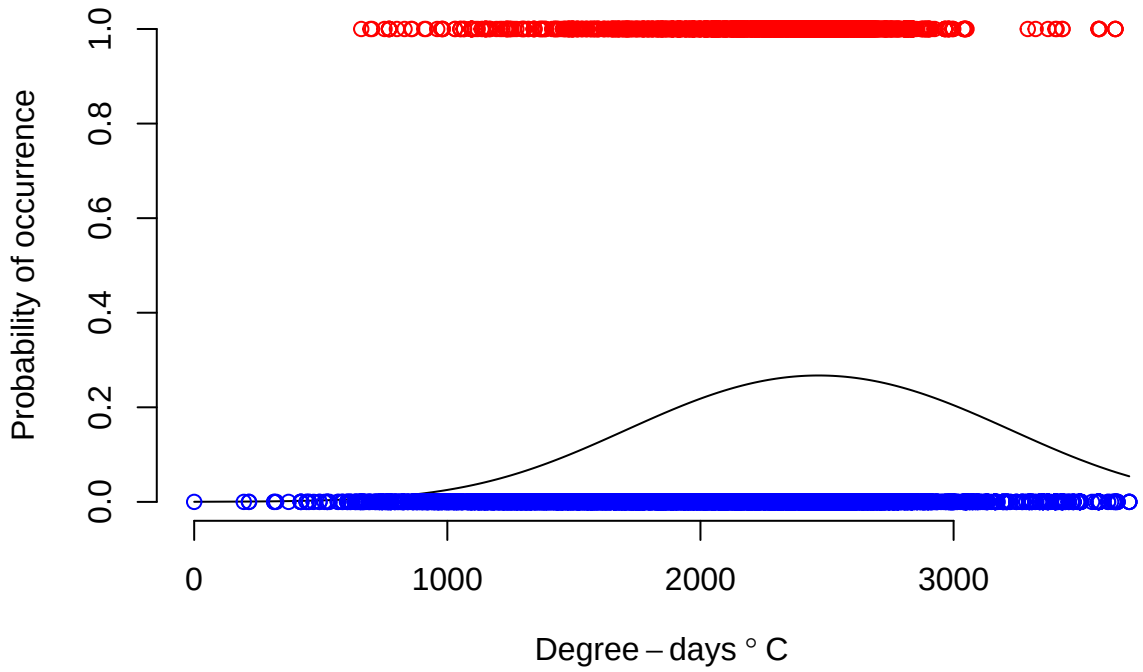

Figure 5: Univariate predictor (degree-days) vs. response scatter plots and fitted model curves. Black line indicates the the probability of Pinus sylvestris presence for a given level of degree-days as predicted by corresponding univariate GLM.

We observe that Pinus sylvestris show a wide distribution range along the gradient of degree-days with an optimum at approximately 2500. The probability remains however relatively low along the entire temperature gradient.

Fit a General Linear Model (GLM) based on **moisture index** only:

```
glm.uni <- glm(presence ~ poly(MIND, 2), data = Occurrences,
    family = binomial, maxit = 100)
pander(summary(glm.uni)$coefficients, caption = "Summary of GLM model based on moisture index.")
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.652	0.02824	-58.49	0
poly(MIND, 2)1	-120.4	4.242	-28.37	4.352e-177
poly(MIND, 2)2	-21.38	3.707	-5.768	8.027e-09

Table 3: Summary of GLM model based on moisture index.

In this case we see that coefficients corresponding to both the linear and quadratic term are negative indicating that, at least for the range of moisture index observed at sampled locations, the less moisture present the higher the probability of observing a Pinus sylvestris tree. We plot a degree-days vs. predicted response curve to corroborate this relationship.

Plot a curve showing the relationship between moisture index and predicted probability of presence. Add observed response variable: presences in red and absences in blue:

```
data_plot <- data.frame(cbind(Occurrences$MIND, predict(glm.uni,
    type = "response")))
sort1 <- na.omit(data_plot[order(data_plot[, 1], decreasing = FALSE),
    ])
data_plot <- data.frame(cbind(sort1[, 1], sort1[, 2]))
plot(data_plot[, 1], data_plot[, 2], xlab = "Moisture index",
    ylab = "Probability of occurrence", frame.plot = F, type = "l",
    ylim = c(0, 1))
# Plot the presences in red and absences in blue
points(Occurrences$MIND[Occurrences$presence == 1], Occurrences$presence[Occurrences$presence ==
    1], col = "red")
points(Occurrences$MIND[Occurrences$presence == 0], Occurrences$presence[Occurrences$presence ==
    0], col = "blue")
```

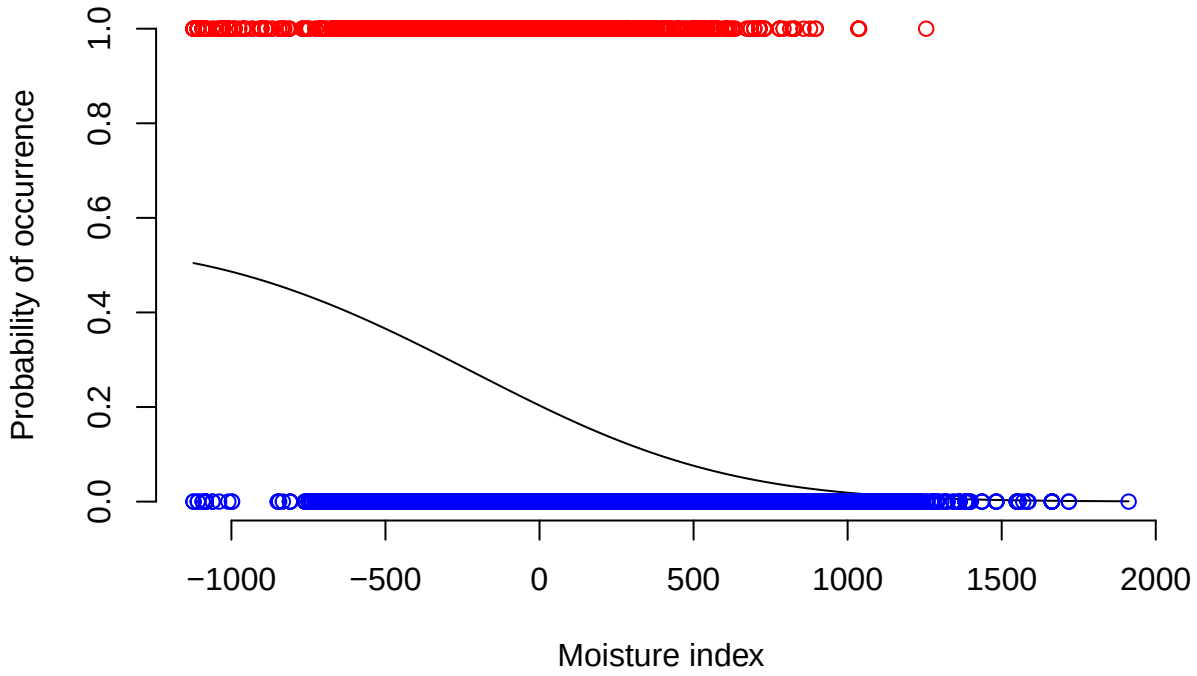

Figure 6: Univariate predictor (moisture index) vs. response scatter plots and fitted model curves. Black line indicates the the probability of pinus sylvestris presence for a given level of moisture as predicted by corresponding univariate GLM.

We confirm that the drier a location the higher the probability of a Pinus sylvestris presence according to the fitted model. However, we have not adjusted for the effects of other predictors here so this univariate model could be misleading and should be considered with care: a multivariate model which considers all relevant predictors concurrently, and which thus has more validity, may include opposite signs for the linear and quadratic coefficients corresponding to moisture index.

Question 4.2.1 *Provide some general comments about the different response curves. Try to relate the shape to the distribution and the ecology of the species and the type of the predictors (degree-day, moisture-index based).*

Now run the GLM including the three climatic predictors- degree-days, moisture index and solar radiation- using linear and quadratic terms.

Fit a multivariate General Linear Model (GLM) based on degree-days and radiation:

```
glm.multi <- glm(presence ~ poly(DDEG, 2) + poly(MIND, 2) + poly(SRAD,
    2), data = Occurrences, family = binomial, maxit = 100)
pander(summary(glm.multi)$coefficients, caption = "Summary of GLM model based on three climatic predicto
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.737	0.02995	-58	0
poly(DDEG, 2)1	3.763	4.273	0.8807	0.3785
poly(DDEG, 2)2	-41.07	4.573	-8.982	2.662e-19
poly(MIND, 2)1	-146.1	5.276	-27.69	1.027e-168
poly(MIND, 2)2	-19.28	3.858	-4.999	5.76e-07
poly(SRAD, 2)1	-54.09	3.526	-15.34	4.068e-53
poly(SRAD, 2)2	-7.028	3.125	-2.249	0.02452

Table 4: Summary of GLM model based on three climatic predictors.

We can see that in the multivariate model the drier the location the higher the probability of observing Pinus sylvestris pinus. The two variables that are proxies for available energy- degree-days and solar radiation- have different signs for their coefficients. This makes it hard to interpret the relationship between available energy and probability of observing Pinus sylvestris at a given location.

4.3 Model diagnostics

We want to check that our model is unbiased by creating a Tukey-Anscombe plot and checking that the Pearson residuals do not have structure when compared to the linear predictor: the level of the Pearson residuals should be more or less the same regardless of the level of the linear predictor. If we see some *pattern* in the Pearson residuals when plotted against the linear predictor it means our model is biased. We also want to create a standardized leverage vs. Cook's distance plot to identify outliers in the dataset, points which possibly don't belong to the dataset and which have a big impact on the model estimation.

The R package `boot` has some useful functions for GLM diagnostics including `glm.diag` which, among other things, calculates the Cook's statistic and leverage of each observation which we use to identify outliers. High Cook's distance indicate influential data points - data points which are influencing the value of estimations - while high leverage indicate data points with unusual values for the predictor variables. We are interested in finding and possibly investigating further those points that have unusual predictor values and are influential since these could be data points that don't belong in our data set and are affecting model estimation.

For Cook's statistic we use the threshold value of

$$\frac{8}{n-2p} \tag{15}$$

to define points with high Cook's statistic (influential points), while for leverage we us the threshold value of

$$\frac{2p}{n-2p} \tag{16}$$

to define points with high leverage (unusual x values). These threshold values are recommended in the documentation of the `glm.diag` function.

First we obtain Cook's distance and leverage for all observations using glm.diag function:

```
library(boot)  #glm.diag
# glm diagnostics- in particular for cook and leverage plot
diags <- glm.diag(glm.multi)
cook <- diags$cook
levg <- diags$h
stdr.levg <- levg/(1 - levg)
```

Next we set cook and leverage thresholds:

```
n <- nrow(Occurrences)
p <- glm.multi$df.null - glm.multi$df.residual + 1
cook.thrsh <- 8/(n - 2 * p)
levg.thrsh <- 2 * p/(n - 2 * p)
```

Next we obtain linear predictors $\beta^T x_i$ and Pearson residuals r_i for all observations i:

```
xx.fit <- predict(glm.multi, type = "response")
yy.fit <- residuals(glm.multi, type = "pearson")
```

We now create a smoother for the relationship Pearson residuals vs. linear predictor. This will serve as an estimation of the expected Pearson residual conditional on different values of the linear predictor. If the model has no bias it should be *reasonably* constant. This will be red line of Tukey-Anscombe plot and should be *reasonably* flat.

Now we create smoother of linear predictor vs. Pearson residuals:

```
ls.fit <- loess.smooth(xx.fit, yy.fit, family = "gaussian")
```

Finally we create Tukey-Anscombe and standardized levarage vs. Cookś distance plots:

```
# Tukey-Anscombe plot with pearson residuals
par(mfrow = c(1, 2))
# plot linear predictor vs. pearson residuals
plot(xx.fit, yy.fit, xlab = "linear predictor", ylab = "Pearson residuals")
# Add smoother for real Pearson residuals (possibly having
# structure) vs. linear predictor
lines(ls.fit$x, ls.fit$y, col = "red")
abline(h = 0, lty = 3)

# Cook distance and leverage plot
plot(stdr.levg, cook, xlab = "Standardized leverage", ylab = "Cook statistic")
abline(h = cook.thrsh, v = levg.thrsh, lty = 2)
```

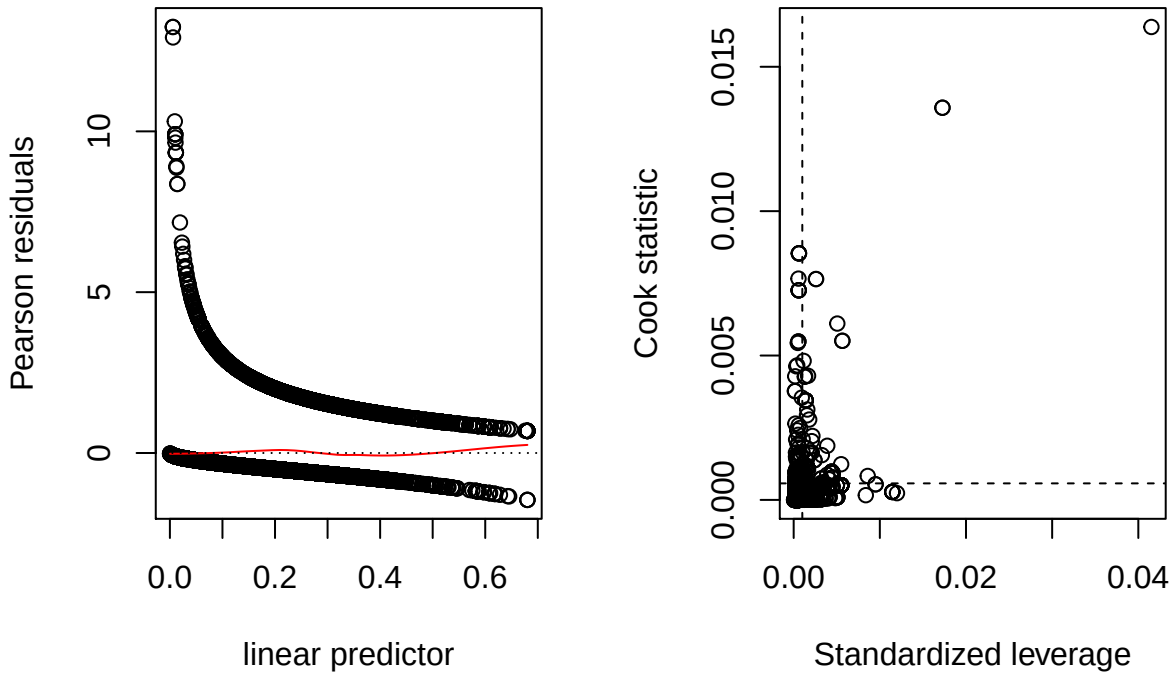

Figure 7: Generalized linear model diagnostics: Tukey-Anscombe plot (left) for identifying model bias and Cook distance vs. leverage plot (right) for identifying outliers.

We can see that the behavior of the residuals is reasonably unbiased. We now identify outliers: those observations that exceed the Cook's distance and leverage thresholds simultaneously.

Identify influential outliers:

```
indx.out <- which(stdr.levg > levg.thrsh & cook > cook.thrsh)
length(indx.out)/nrow(Occurrences) * 100
```

```
## [1] 0.6502787
```

```
outs <- Occurrences[indx.out, ]
```

0.67 % of the data is made up of outliers.

Plot outliers:

```
par(mar = c(0, 0, 0, 0))
plot(DDEG, col = rev(heat.colors(20)), box = F, axes = F)
points(Occurrences$x, Occurrences$y, col = c("green4", "blue")[Occurrences$presence +
    1], pch = 3, cex = 0.5)
points(outs$x, outs$y, cex = 3, col = "black")
legend(750000, 90000, legend = c("present", "absent", "outlier"),
    pch = c(3, 3, 1), col = c("green4", "blue", "black"))
```

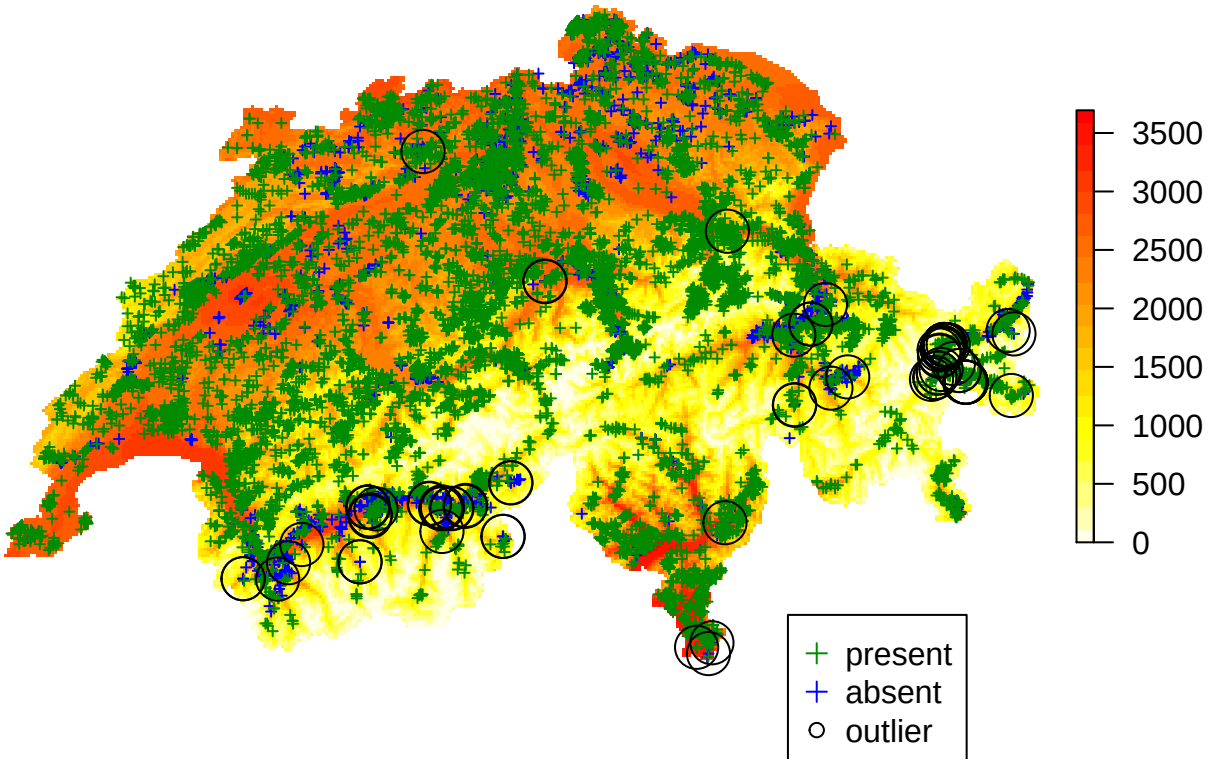

Figure 8: Map of degree-days (raster) with pinus sylvestris occurrence and absences, and outliers overlayed.

We can see that most outliers are located in the Rhone valley and in the eastern most part of the canton of Graubunden. To investigate more we could make raster plots of all the predictors available to try and understand why these are special locations.

4.4 Model evaluation

To evaluate the model one can use a contingency table to calculate a number of useful metrics (see Fielding & Bell 1997 and Allouche et al. 2006 for further details). The structure of a contingency table is usually formalized as follows:

predicted	Observed		Total
Null/Alternative	0	1	
0	U	T	W
1	V	S	R
Total	N_0	N_1	N

Table 5: Contingency table showing partition of observations according to predicted and observed presence/absence outcome.

In the following table some useful metrics based on the contingency table are summarized.

Metric	Formula
Prevalence	$\frac{N_1}{N}$
Correct classification rate	$\frac{S+U}{N}$
Mis-classification rate	$\frac{V+T}{N}$
Positive predictive power	$\frac{S}{R}$
Negative predictive power	$\frac{U}{W}$
Sensitivity (True positive rate)	$\frac{S}{N_1}$
Specificity (True negative rate)	$\frac{U}{N_0}$
False positive rate	$\frac{V}{N_0}$
False negative rate	$\frac{T}{N_1}$
Odds ratio	$\frac{U*S}{T*V}$
Kappa	$\frac{\frac{U+S}{N} - \frac{R*N_1+W*N_0}{N^2}}{1-\frac{R*N_1+W*N_0}{N^2}}$
True Skill Statistic (TSS)	$\frac{U*S-T*V}{(S+T)*(U+V)}$ = sensitivity + specificity -1

Table 6: Accuracy metrics for evaluating a logistic model.

Kappa and TSS are among the most widely used metrics. They both give equal weight to correct prediction of presence and absence and also correct for chance performance. We will focus on the TSS metric for evaluating our model.

First create a dataframe containing the observed presence/absence of Pinus sylvestris and the predicted probability of occurrence for the response variable at each sampled location:

```
data_validation <- data.frame(cbind(Occurrences$presence, predict(glm.multi,
    type = "response")))
colnames(data_validation) <- c("Observed", "Projected")
```

Predictions generated by most modeling techniques are on a continuous scale between 0 and 1, while the observations are binary (i.e. 0 or 1). To compare the values we thus need to change predictions from continuous to binary meaning we need a *threshold* to change the probabilities to zeros or ones. Let's start with a simple arbitrary threshold of 0.5: everything that is smaller or equal to 0.5 becomes 0 and everything that is bigger than 0.5 becomes 1.

The Functions.R file contains necessary functions for model evaluation. The `meva.table` function takes three arguments:

- Projected probability of occurrence,
- Observed occurrence/absence, and,
- threshold value.

This function first transforms the probabilites into occurrences or absences using the threshold value, then builds a contingency table as in table 5 and finally calculates the accuracy metrics of table 6.

Load Functions.R file with necessary functions for model evaluation. Compute the table of accuracy statistics with an arbitrary threshold of 0.5:

```
source("../R/Functions.r")
meva.table(data_validation$Projected, data_validation$Observed,
    0.5)
```

```
## $CONTINGENCY_TABLE
##                  Observed values
## Predicted values     0     1
##            FALSE 11082  2551
##            TRUE     91   270
##
## $EVALUATION_METRICS
##    Metric                           Value
## 1  "Prevalence"                     "0.2016"
## 2  "Correct classification rate"    "0.8112"
## 3  "Mis-classification rate"        "0.1888"
## 4  "Sensitivity"                    "0.0957"
## 5  "Specificity"                    "0.9919"
## 6  "Positive predictive power"      "0.0081"
## 7  "Negative predictive power"      "0.9043"
## 8  "False positive rate"            "0.7479"
## 9  "False negative rate"            "0.8129"
## 10 "Odds Ratio"                     "12.8893"
## 11 "Kappa"                          "0.1299"
## 12 "Normalized mutual information" "0.9633"
## 13 "True skill statistic"           "0.0876"
```

You can calculate evaluation metrics for a series of possible thresholds on the training dataset. We then choose the threshold that optimizes a given metric on this set and use it to calculate the same metric on the independent evaluation dataset. We will use a training set consisting of 70% of the data and use the remaining 30% for calculation of the evaluation metrics. By calculating metrics on separate data as is used to optimize the threshold we avoid overfitting the data which results in over-optimistic evaluation metrics.

In this case we want to maximize TSS. The custom function `max.TSS` calculates TSS for all threshold values between 0 and 1, for all 0.01 increment steps and automatically returns the maximum value of TSS and the corresponding threshold. The custom function `plot.tss` does the same and additionally plots a line graph going through all threshold-TSS pairs:

First separate data into training (70%) and evaluation sets (30%):

```
set.seed(5)
indx.train <- sample(1:nrow(data_validation), size = round(0.7 *
    nrow(data_validation)))
indx.eval <- setdiff(1:nrow(data_validation), indx.train)
```

Plot threshold-TSS curve for training data using plot.tss function

```
plot.tss(data_validation$Projected[indx.train], data_validation$Observed[indx.train])
```

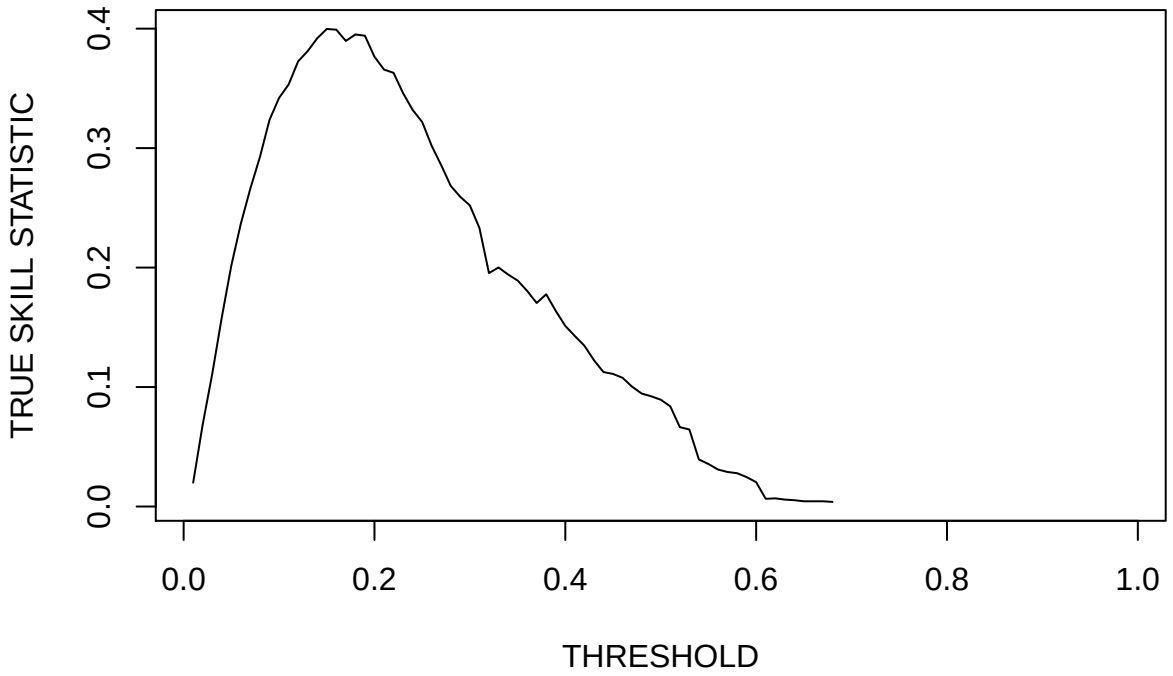

Figure 9: True skill statistic (TSS) vs. threshold value used for fitted GLM model.

Calculate optimum TSS value for training data using max.TSS function:

```
max_TSS <- max.TSS(data_validation$Projected[indx.train], data_validation$Observed[indx.train])
max_TSS[[2]]
```

```
##      [,1]                      [,2]
## [1,] "Maximum TSS"             "0.3997"
## [2,] "Correspondent threshold" "0.15"
```

Compute the table of accuracy statistics for the evaluation data with the optimized threshold of 0.15:

```
(tab <- meva.table(data_validation[indx.eval, 2], data_validation[indx.eval,
    1], as.numeric(max_TSS[[2]][2, 2])))
```

```
## $CONTINGENCY_TABLE
##                  Observed values
## Predicted values    0    1
##            FALSE 1618  112
##            TRUE  1699  769
##
## $EVALUATION_METRICS
##    Metric                          Value
## 1  "Prevalence"                    "0.2099"
## 2  "Correct classification rate"   "0.5686"
## 3  "Mis-classification rate"       "0.4314"
## 4  "Sensitivity"                   "0.8729"
## 5  "Specificity"                   "0.4878"
## 6  "Positive predictive power"     "0.5122"
```

```
## 7  "Negative predictive power"     "0.1271"
## 8  "False positive rate"           "0.3116"
## 9  "False negative rate"           "0.9353"
## 10 "Odds Ratio"                    "6.5387"
## 11 "Kappa"                         "0.2171"
## 12 "Normalized mutual information" "0.9022"
## 13 "True skill statistic"          "0.3607"
```

We can see that the TSS for the evaluation data is 0.3607 which is lower than the TSS of 0.3997 for the training set. The latter value represents a more accurate estimate of the population TSS.

4.5 Projection of the species distribution model over Switzerland

Now you can visualize the predicted probability of a Pinus sylvestris presence according to our *multivariate* model for all Switzerland.

We first obtain a dataframe with the x-y coordinates of all cells in the Switzerland raster:

```
CellsXY <- na.omit(as.data.frame(DDEG, xy = T))
```

Now append the predictor values for these cells:

```
Projection <- na.omit(cbind(CellsXY[, c("x", "y")], DDEG = extract(DDEG,
    CellsXY[, c("x", "y")]), MIND = extract(MIND, CellsXY[, c("x",
    "y")]), SRAD = extract(SRAD, CellsXY[, c("x", "y")])))
```

We will now project multivariate model over Switzerland.

The function `rasterFromXYZ` can be used to create a raster from a matrix containing the x and y coordinates in the first two columns, and the corresponding raster value in the third. The x and y coordinates must be on a regular raster grid in order to obtain a raster with the desired resolution. This is because if the resolution is not specified explicitly, it is assumed to be the minimum distance between x and y coordinates. A resolution of up to 10 times smaller than the mininimum may be attempted if a regular grid can otherwise not be created.

Project model over all of Switzerland and store result as a raster.

```
pred.glm.multi <- predict(glm.multi, Projection, type = "response")
proj.glm.multi <- cbind(Projection[, c("x", "y")], pred = pred.glm.multi)
pinus_sylvestris <- rasterFromXYZ(proj.glm.multi[, c("x", "y",
    "pred")])
```

Plot raster of projected probabilities of occurrence.

```
par(mfrow = c(2, 1), mar = c(0.1, 0.1, 0.1, 0.1))
plot(pinus_sylvestris, axes = F, box = F, col = brewer.pal(9,
    "PuRd"))
points(Occurrences$x[indx.oc], Occurrences$y[indx.oc], col = "green4",
    pch = 3, cex = 0.5)
plot(pinus_sylvestris, axes = F, box = F, col = brewer.pal(9,
    "PuRd"))
points(Occurrences$x[indx.ab], Occurrences$y[indx.ab], col = "blue",
    pch = 3, cex = 0.5)
legend("bottomright", legend = c("present", "absent"), pch = c(3,
    3), col = c("green4", "blue"))
```

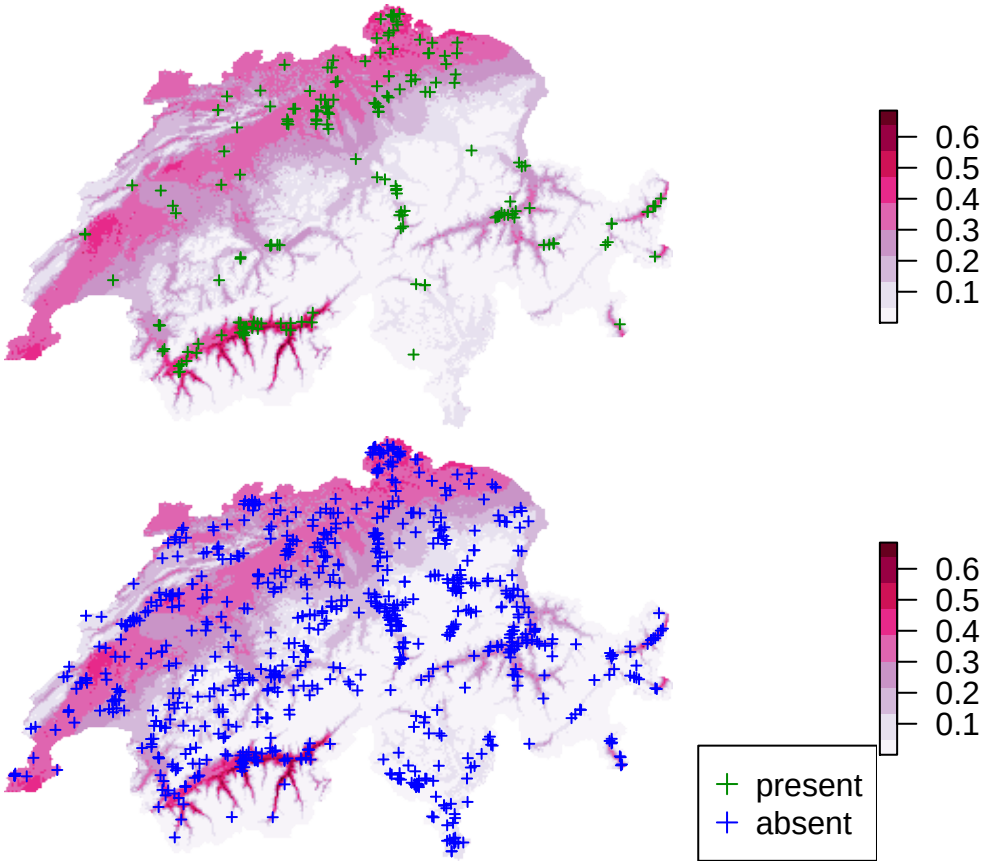

Figure 10: Map of projected probability of tree presence (raster) over Switzerland. Displays the probability of pinus sylvestris presence as predicted by the fitted GLM model. Pinus sylvestris observed presences (top) and absences (bottom) overlayed.

Question 4.5.1 *Is there a good match between the locations predicted as suitable by the model and the presence and absence data?*

Now you can compute the binary distribution map by converting the probabilities to binary values using the optimal threshold computed above.

Convert probabilities in presence and absence:

```
pinus_sylvestris_binary <- pinus_sylvestris > as.numeric(max_TSS[[2]][2,
    2])
par(mar = c(0, 0, 0, 0), mfrow = c(2, 1))
plot(pinus_sylvestris_binary, box = F, axes = F, legend = F,
    col = c("grey", "green4"))
points(Occurrences$x[indx.oc], Occurrences$y[indx.oc], col = "black",
    pch = 3, cex = 0.5)
plot(pinus_sylvestris_binary, box = F, axes = F, legend = F,
    col = c("grey", "green4"))
points(Occurrences$x[indx.ab], Occurrences$y[indx.ab], col = "red",
    pch = 3, cex = 0.5)
legend(690000, 125000, legend = c("predicted occurrence", "predicted absence"),
    fill = c("green4", "grey"), bg = "white")
```

```
legend(520000, 125000, legend = c("observed occurrence", "observed absence"),
    col = c("black", "red"), pch = 3, bg = "white")
```

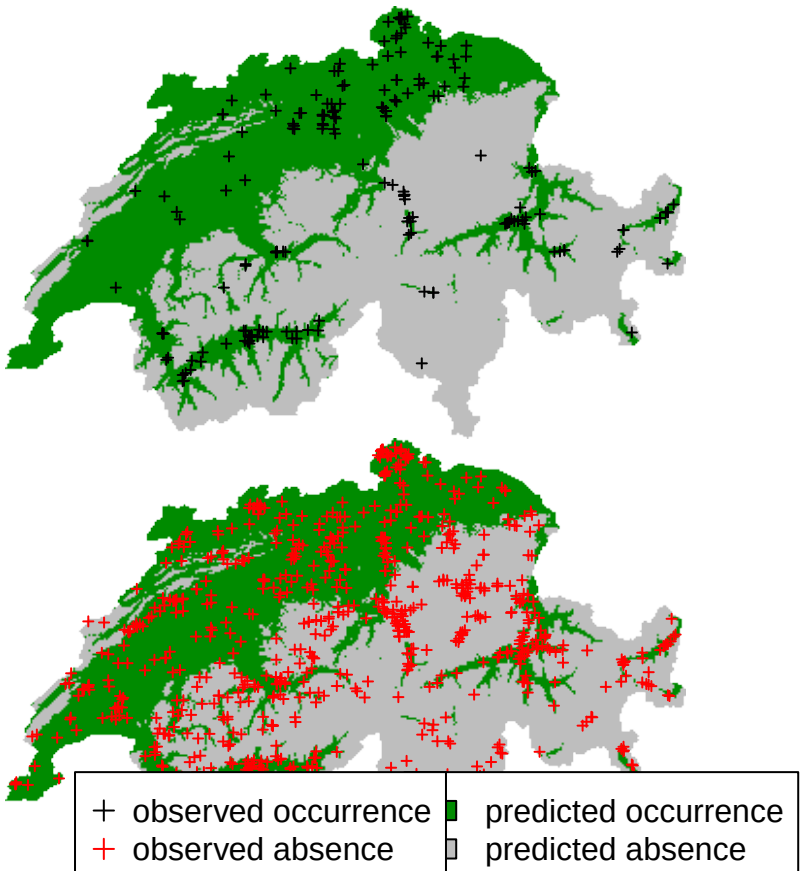

Figure 11: Projected presence/absence of pinus sylvestris (raster) for Switzerland. Pinus sylvestris presences (top) and absences (bottom) overlayed.

Question 4.5.2 *Why is it useful to convert species probability of occurrence to presence and absence?*

5 Installing rgeos and rgdal packages on MacOSX

Instructions for installing rgeos and rgdal packages on MacOSX:

- install xquartz, see http://www.xquartz.org/
- install GEOS and GDAL frameworks, see http://www.kyngchaos.com/files/software/frameworks/GDAL_Complete-1.11.dmg
- optionally XCode and Apple Command Line Developer Tools, see http://osxdaily.com/2014/02/12/install-command-line-tools-mac-os-x/
- optionally Fortran compiler, see http://stat.ethz.ch/CRAN/doc/manuals/r-release/R-admin.html#OS-X