Environmental Systems Data Science

Loïc Pellissier, Joshua Payne, Benjamin Stocker

2021-02-02

Contents

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

Chapter 2

Primers

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter **??**. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

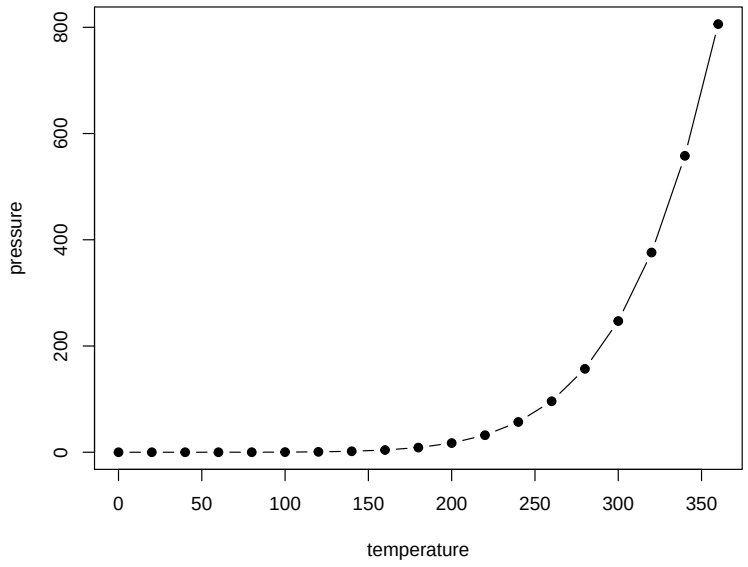


Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2020) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

Chapter 3

Data wrangling

TBC

Chapter 4

Data variety

4.1 Introduction

4.1.1 Overview

In this practical we look at data variety in environmental sciences. We start with the data from one eddy covariance measurement site in Switzerland which we had already encountered in Chapter 1, and complement it with a different type of data (remote sensing data). Specifically, we download remote sensing data that measures vegetation greenness data, quantified by the normalised difference vegetation index, NDVI. We are interested in this data for example, because we expect ecosystem photosynthesis (gross primary production), measured at the eddy covariance tower, to covary with NDVI. Such covariation can provide powerful information for modelling. For example, we can train a machine learning model with the observed covariation between measured GPP and NDVI at eddy covariance sites and use that model to scale GPP up in space (NDVI is available for the whole globe, while GPP can only be measured locally.) You will actually be doing this kind of modelling in session 13. For now, we focus on working with the spatial aspects of the data (since remote sensing data is inherently spatial) and introduce you to the toolset to work with spatial, in particular, geo-spatial data.

To get you familiar with spatial data types we will use a variety of data from freely accessible sites. We can investigate abiotic and biotic conditions in these locations. We'll be plotting the Eddy towers you have already encountered in the first two chapters and extracting values for these locations from other data. By doing this we can show that the climate at these sites spans a gradient temperature and landcover. There are also biotic components that could explain variation in productivity among sites, including species composition and functional traits of plants.

4.1.2 Learning objectives

After this learning unit, you will be able to …

- explain the possible sources of data in environmental sciences;
- understand the range of operations applied to data from basic to complex;
- read various types of data in R and apply basic operations;
- understand the structure of spatial data, including rasters and shapefiles;
- apply a range of operations to prepare data for analyses.

4.1.3 Key points of the lecture

Data operations range in complexity, here they are listed from simplest to extremely complex:

- Statistical operations
- Similarity metrics
- Ordinations
- Clustering
- Classifications
- Regressions
- Neural network deep learning

Examples of methods for the computing differences between datasets:

- *Euclidean distance* is used for the simple quantification of variation
- *PCA* is a valuable tool to explore variation and reduce dimension
- *Hamming distance* helps measure the difference between strings
- *Jaccard index* describes the similarity between two or more binary data sets

Machine learning is loosely defined as any learning done by a computer. It can be divided into…

- *Unsupervised learning*: without explanatory data, which uses clustering to obtain a categorical output
- *Supervised learning*: with explanatory data, which uses either classification also resulting in a categorical output, or regression when a numerical output is required

Data errors are divided into errors of accuracy and precision. If data is inaccurate, it strays from the true value. If data is imprecise, individual data points are variable under the same conditions. Systematic errors and inaccurate data is much harder to correct!

Errors can arise throughout the data science workflow.

- *Inherent error* refers to the error present in the source documents and data.

- *Operational error* is created after data collection.

Data can be cleaned from errors by looking for:

- Non-uniform data
- Missing data
- Duplicated data
- Outliers
- Measurement errors

Quantifying uncertainty is extremely important, as undetectable errors will persist even after data cleaning. Examples of uncertainty quantification include…

- Confidence interval (CI)
- Prediction interval

Error propagation is used for datasets containing different uncertainties to determine the overall uncertainty.

4.2 Tutorial

4.2.1 Overview

In this practical, you will be downloading and getting to know spatial data. There are three types of spatial data:

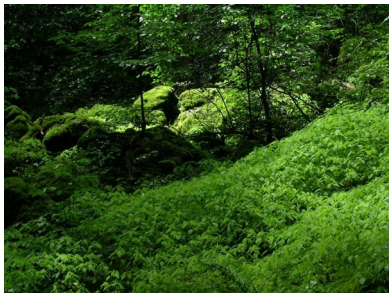
- points;
- shapefiles;
- rasters.

The first part of this practical is mandatory and will start with a little introduction on downloading remote sensing data for our tower sites and seeing what kinds of information we can gather from this. Then we will get into the nitty-gritty aspects of the afore mentioned spatial data types.

The second part is the bonus part, there we will discuss trait data and the principal component analysis (PCA), which is a method to reduce the dimensions of datasets.

4.2.2 MODIS remote download

We'll be starting off this tutorial with an example on using remote sensing data. We'll focus on a Fluxnet site already familiar to you: CH-Lae. The Lägern site is located on a mountain in the Swiss Plateau NW of Zürich and is surrounded by managed mixed deciduous mountain forests. The forest is highly diverse and dominated by beech. In the following figures, you can get an impression of the tower and its surroundings.



As always we start by loading `tidyverse`and our fluxnet site data.

```
library(tidyverse)
df_sites <- read_csv("./data/fluxnet_site_info_reduced.csv")
```

Since for this section we will only be working with the Lägern site we can directly extract its latitude and longitude.

```
lon_lae <- df_sites %>%
  filter(site == "CH-Lae") %>%
  pull(lon)

lat_lae <- df_sites %>%
  filter(site == "CH-Lae") %>%
  pull(lat)
```

In this first analysis, we want to complement temporal data measured by the Fluxnet tower with remote sensing data. *Remote sensing* is the detection or monitoring of physical characteristics of the Earth's surface by measuring reflected and emitted radiation at a distance. Data is collected in the form of images by special cameras typically from aircrafts or satellites. With this method, a huge amount of global data at large spatial scales gets easily accessible and therefore represents a useful method for the monitoring of ecosystems. However, this remotely acquired data requires validation from ground measurements, which can be done for example using the Eddy towers from the previous practicals.

One of these satellite remote sensing instruments is MODIS (Moderate Resolution Imaging Spectroradiometer). Terra MODIS and Aqua MODIS are viewing the entire Earth's surface every 1 to 2 days, acquiring data in 36 spectral bands, or groups of wavelengths.

You used to have to visit an website to be able to download the data file by file, which was a long and tedious process. Luckily, now we have access to tools that make it much more straightforward to download remote sensing data from satellites. It provides Atmosphere, Ocean, Cryosphere and Land data series. The `MODISTools` package provides a programmatic interface to the MODIS Land Products Subsets web services allows for easy downloads of "MODIS" time series (of single pixels or small regions of interest) directly to your R workspace or your computer.

- Using the `mt_...()` functions of the MODISTools we can check which products are available in MODIS, how the product we are searching for is called, and what temporal and spatial resolutions are available.
- `t_products()` lists all available products from MODIS with their temporal and spatial resolution. Products are parent categories of variables measured by the satellites, such as vegetation indices.
- `mt_bands()` requires the entry of a product chosen from "products" and list the available data variables that are represented by the different actually measured wavelengths.

- Finally, `mt_dates()` lists all dates of which data from the chosen product and band are available.

We will make use of this by loading NDVI data around our towers directly from MODIS.

```
library(MODISTools)
products <- mt_products() %>% as_tibble()
bands <- mt_bands(product = "MOD13Q1") %>% as_tibble()
dates <- mt_dates(product = "MOD13Q1", lat = lat_lae, lon = lon_lae) %>% as_tibble()
```

Now, we'll get the NDVI data for 6 years for a square of 2km x 2km around the tower site. We use the function `mt_subset()` for this. The parameters of this function download a subset of data within the chosen product (here 'MOD13Q1'). We specify the location as latitude and longitude, and the band we chose, which is at 250m spatial resolution and 16-day temporal resolution. The time period can be chosen as a subset of "dates" and must be provided in the form of start and end date. We will be looking at the beginning of 2009 to the end of 2014. `km_lr` and `km_ab` define the kilometers to the left and right of the location and kilometers above and below respectively. Since these values are being rounded to the nearest integer, we chose this minimum of 1, corresponding to 2km2. Sitename is used in writing data to file, internal gives the command whether the data should be returned as an internal structure. The progress setting defines whether the download progress should be shown or not.

```
df_ndvi <- mt_subset(product = "MOD13Q1",            # the chosen product
                     lat = lat_lae,                  # desired lat/lon
                     lon = lon_lae,
                     band = "250m_16_days_NDVI",     # chosen band defining spatial an
                     start = "2009-01-01",           # start date: 1st Jan 2009
                     end = "2014-12-19",             # end date: 19th Dec 2014
                     km_lr = 1,                      # kilometers left & right of the
                     km_ab = 1,                      # kilometers above and below the
                     site_name = "CH-Lae",           # the site name we want to give t
                     internal = TRUE,
                     progress = FALSE
                     ) %>%
  as_tibble()
```

```
head(df_ndvi)
```

```
## # A tibble: 6 x 21
##   xllcorner yllcorner cellsize nrows ncols band  units scale latitude longitude
##   <chr>     <chr>     <chr>    <int> <int> <chr> <chr> <chr>    <dbl>     <dbl>
## 1 627557.11 5278290.~ 231.656~     9     9 250m~ NDVI~ 0.00~     47.5      8.36
## 2 627557.11 5278290.~ 231.656~     9     9 250m~ NDVI~ 0.00~     47.5      8.36
## 3 627557.11 5278290.~ 231.656~     9     9 250m~ NDVI~ 0.00~     47.5      8.36
## 4 627557.11 5278290.~ 231.656~     9     9 250m~ NDVI~ 0.00~     47.5      8.36
```

```
## 5 627557.11 5278290.~ 231.656~     9     9 250m~ NDVI~ 0.00~     47.5      8.36
## 6 627557.11 5278290.~ 231.656~     9     9 250m~ NDVI~ 0.00~     47.5      8.36
## # ... with 11 more variables: site <chr>, product <chr>, start <chr>,
## #   end <chr>, complete <lgl>, modis_date <chr>, calendar_date <chr>,
## #   tile <chr>, proc_date <chr>, pixel <int>, value <int>
```

This dataframe takes some effort to make sense of. Let's make our lives a little easier and put this data into a useful spatial context. For that it needs to be converted to a raster. This is a key spatial data type and will be described in more detail in the section Raster below. MODISTools provides a handy integrated function for converting the data to a raster `mt_to_raster()`.

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj =
## prefer_proj): Discarded ellps unknown in Proj4 definition: +proj=sinu +lon_0=0
## +x_0=0 +y_0=0 +R=6371007.181 +units=m +no_defs +type=crs
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj =
## prefer_proj): Discarded datum unknown in Proj4 definition
```

```
## Warning in showSRID(SRS_string, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded ellps unknown in Proj4 definition: +proj=sinu +lon_0=0
## +x_0=0 +y_0=0 +R=6371007.181 +units=m +no_defs +type=crs
```

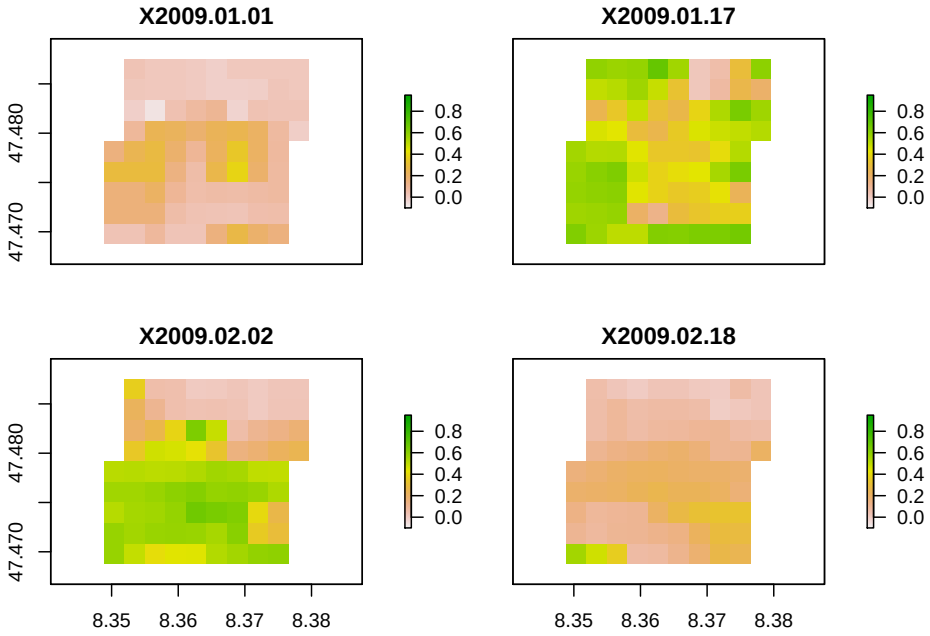
```
## Warning in showSRID(SRS_string, format = "PROJ", multiline = "NO", prefer_proj =
## prefer_proj): Discarded datum unknown in Proj4 definition
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj =
## prefer_proj): Discarded ellps unknown in Proj4 definition: +proj=sinu +lon_0=0
## +x_0=0 +y_0=0 +R=6371007.181 +units=m +no_defs +type=crs
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj =
## prefer_proj): Discarded datum unknown in Proj4 definition
```

By plotting the data we can verify that we loaded a square of NDVI data around our tower for every 16-day time step. We plot four images in the winter season.

```
plot(raster_ndvi[[1:4]], zlim=c(-0.1, 0.95))
```



Just out of interest let's calculate the minimum and maximum of these plots.

```
min_winter <- min(minValue(raster_ndvi[[1:4]]))
max_winter <- max(maxValue(raster_ndvi[[1:4]]))
paste0("Minimum Winter:", sep=" ", min_winter)
```

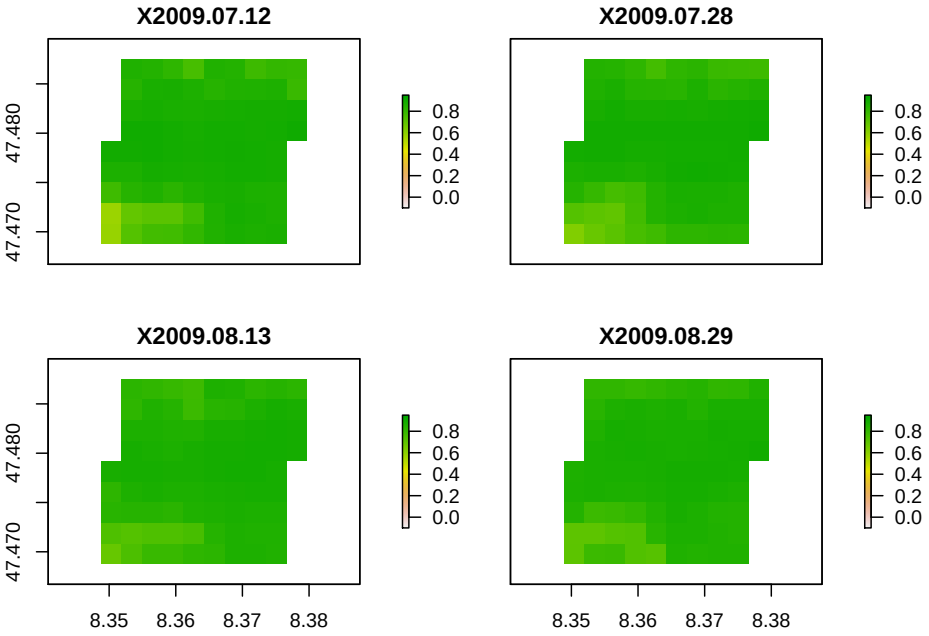
```
## [1] "Minimum Winter: -0.0679177447153291"
```

```
paste0("Maximum Winter:", sep=" ", max_winter)
```

```
## [1] "Maximum Winter: 0.683717115657303"
```

We do the same for the summer season.

```
plot(raster_ndvi[[13:16]], zlim=c(-0.1, 0.95))
```



We compute minimum and maximum.

```
min_summer <- min(minValue(raster_ndvi[[13:16]]))
max_summer <- max(maxValue(raster_ndvi[[13:16]]))
paste0("Minimum Summer:", sep=" ", min_summer)
```

```
## [1] "Minimum Summer: 0.569610980491112"
```

```
paste0("Maximum Summer:", sep=" ", max_summer)
```

```
## [1] "Maximum Summer: 0.903467898529256"
```

From the two plots, we can clearly distinguish summer and winter time steps due to the differences in NDVI. Summer NDVI is much higher since our tower site is located in the Northern hemisphere where leaves get lost during winter for many vegetation types. Heterogeneity in winter is higher which might be due to conifers that don't lose their leaves.

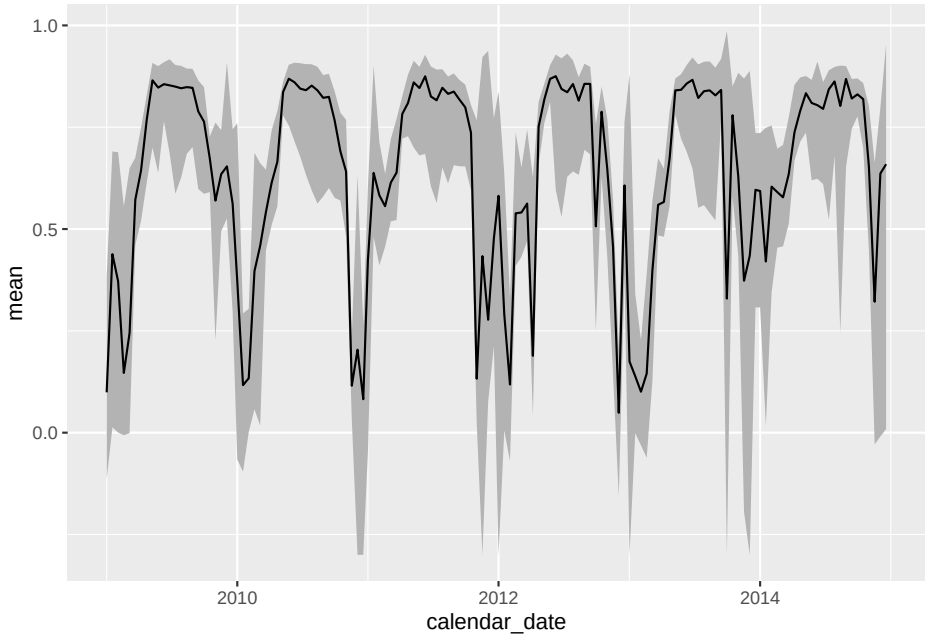
However, we are also interested in the mean across all (spatial) pixels for each date. To clarify this means we collapse the pixel dimension into a single mean value for each date. This is can done using `group_by()` in combination with `summarise()`, which were explained in chapter 2.

```
library(lubridate) # lubridate is loaded to work with dates

## first determine the scaling factor that is to be applied to the NDVI values. This has practica
scale_factor <- bands %>%
  filter(band == "250m_16_days_NDVI") %>%
```

```
  pull(scale_factor) %>%
  as.numeric()

df_ndvi_spatialmean <- df_ndvi %>%
  mutate(calendar_date = ymd(calendar_date)) %>%  # make the dates into comprehensible
  group_by(calendar_date) %>%                     # group the data by day
  summarise(mean = mean(value), min = min(value), max = max(value)) %>%   # calculate
  mutate(mean = mean * scale_factor, min = min * scale_factor, max = max * scale_factor
```

We can now plot our NDVI time series across our years (2009 to 2014) using the means we calculated above.

```
df_ndvi_spatialmean %>%
  ggplot(aes(x = calendar_date)) +
  geom_ribbon(aes(ymin = min, ymax = max), fill = "grey70") +
  geom_line(aes(y = mean))
```



After having aggregated the data to the mean NDVI across all 81 pixels for each date, we have reduced the dimensions of the data frame to only one (time). It now has a structure that can easily be combined with the time series data from the eddy covariance tower (each site is a point in space).

```
# this can now be combined to the data frame with flux data that also has dates along
ddf_ch_lae <- read_csv("./data/ddf_ch_lae.csv")
```

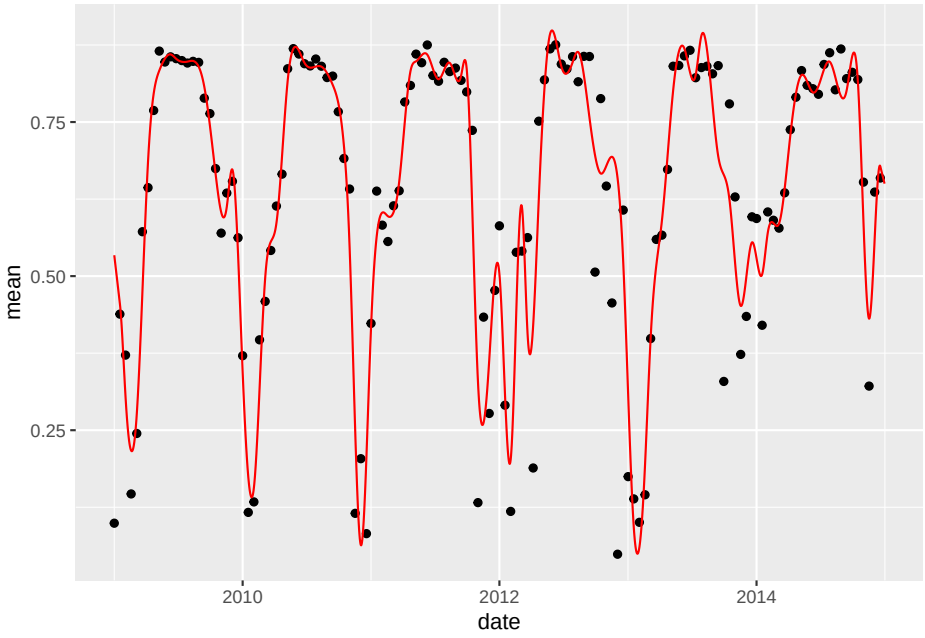
```
# now we can combine the two data frames along dates.
ddf_ch_lae_ndvi <- ddf_ch_lae %>%
  dplyr::rename(date = TIMESTAMP) %>%
  dplyr::mutate(year = year(date)) %>%
  dplyr::filter(year %in% 2009:2014) %>%  # NDVI data was downloaded only for these years
  left_join(df_ndvi_spatialmean %>% rename(date = calendar_date), by = "date") %>%
  dplyr::select(-year)
```

NDVI data is provided every 16 days. In order to predict values at other time steps, we can fit a cubic smoothing spline to our daily values. (Cubic smoothing splines embody a curve fitting technique that blends the ideas of cubic splines and curvature minimization to create an effective data modeling tool for noisy data. Traditional interpolating cubic splines represent the tabulated data as a piece-wise continuous curve which passes through each value in the data table. The curve spanning each data interval is represented by a cubic polynomial, with the requirement that the endpoints of adjacent cubic polynomials match in location and in their first and second derivatives).

```
ddf_ch_lae_ndvi <- ddf_ch_lae_ndvi %>%
  mutate(date_dec = decimal_date(date))

df_nona <- ddf_ch_lae_ndvi %>%
  drop_na()

out_spline <- smooth.spline( df_nona$date_dec, df_nona$mean, spar=0.1 )
vals_spline <- predict( out_spline, ddf_ch_lae_ndvi$date_dec )$y

ddf_ch_lae_ndvi <- ddf_ch_lae_ndvi %>%
  mutate(ndvi_splined = vals_spline)
```

Let's take a look...

```
ddf_ch_lae_ndvi %>%
  ggplot() +
  geom_point(aes(x = date, y = mean)) +
  geom_line(aes(x = date, y = ndvi_splined), color = "red")
```



To see how well this data correlates with measurements from the fluxnet tower, we look at the relationship of GPP vs. NDVI*PPFD.

```
ddf_ch_lae_ndvi <- ddf_ch_lae_ndvi %>%
  mutate(ppfd_abs = ndvi_splined * PPFD_IN)
```

Does NDVI affect the relationship between GPP and PPFD?

```
linmod1 <- lm(GPP_NT_VUT_REF ~ PPFD_IN, data = ddf_ch_lae_ndvi)
summary(linmod1)
```

```
##
## Call:
## lm(formula = GPP_NT_VUT_REF ~ PPFD_IN, data = ddf_ch_lae_ndvi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3749  -2.0413  -0.0137   1.9913  14.8576
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.9348433  0.1097112   8.521   <2e-16 ***
## PPFD_IN     0.0148270  0.0003038  48.797   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.06 on 2173 degrees of freedom
```

```
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.5229, Adjusted R-squared:  0.5226
## F-statistic:  2381 on 1 and 2173 DF,  p-value: < 2.2e-16
```

```
linmod2 <- lm(GPP_NT_VUT_REF ~ ppfd_abs, data = ddf_ch_lae_ndvi)
summary(linmod2)
```

```
##
## Call:
## lm(formula = GPP_NT_VUT_REF ~ ppfd_abs, data = ddf_ch_lae_ndvi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9202 -1.8307 -0.0829  1.7236 14.4273
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.3395678  0.0903180   14.83   <2e-16 ***
## ppfd_abs    0.0180868  0.0003148   57.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.792 on 2173 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.603,  Adjusted R-squared:  0.6028
## F-statistic:  3301 on 1 and 2173 DF,  p-value: < 2.2e-16
```

Indeed, R2 increased from 0.5229 to 0.603 when including considering NDVI * PPFD, instead of just PPFD. In other words, it also matters whether leaves are actually out and green for modelling the relationship between GPP and PPFD. NDVI * PPFD approximates the amount of *absorbed* incoming light (not just the incoming light) and is therefore closer to being a physiologically relevant quantity.

4.2.3 Points on the globe

Documenting the coordinates of data as it is collected allows us to assign a precise location to that data, which can be important in further analysis when, for example, comparing location. As you all undoubtedly know coordinates come with a latitude (defining the position N or S of the equator) and longitude (defining the position E or W of the meridian). With the coordinates we can already plot our data points, remembering that the latitude is equivalent to the y-axis and longitude to the x-axis.

In this section, we look at the position of the Fluxnet sites. The Fluxnet network measures land-atmosphere exchanges of greenhouse gases and energy for

sites across the globe using the eddy covariance technique. High-frequency measurements of vertical wind velocity and a scalar mixing ratio (CO2, H2O, temperature, etc.) provides estimates of the net exchange of the scalar. Eddy covariance is currently the standard method to measure fluxes of trace gases between ecosystems and the atmosphere (sources: https://fluxnet.org/about/, https://www.nature.com/articles/s41597-020-0534-3.pdf).

Right at the beginning of this tutorial we loaded the dataset containing information on sites in Europe. We treat site locations as a geographical position. So we will extract the name, longitude and latitude of our sites.

```
head(df_sites)
```

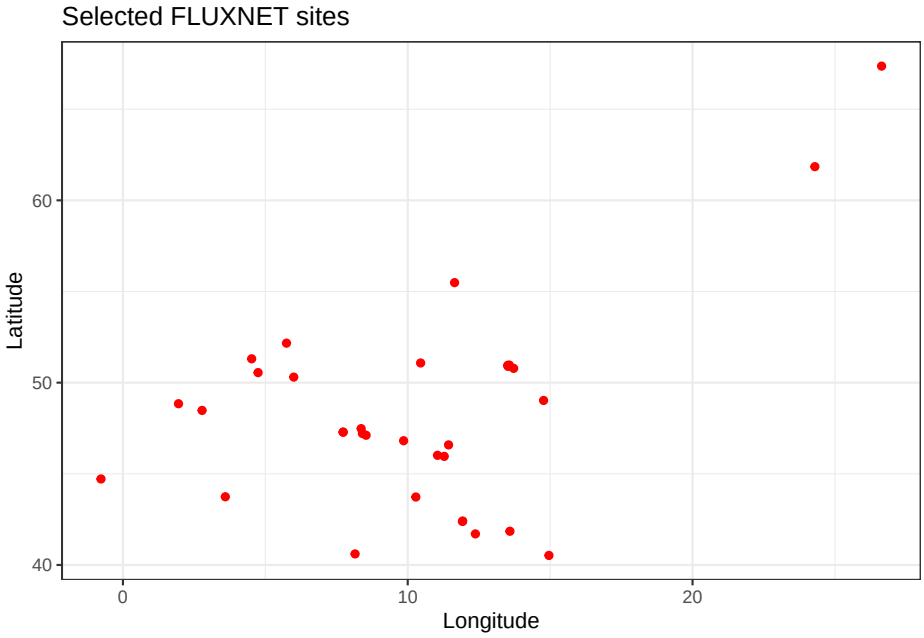
```
## # A tibble: 6 x 3
##   site     lon   lat
##   <chr>  <dbl> <dbl>
## 1 BE-Bra  4.52  51.3
## 2 BE-Lon  4.75  50.6
## 3 BE-Vie  6.00  50.3
## 4 CH-Cha  8.41  47.2
## 5 CH-Fru  8.54  47.1
## 6 CH-Lae  8.36  47.5
```

Our dataset *'df_sites'* contains two sites in Greenland and Siberia. We'll focus on a smaller spatial domain, restricted to "mainland" Europe, therefore, we exclude sites RU-Cok and DK-ZaH.

```
df_sites <- df_sites %>%
  filter(!(site %in% c("RU-Cok", "DK-ZaH")))
```

Already now we can plot our data points without any further adjustments.

```
ggplot() +
  geom_point(data = df_sites, aes(x = lon, y = lat), color = "red") +
  labs(title = "Selected FLUXNET sites", x = "Longitude", y = "Latitude")+
  theme_bw()
```



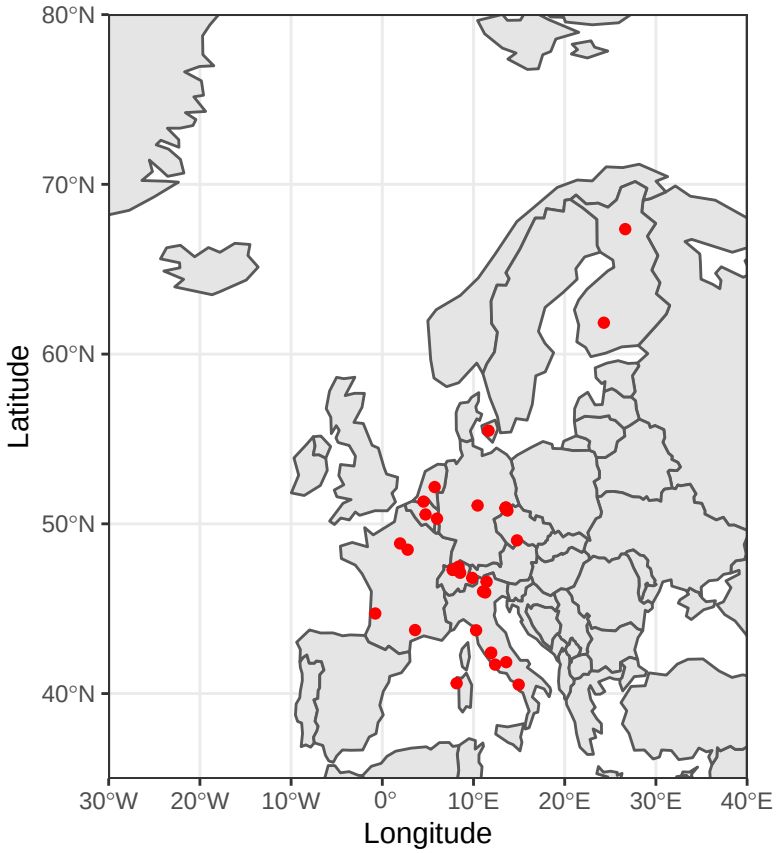
We can make an educated guess where the points are but this plot doesn't tell us much by itself without a map as a reference.

So to visualize the location of our points we plot them on a map. We use the package `SpData` to get the world map and crop it to Europe.

```
library(sf)
library(spData)
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
ggplot() +
  geom_sf(data = world) +
  geom_point(data = df_sites, aes(x = lon, y = lat), color = "red") +
  labs(x = "Longitude", y = "Latitude") +
  coord_sf(xlim = c(-30,40), ylim = c(35,80), expand = FALSE) +
  theme_bw()
```




```
# coord_sf(crs = "+proj=robin")
```

Of course, we can carry on using the data points as they are (for plotting or analysis) but for more detailed spatial analysis we will turn them into *SpatialPoints*.

SpatialPoints consist of a matrix with n rows and 2 columns, one for each coordinate (latitude & longitude). n is the number of points in the data. These points also have a so-called *'projection string'* or *'crs'* indicating the coordinate reference system in which coordinates of points are expressed. There are different ways (formulas) to project the earth (an ellipsoid) onto a 2-dimensional map. You can only perform calculations, if the same method (projection, coordinate reference system (crs)) is used. Having points in different coordinate reference systems can cause data to look very skewed. It's good to know which coordinate reference system your data is in, the most commonly used one is **WGS84 (EPSG: 4326)**. There are some useful resources to convert help you find the correct spatial reference.

To transform points into *SpatialPoints* we use the function `SpatialPoints` in the r package `sp`.

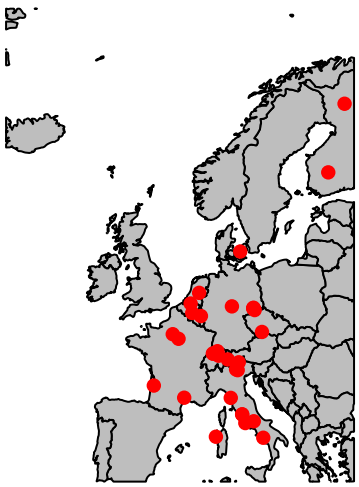
```
library(sp)
sp_sites <- SpatialPoints(dplyr::select(df_sites, -site)) # remove character column
```

Below you can see how with the R-base call `plot()` you can plot the *SpatialPoints* directly. But we'll have to also provide a map to plot them onto. The loaded shapefile of europe will also be used in the next part called Shapefiles.

```
library(rgdal)
europe_shape <- readOGR(dsn="./data/shapefiles", layer="europe_map")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/pascalschneider/Polybox/Shared/Data Science Lecture Planning - shared folder/4
## with 53 features
## It has 94 fields
```

```
plot(europe_shape, col="grey")
plot(sp_sites, col = "red", add = TRUE, pch = 16)
```



4.2.4 Shapefiles

Above we looked at points on the globe and transformed our data to *SpatialPoints*. This was already the first example of what forms a shapefile can come in. Shapefiles basically store geographic information, meaning location and any additional information, in shape objects. Shape objects in R are defined by *SpatialPoints*, *SpatialLines*, and *SpatialPolygons* classes of the `sp` package. The corresponding *SpatialPointsDataFrame*, *SpatialLinesDataFrame*, and *SpatialPolygonsDataFrame* classes allow storing shape objects together with a dataframe. The number of rows in the dataframe corresponds to the number of points (number of rows in coordinate matrix), lines (size of the list of Lines), or polygons (size of the list of Polygons). This allows us to associate a vector of variables (a

row of the dataframe) to each shape object. In the next step, we load a shapefile containing the countries of the world provided by Natural Earth, as a reference with which we can contextualize our spatial data. Since we only are looking at towers in Europe, we have cropped the raster to Europe for you.

```
library(rgdal)
europe_shape <- readOGR(dsn="./data/shapefiles", layer="europe_map")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/pascalschneider/Polybox/Shared/Data Science Lecture Planning - share
## with 53 features
## It has 94 fields
```

Let's see what class *europe_shape* is:

```
class(europe_shape)
```

```
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
```

Above we mentioned that *SpatialPolygonsDataFrame* contains both a list of *SpatialPolygons* and a dataframe with information on each polygon. Here's how you can access the information in the dataframe. We'll just display the header for now.

```
head(europe_shape@data)
```

```
##        featurecla scalerank LABELRANK     SOVEREIGNT SOV_A3 ADM0_DIF LEVEL
## 0 Admin-0 country         6         6        Vatican    VAT        0     2
## 1 Admin-0 country         4         6 United Kingdom    GB1        1     2
## 2 Admin-0 country         4         6 United Kingdom    GB1        1     2
## 3 Admin-0 country         3         6 United Kingdom    GB1        1     2
## 4 Admin-0 country         1         2 United Kingdom    GB1        1     2
## 5 Admin-0 country         1         3        Ukraine    UKR        0     2
##                TYPE          ADMIN ADM0_A3 GEOU_DIF        GEOUNIT GU_A3 SU_DIF
## 0 Sovereign country        Vatican     VAT        0        Vatican   VAT      0
## 1           Country         Jersey     JEY        0         Jersey   JEY      0
## 2           Country       Guernsey     GGY        0       Guernsey   GGY      0
## 3           Country    Isle of Man     IMN        0    Isle of Man   IMN      0
## 4           Country United Kingdom     GBR        0 United Kingdom   GBR      0
## 5 Sovereign country        Ukraine     UKR        0        Ukraine   UKR      0
##          SUBUNIT SU_A3 BRK_DIFF           NAME      NAME_LONG BRK_A3
## 0        Vatican   VAT        0        Vatican        Vatican    VAT
## 1         Jersey   JEY        0         Jersey         Jersey    JEY
## 2       Guernsey   GGY        0       Guernsey       Guernsey    GGY
## 3    Isle of Man   IMN        0    Isle of Man    Isle of Man    IMN
## 4 United Kingdom   GBR        0 United Kingdom United Kingdom    GBR
## 5        Ukraine   UKR        0        Ukraine        Ukraine    UKR
```

```
##         BRK_NAME       BRK_GROUP ABBREV POSTAL
## 0        Vatican            <NA>   Vat.      V
## 1         Jersey Channel Islands   Jey.     JE
## 2       Guernsey Channel Islands Guern.     GG
## 3    Isle of Man            <NA>  IoMan     IM
## 4 United Kingdom            <NA>   U.K.     GB
## 5        Ukraine            <NA>   Ukr.     UA
##                                           FORMAL_EN FORMAL_FR
## 0                           State of the Vatican City      <NA>
## 1                                 Bailiwick of Jersey      <NA>
## 2                              Bailiwick of Guernsey      <NA>
## 3                                               <NA>      <NA>
## 4 United Kingdom of Great Britain and Northern Ireland      <NA>
## 5                                            Ukraine      <NA>
##                NAME_CIAWF             NOTE_ADM0 NOTE_BRK          NAME_SORT
## 0 Holy See (Vatican City)                  <NA>     <NA> Vatican (Holy See)
## 1                  Jersey U.K. crown dependency     <NA>             Jersey
## 2                Guernsey U.K. crown dependency     <NA>           Guernsey
## 3             Isle of Man U.K. crown dependency     <NA>        Isle of Man
## 4          United Kingdom                  <NA>     <NA>     United Kingdom
## 5                 Ukraine                  <NA>     <NA>            Ukraine
##   NAME_ALT MAPCOLOR7 MAPCOLOR8 MAPCOLOR9 MAPCOLOR13  POP_EST POP_RANK
## 0 Holy See         1         3         4          2     1000        3
## 1     <NA>         6         6         6          3    98840        8
## 2     <NA>         6         6         6          3    66502        8
## 3     <NA>         6         6         6          3    88815        8
## 4     <NA>         6         6         6          3 64769452       16
## 5     <NA>         5         1         6          3 44033874       15
##   GDP_MD_EST POP_YEAR LASTCENSUS GDP_YEAR                    ECONOMY
## 0          0     2015        -99        0 2. Developed region: nonG7
## 1       5080     2017       2001     2015 2. Developed region: nonG7
## 2       3465     2017       2001     2015 2. Developed region: nonG7
## 3       7428     2017       2006     2014 2. Developed region: nonG7
## 4    2788000     2017       2011     2016    1. Developed region: G7
## 5     352600     2017       2001     2016       6. Developing region
##                INCOME_GRP WIKIPEDIA FIPS_10_ ISO_A2 ISO_A3 ISO_A3_EH ISO_N3
## 0 2. High income: nonOECD         0       VT     VA    VAT       VAT    336
## 1 2. High income: nonOECD       -99       JE     JE    JEY       JEY    832
## 2 2. High income: nonOECD       -99       GK     GG    GGY       GGY    831
## 3 2. High income: nonOECD       -99       IM     IM    IMN       IMN    833
## 4    1. High income: OECD       -99       UK     GB    GBR       GBR    826
## 5  4. Lower middle income       -99       UP     UA    UKR       UKR    804
##   UN_A3 WB_A2 WB_A3   WOE_ID WOE_ID_EH
## 0   336   -99   -99 23424986  23424986
## 1   832    JG   CHI 23424857  23424857
## 2   831    JG   CHI 23424827  23424827
```

```
## 3   833    IM   IMY 23424847  23424847
## 4   826    GB   GBR      -90  23424975
## 5   804    UA   UKR 23424976  23424976
##
## 0
## 1
## 2
## 3
## 4 Eh ID includes Channel Islands and Isle of Man. UK constituent countries of Englar
## 5
##   ADM0_A3_IS ADM0_A3_US ADM0_A3_UN ADM0_A3_WB CONTINENT REGION_UN
## 0        VAT        VAT        -99        -99    Europe    Europe
## 1        JEY        JEY        -99        -99    Europe    Europe
## 2        GGY        GGY        -99        -99    Europe    Europe
## 3        IMN        IMN        -99        -99    Europe    Europe
## 4        GBR        GBR        -99        -99    Europe    Europe
## 5        UKR        UKR        -99        -99    Europe    Europe
##         SUBREGION             REGION_WB NAME_LEN LONG_LEN ABBREV_LEN TINY
## 0 Southern Europe Europe & Central Asia        7        7          4    4
## 1 Northern Europe Europe & Central Asia        6        6          4  -99
## 2 Northern Europe Europe & Central Asia        8        8          6  -99
## 3 Northern Europe Europe & Central Asia       11       11          5  -99
## 4 Northern Europe Europe & Central Asia       14       14          4  -99
## 5  Eastern Europe Europe & Central Asia        7        7          4  -99
##   HOMEPART MIN_ZOOM MIN_LABEL MAX_LABEL      NE_ID WIKIDATAID         NAME_AR
## 0        1        0       5.0      10.0 1159321407       Q237       ?????????
## 1      -99        0       5.0      10.0 1159320725       Q785           ?????
## 2      -99        0       5.0      10.0 1159320715     Q25230          ??????
## 3      -99        0       5.0      10.0 1159320721      Q9676       ????? ???
## 4        1        0       1.7       6.7 1159320713       Q145 ??????? ???????
## 5        1        0       3.0       7.0 1159321345       Q212        ????????
##          NAME_BN                NAME_DE        NAME_EN             NAME_ES
## 0 ????????? ????           Vatikanstadt   Vatican City Ciudad del Vaticano
## 1         ??????                 Jersey         Jersey              Jersey
## 2           <NA>                Guernsey       Guernsey            Guernsey
## 3   ??? ?? ?????            Isle of Man    Isle of Man         Isla de Man
## 4     ?????????? Vereinigtes Königreich United Kingdom         Reino Unido
## 5        ???????                Ukraine        Ukraine             Ucrania
##       NAME_FR          NAME_EL         NAME_HI             NAME_HU
## 0     Vatican         ????????      ?????? ???             Vatikán
## 1      Jersey          ???????           ?????              Jersey
## 2   Guernesey         ????????       ????????? Guernsey Bailiffség
## 3  île de Man    ????? ??? ???  ?????? ?? ????                 Man
## 4 Royaume-Uni ??????? ???????? ???????? ??????  Egyesült Királyság
## 5     Ukraine         ????????         ???????             Ukrajna
##         NAME_ID            NAME_IT NAME_JA NAME_KO             NAME_NL
```

```
## 0       Vatikan Città del Vaticano    ????  ??? ??        Vaticaanstad
## 1        Jersey  Baliato di Jersey   ?????    ?? ?              Jersey
## 2      Guernsey           Guernsey   ?????    ?? ?            Guernsey
## 3     Pulau Man       Isola di Man     ???     ? ?                 Man
## 4 Britania Raya        Regno Unito    ????      ?? Verenigd Koninkrijk
## 5       Ukraina            Ucraina   ?????   ?????            Oekraïne
##           NAME_PL     NAME_PT        NAME_RU        NAME_SV          NAME_TR
## 0         Watykan    Vaticano        ???????  Vatikanstaten          Vatikan
## 1          Jersey      Jersey         ??????         Jersey           Jersey
## 2        Guernsey    Guernsey         ??????       Guernsey         Guernsey
## 3       Wyspa Man Ilha de Man     ?????? ???    Isle of Man        Man Adas?
## 4 Wielka Brytania Reino Unido ????????????? Storbritannien Birle?ik Krall?k
## 5         Ukraina     Ucrânia        ???????        Ukraina          Ukrayna
##                                   NAME_VI NAME_ZH
## 0                           Thành Vatican     ???
## 1                                  Jersey     ???
## 2                                Guernsey     ???
## 3                                 ??o Man     ???
## 4 V??ng qu?c Liên hi?p Anh và B?c Ireland      ??
## 5                                 Ukraina     ???
```

Let's first plot our towers on this map of Europe. Remember, how when you transformed the tower locations we mentioned each point has a designated coordinate reference system? Here, we need to extract this projection from *europe_shape* using `proj4string()` and add it to our *SpatialPoints*. This way we can make sure they are in the same reference system for plotting them together.

```
geo.proj <- sp::proj4string(europe_shape)
```

```
## Warning in sp::proj4string(europe_shape): CRS object has comment, which is lost
## in output
```

```
pts <- sp::SpatialPoints(sp_sites, proj4string = sp::CRS(geo.proj))
```

Then we can take the *SpatialPoints*, now in the correct coordinate reference system, the data from *europe_shape* and bind them to our tower sites in the dataframe *df_sites*.

```
df_sites_country_info <- sp::over(pts, europe_shape) %>%
    as_tibble() %>%
    bind_cols(df_sites, .)

head(df_sites_country_info)
```

```
## # A tibble: 6 x 97
##   site    lon   lat featurecla scalerank LABELRANK SOVEREIGNT SOV_A3 ADM0_DIF
##   <chr> <dbl> <dbl> <chr>          <int>     <int> <chr>      <chr>     <int>
## 1 BE-B~  4.52  51.3 Admin-0 c~         1         2 Belgium    BEL           0
```

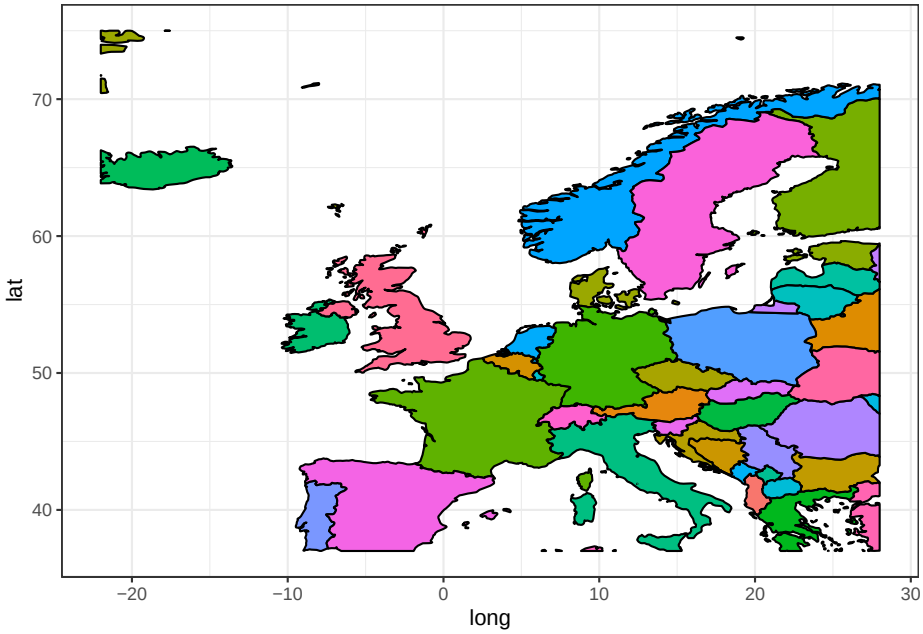
```
## 2 BE-L~  4.75  50.6 Admin-0 c~         1         2 Belgium    BEL           0
## 3 BE-V~  6.00  50.3 Admin-0 c~         1         2 Belgium    BEL           0
## 4 CH-C~  8.41  47.2 Admin-0 c~         1         4 Switzerla~ CHE           0
## 5 CH-F~  8.54  47.1 Admin-0 c~         1         4 Switzerla~ CHE           0
## 6 CH-L~  8.36  47.5 Admin-0 c~         1         4 Switzerla~ CHE           0
## # ... with 88 more variables: LEVEL <int>, TYPE <chr>, ADMIN <chr>,
## #   ADM0_A3 <chr>, GEOU_DIF <int>, GEOUNIT <chr>, GU_A3 <chr>, SU_DIF <int>,
## #   SUBUNIT <chr>, SU_A3 <chr>, BRK_DIFF <int>, NAME <chr>, NAME_LONG <chr>,
## #   BRK_A3 <chr>, BRK_NAME <chr>, BRK_GROUP <chr>, ABBREV <chr>, POSTAL <chr>,
## #   FORMAL_EN <chr>, FORMAL_FR <chr>, NAME_CIAWF <chr>, NOTE_ADM0 <chr>,
## #   NOTE_BRK <chr>, NAME_SORT <chr>, NAME_ALT <chr>, MAPCOLOR7 <int>,
## #   MAPCOLOR8 <int>, MAPCOLOR9 <int>, MAPCOLOR13 <int>, POP_EST <chr>,
## #   POP_RANK <int>, GDP_MD_EST <dbl>, POP_YEAR <int>, LASTCENSUS <int>,
## #   GDP_YEAR <int>, ECONOMY <chr>, INCOME_GRP <chr>, WIKIPEDIA <int>,
## #   FIPS_10_ <chr>, ISO_A2 <chr>, ISO_A3 <chr>, ISO_A3_EH <chr>, ISO_N3 <chr>,
## #   UN_A3 <chr>, WB_A2 <chr>, WB_A3 <chr>, WOE_ID <int>, WOE_ID_EH <int>,
## #   WOE_NOTE <chr>, ADM0_A3_IS <chr>, ADM0_A3_US <chr>, ADM0_A3_UN <int>,
## #   ADM0_A3_WB <int>, CONTINENT <chr>, REGION_UN <chr>, SUBREGION <chr>,
## #   REGION_WB <chr>, NAME_LEN <int>, LONG_LEN <int>, ABBREV_LEN <int>,
## #   TINY <int>, HOMEPART <int>, MIN_ZOOM <dbl>, MIN_LABEL <dbl>,
## #   MAX_LABEL <dbl>, NE_ID <chr>, WIKIDATAID <chr>, NAME_AR <chr>,
## #   NAME_BN <chr>, NAME_DE <chr>, NAME_EN <chr>, NAME_ES <chr>, NAME_FR <chr>,
## #   NAME_EL <chr>, NAME_HI <chr>, NAME_HU <chr>, NAME_ID <chr>, NAME_IT <chr>,
## #   NAME_JA <chr>, NAME_KO <chr>, NAME_NL <chr>, NAME_PL <chr>, NAME_PT <chr>,
## #   NAME_RU <chr>, NAME_SV <chr>, NAME_TR <chr>, NAME_VI <chr>, NAME_ZH <chr>
```

For the next part we will be using a map European with country boarders. For this, we'll make an object called *shp_df*, this we do using the column *SOVEREIGNT*, which we saw when looking at `europe_shape@data`. This preserves only the data we need for further analysis and plotting.

```
library(maptools)

shp_df <- broom::tidy(europe_shape, region = "SOVEREIGNT")
head(shp_df)
```

```
## # A tibble: 6 x 7
##    long   lat order hole  piece group     id
##   <dbl> <dbl> <int> <lgl> <fct> <fct>     <chr>
## 1  19.3  41.9     1 FALSE 1     Albania.1 Albania
## 2  19.3  41.9     2 FALSE 1     Albania.1 Albania
## 3  19.4  42.0     3 FALSE 1     Albania.1 Albania
## 4  19.4  42.0     4 FALSE 1     Albania.1 Albania
## 5  19.4  42.1     5 FALSE 1     Albania.1 Albania
## 6  19.3  42.1     6 FALSE 1     Albania.1 Albania
```

```
ggplot() +
  geom_polygon(data = shp_df, aes(x = long, y = lat, group = group, fill = id), colour = "black") +
  theme_bw() +
  theme(legend.position = "none")
```



As we saw in the earlier plot of the tower locations only some countries in Europe contain towers. Therefore, we want to crop our map to include only those countries.

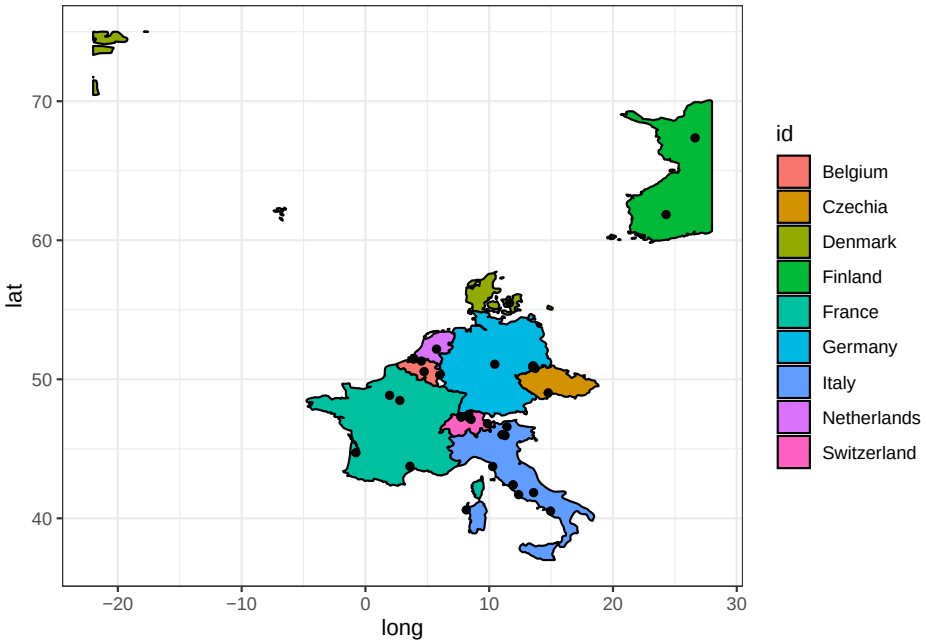
To do this we first make a vector which is a list of the country each tower is located in. We get this from *df_site_country_info*. Then we filter out only those countries with a tower in them and voilà, plot the countries and tower locations as a result.

```
countries_with_site <- df_sites_country_info %>% pull(SOVEREIGNT) %>% as.character()

# resulting character vector (without NAs)
countries_with_site %>% na.omit()
```

```
##  [1] "Belgium"     "Belgium"     "Belgium"     "Switzerland" "Switzerland"
##  [6] "Switzerland" "Switzerland" "Switzerland" "Switzerland" "Germany"
## [11] "Germany"     "Germany"     "Germany"     "Germany"     "Denmark"
## [16] "Finland"     "Finland"     "France"      "France"      "France"
## [21] "France"      "Italy"       "Italy"       "Italy"       "Italy"
## [26] "Italy"       "Italy"       "Italy"       "Italy"       "Italy"
## [31] "Netherlands" "Czechia"
```

```
## attr(,"na.action")
## [1] 33
## attr(,"class")
## [1] "omit"
```

```
# country with towers filtered
shp_df_sub <- shp_df %>%
  filter(id %in% countries_with_site)

# plot result
ggplot() +
  geom_polygon(data = shp_df_sub, aes(x = long, y = lat, group = group, fill = id), col
  geom_point(data = df_sites, aes(x = lon, y = lat)) +
  theme_bw()
```



Checkpoint

You just learnt how to plot *europe_shape* coloured by country ("SOVREIGNT"), now plot it coloured by region in europe. Start by finding the column that gives you this information in `europe_shape@data`.

Solution

```
# take a look at the column names
names(europe_shape@data)
```

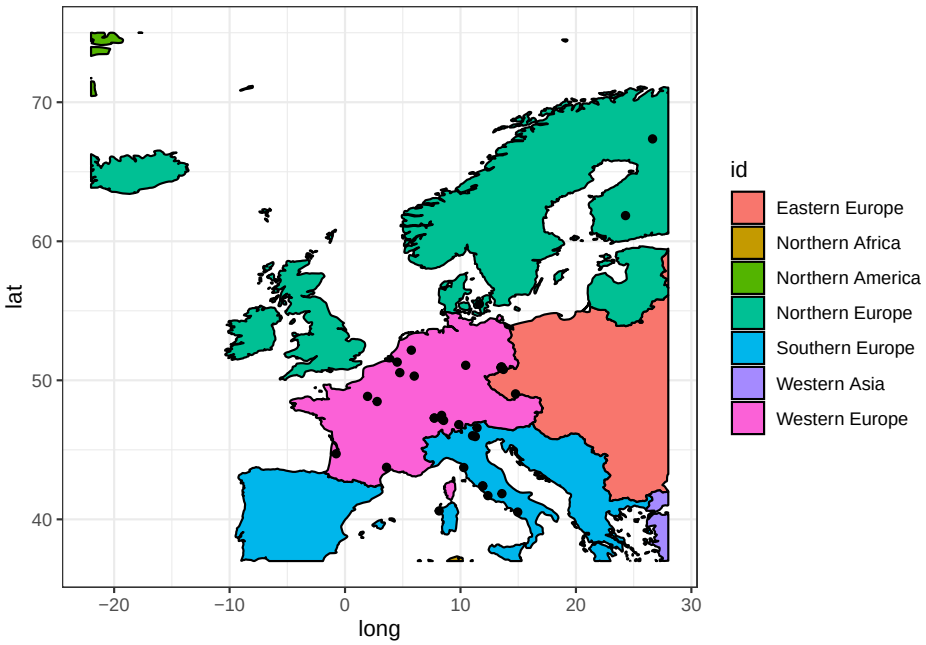
```
##  [1] "featurecla" "scalerank"  "LABELRANK"  "SOVEREIGNT" "SOV_A3"
##  [6] "ADM0_DIF"   "LEVEL"      "TYPE"       "ADMIN"      "ADM0_A3"
```

```
## [11] "GEOU_DIF"   "GEOUNIT"    "GU_A3"      "SU_DIF"     "SUBUNIT"
## [16] "SU_A3"      "BRK_DIFF"   "NAME"       "NAME_LONG"  "BRK_A3"
## [21] "BRK_NAME"   "BRK_GROUP"  "ABBREV"     "POSTAL"     "FORMAL_EN"
## [26] "FORMAL_FR"  "NAME_CIAWF" "NOTE_ADM0"  "NOTE_BRK"   "NAME_SORT"
## [31] "NAME_ALT"   "MAPCOLOR7"  "MAPCOLOR8"  "MAPCOLOR9"  "MAPCOLOR13"
## [36] "POP_EST"    "POP_RANK"   "GDP_MD_EST" "POP_YEAR"   "LASTCENSUS"
## [41] "GDP_YEAR"   "ECONOMY"    "INCOME_GRP" "WIKIPEDIA"  "FIPS_10_"
## [46] "ISO_A2"     "ISO_A3"     "ISO_A3_EH"  "ISO_N3"     "UN_A3"
## [51] "WB_A2"      "WB_A3"      "WOE_ID"     "WOE_ID_EH"  "WOE_NOTE"
## [56] "ADM0_A3_IS" "ADM0_A3_US" "ADM0_A3_UN" "ADM0_A3_WB" "CONTINENT"
## [61] "REGION_UN"  "SUBREGION"  "REGION_WB"  "NAME_LEN"   "LONG_LEN"
## [66] "ABBREV_LEN" "TINY"       "HOMEPART"   "MIN_ZOOM"   "MIN_LABEL"
## [71] "MAX_LABEL"  "NE_ID"      "WIKIDATAID" "NAME_AR"    "NAME_BN"
## [76] "NAME_DE"    "NAME_EN"    "NAME_ES"    "NAME_FR"    "NAME_EL"
## [81] "NAME_HI"    "NAME_HU"    "NAME_ID"    "NAME_IT"    "NAME_JA"
## [86] "NAME_KO"    "NAME_NL"    "NAME_PL"    "NAME_PT"    "NAME_RU"
## [91] "NAME_SV"    "NAME_TR"    "NAME_VI"    "NAME_ZH"
```

```
# make a df of the data needed from europe_shape
shp_df_region <- broom::tidy(europe_shape, region = "SUBREGION")
head(shp_df_region)
```

```
## # A tibble: 6 x 7
##    long   lat order hole  piece group           id
##   <dbl> <dbl> <int> <lgl> <fct> <fct>           <chr>
## 1  28    42.0     1 FALSE 1     Eastern Europe.1 Eastern Europe
## 2  27.9  42.0     2 FALSE 1     Eastern Europe.1 Eastern Europe
## 3  27.8  42.0     3 FALSE 1     Eastern Europe.1 Eastern Europe
## 4  27.8  42.0     4 FALSE 1     Eastern Europe.1 Eastern Europe
## 5  27.7  42.0     5 FALSE 1     Eastern Europe.1 Eastern Europe
## 6  27.7  42.0     6 FALSE 1     Eastern Europe.1 Eastern Europe
```

```
# plot europe coloured by region
ggplot() +
  geom_polygon(data = shp_df_region, aes(x = long, y = lat, group = group, fill = id), colour = '
  geom_point(data = df_sites, aes(x = lon, y = lat)) +
  theme_bw()
```



4.2.5 Rasters

Next up are rasters. A raster is a matrix or grid of cells each of which contains information in the form of a value.

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 CRS, projection, resolution, etc.

Let's take a look at some rasters and get familiar with some functions to analyse them.

Since our goal ultimate goal is to predict productivity, we will consider different factors that might explain productivity, for example, landcover and temperature. To start off we'll look at some landcover data. With this data, we can investigate the surroundings of the Fluxnet towers. This data gives us the different classes of physical coverage of the Earth's surface, such as forests, grasslands, croplands, lakes, etc. Landcover types affect fluxes measured by the tower through the vegetation that characterizes them, or by influencing the radiation budget through albedo, or water availability. Although we have some information from the overview file of the towers, we can still check whether we can confirm this

information. We will load landcover data from GlobCover provided by the European space agency (ESA). This project provides landcover maps observations of surface reflectance from the 300m MERIS sensor on board the ENVISAT satellite mission as their input.

```
# library(raster) -> we have already done this but remember you need this to load rasters!

raster_landcover <- raster("./data/Globcover_EU.tif")
```

This landcover data consists of different categories, these GlobCover categories are based on the Land Cover Classification System (LCCS) which was developed by the Food and Agricultural Organization of the United Nations (FAO) to provide a consistent framework for the classification of mapping land cover. They include various types of forest, shrub- and grasslands, cropland and artificial surfaces.

Before we look at the data, we first reduce the spatial extent that the raster covers. We crop it to a rectangle around the site location of CH-Lae (+/- 2 degrees in longitudinal direction, +/- 1 degree in latitudinal direction).

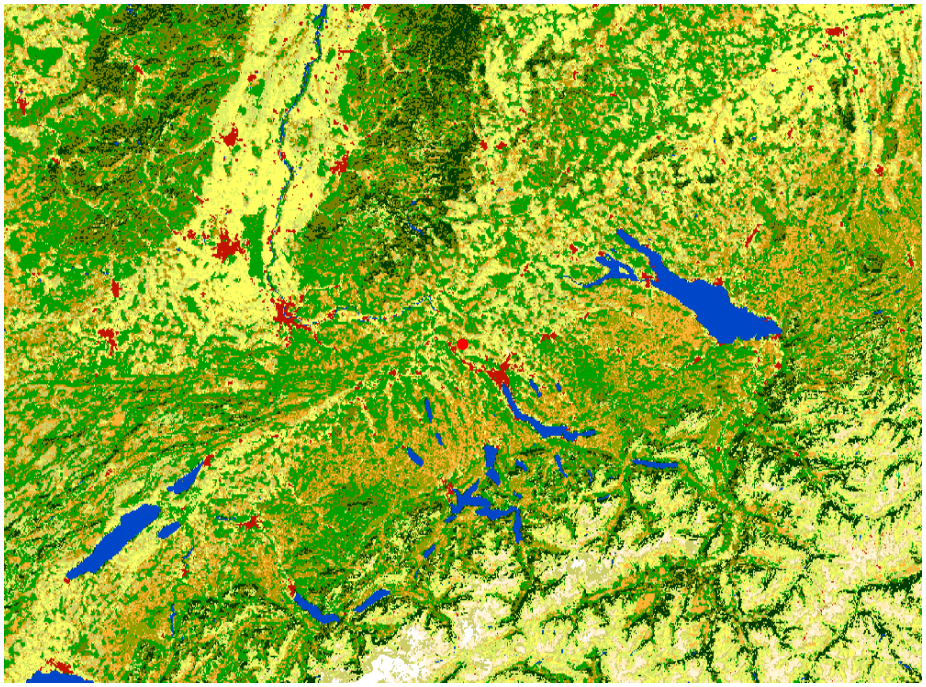
```
bounding_box <- extent(lon_lae-2, lon_lae+2, lat_lae-1, lat_lae+1)
raster_landcover_crop <- crop(raster_landcover, bounding_box)
```

We can plot our raster as the region around our tower. To indicate the position of our tower, we add a red dot.

```
plot(raster_landcover_crop, legend=FALSE, xlab="longitude", ylab="latitude")
```

```
## Warning in plot.window(...): "legend" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "legend" is not a graphical parameter

## Warning in title(...): "legend" is not a graphical parameter

## Warning in graphics::rasterImage(z, bb[1], bb[3], bb[2], bb[4], interpolate =
## interpolate, : "legend" is not a graphical parameter
```

```
points(lon_lae, lat_lae, pch = 16, col = "red")
```



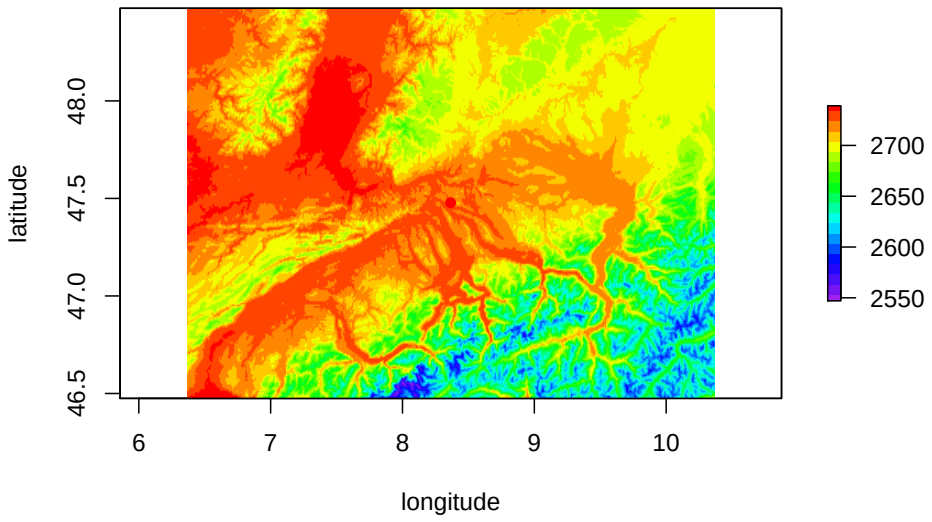
If we want to conduct analysis using two different rasters, we have to make sure they have the same resolution and are aligned on the same grid. Let's read a second raster and compare the grids. We load spatio-temporal temperature rasters available from CHELSA to complement our flux tower measurements for further analysis. The temperature raster data from CHELSA provides free high-resolution climate data (temperature and precipitation) for the past and the future. The past data we use here are downscaled model output temperature and precipitation estimates of the ERA-Interim climatic reanalysis (forecast models and data assimilation systems reanalysing archived observations). While the temperature algorithm is based on statistical downscaling of atmospheric temperatures, the precipitation algorithm incorporates orographic predictors including wind fields, valley exposition, and boundary layer height, with subsequent bias correction. The data we use consists of a monthly temperature and precipitation time series over the area of Europe from 2006 to 2012 in January. To compare this new raster with the previous one we need to crop it to the same

region as the tower.

```
raster_chelsa <- raster("./data/Chelsa_t_mean_2006-2012.tif")
raster_chelsa_crop <- crop(raster_chelsa, bounding_box)
```

We now plot our raster.

```
pal <- colorRampPalette(c("purple","blue","cyan","green","yellow","red"))
plot(raster_chelsa_crop, col = pal(20), xlab="longitude", ylab="latitude")
points(lon_lae, lat_lae, pch = 16, col = "red")
```

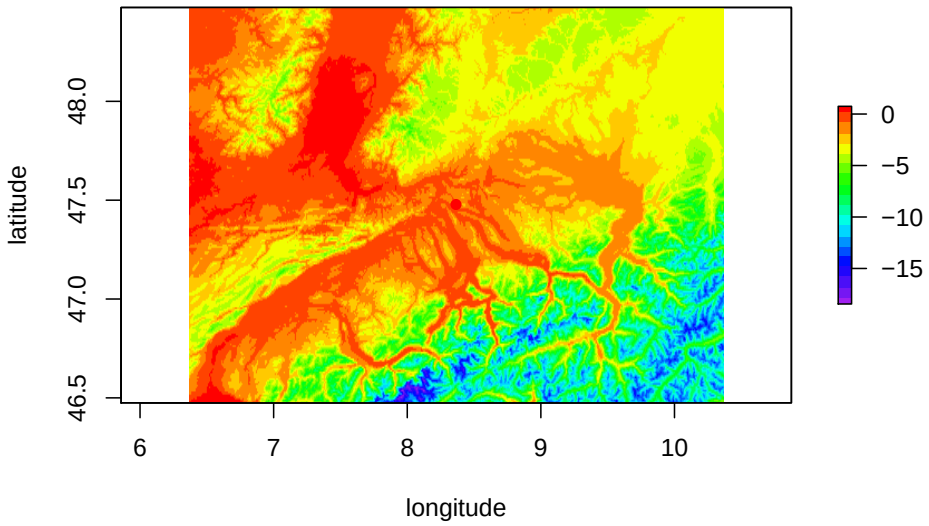


We see that the temperature units are not in a comprehensive format. To change the unit from Kelvin x10 to degree Celsius we have to divide the data by 10 and subtract 273.15.

```
raster_chelsa_crop <- raster_chelsa_crop/10-273.15
```

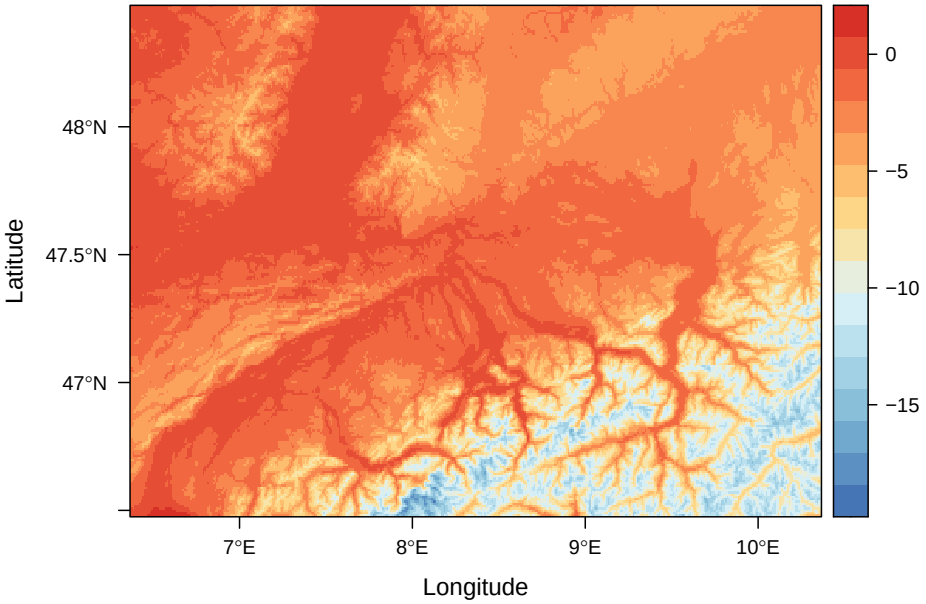
We plot our raster again.

```
pal <- colorRampPalette(c("purple","blue","cyan","green","yellow","red"))
plot(raster_chelsa_crop, col = pal(20), xlab="longitude", ylab="latitude")
points(lon_lae, lat_lae, pch = 16, col = "red")
```



We have seen some of the basic functions in the package `raster`. Now we will load one more package to work with raster files. We use the package `rasterVis`. This package contains methods for enhanced visualization and interaction with raster data. It implements visualization methods for quantitative data and categorical data, both for univariate and multivariate rasters. It also provides methods to display spatiotemporal rasters, and vector fields.

```
library(rasterVis)
library(RColorBrewer)

mapTheme <- rasterTheme(region = rev(brewer.pal(8,"RdYlBu")))
plt <- levelplot(raster_chelsa_crop, margin=FALSE, par.settings = mapTheme)
plt
```



Checkpoint

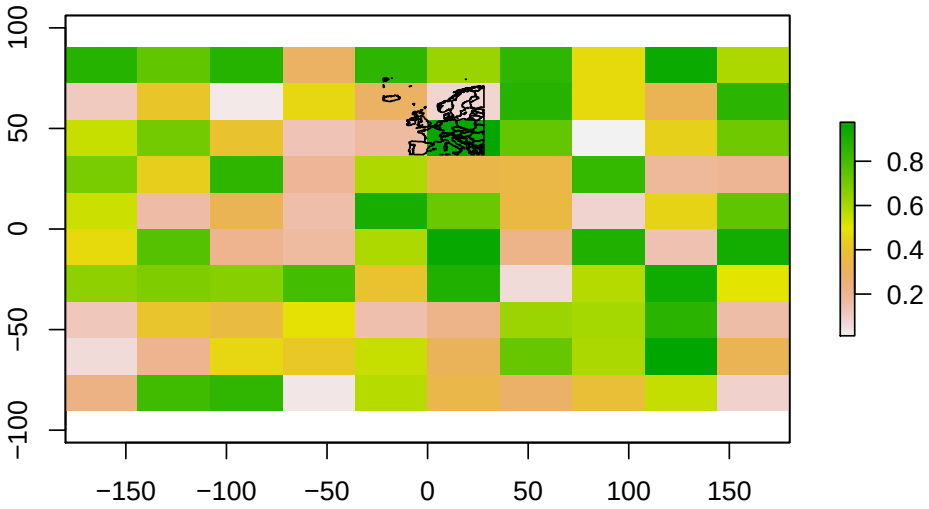
Create your own small raster with 100 grid cells. Give it values, a projection (e.g. '+proj=utm +zone=48 +datum=WGS84') and plot it.

Bonus: If you chose the example projection provided, add *europe_shape* to your plot.

Solution

```
# create a raster
# short compact version
myraster <- raster(ncol=10, nrow=10)
# OR
# longer version
myraster <- raster()
ncol(myraster) <- 10
nrow(myraster) <- 10

# add values to the raster, this can be done in many ways here are two examples:
values(myraster) <- 1:ncell(myraster)
# OR
values(myraster) <- runif(ncell(myraster))

# add a projection
projection(myraster) <- "+proj=utm +zone=48 +datum=WGS84"

# plot it
```

```
plot(myraster)
# for fun let's see where europe would be on our raster, since it has a projection...
plot(europe_shape, add=TRUE)
```



4.2.5.1 Aggregating

A **raster grid** is uniquely defined by an *origin*, a point that one of the intersections of grid lines, and its resolution, the magnitude of each cell-side. In the R package `raster`, the origin is defined as the point closest to (0,0) that is still an intersection of grid lines. The functions `origin()` and `res()` from the R package `raster` return the origin and resolution of a raster object. So let's see what the resolution and origin of the rasters we loaded before are.

```
res(raster_landcover)
```

```
## [1] 0.002777778 0.002777778
```

```
res(raster_chelsa)
```

```
## [1] 0.008333333 0.008333333
```

```
origin(raster_landcover)
```

```
## [1]  0.001388889 -0.001388889
```

```
origin(raster_chelsa)
```

```
## [1] -0.000139609 -0.000139249
```

Landcover has a much higher resolution. Let's re-grid the Landcover raster to match the grid of the Chelsa raster. Aggregating means resampling an input

raster to a coarser resolution based on a specified aggregation strategy. We now determine the aggregation factor by dividing the resolution of the landcover raster by that of the Chelsa raster. We do it both for the longitude and latitude to make sure they match.

```
res_landcover <- res(raster_landcover)
res_chelsa <- res(raster_chelsa)
factor_agg_lon <- res_chelsa[1] / res_landcover[1]
factor_agg_lat <- res_chelsa[2] / res_landcover[2]
print(factor_agg_lon); print(factor_agg_lat)
```

```
## [1] 3
```

```
## [1] 3
```

First, we observed that the land cover raster has a higher resolution than the temperature raster, now we now it is 3 times higher. So we need to transform the landcover raster to make sure it has the same grid (same *origin* and *resolution*) as the temperature raster. This will allow us to combine them in a potential analysis. We achieve this using `aggregate()` from the R package `raster` to obtain a sum of edges raster at 1000m x 1000m resolution.

The aggregate function takes three main arguments. * ***x*** : the raster object to aggregate, * ***fact*** : aggregation factor, i.e. the number of current cells that will make up the side of one of the new cells, and, * ***fun*** : the function to apply to summarize the raster values of the fact^2 (in this case 3^2=9) cells.

We aggregate the landcover raster at 1km using the aggregation factor calculated above.

```
raster_landcover_crop_agg <- aggregate(raster_landcover_crop, fact = factor_agg_lon, fun = modal)
```

We check the resolutions and origin of the two rasters and plot our the new landcover one.

```
res(raster_landcover_crop_agg)
```

```
## [1] 0.008333333 0.008333333
```

```
res(raster_chelsa)
```

```
## [1] 0.008333333 0.008333333
```

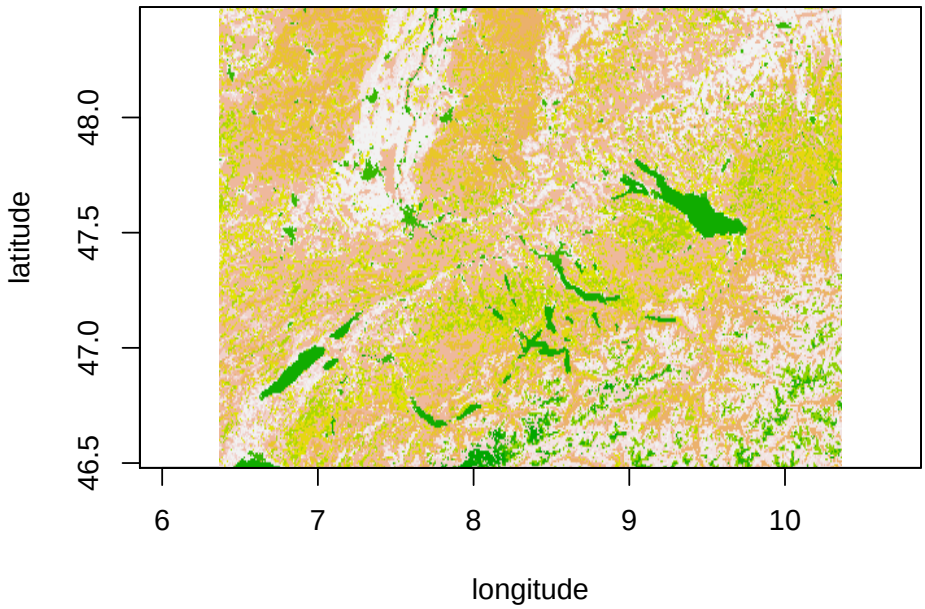
```
origin(raster_landcover_crop_agg)
```

```
## [1] -0.001388889 -0.004166667
```

```
origin(raster_chelsa)
```

```
## [1] -0.000139609 -0.000139249
```

```
plot(raster_landcover_crop_agg, legend=FALSE, xlab="longitude", ylab="latitude")
```



We can see that the resolutions are now the same but the origin still differs. To get matching origins we must do another step: align the rasters.

4.2.5.2 Aligning

Aligning rasters is done using the function `resample()`. There are two main resampling methods: *Nearest Neighbour* or *Bilinear Interpolation*, which method is used depends upon the input data and its use after the operation is performed.

Nearest Neighbour is best used for categorical data like land-use classification or slope classification. The values that go into the grid stay exactly the same, a 2 comes out as a 2, and 99 comes out as 99. The value of the output cell is determined by the nearest cell center on the input grid. Nearest Neighbor can be used on continuous data but the results can be blocky.

Bilinear Interpolation uses a weighted average of the four nearest cell centers. The closer an input cell center is to the output cell center, the higher the influence of its value is on the output cell value. This means that the output value could be different than the nearest input, but is always within the same range of values as the input. Since the values can change, Bilinear is not recommended for categorical data. Instead, it should be used for continuous data like elevation and raw slope values.

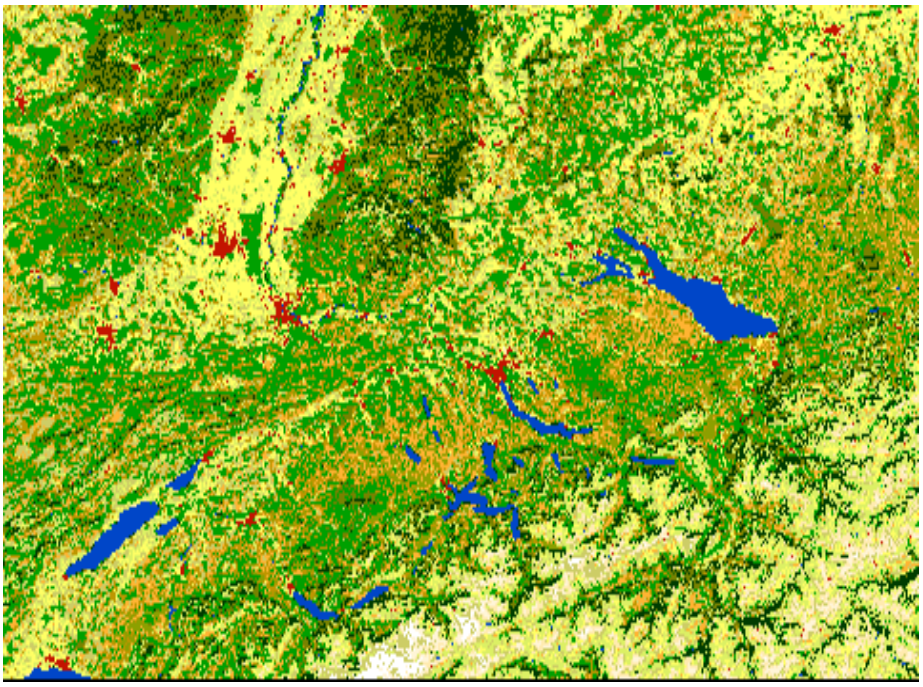
```
raster_landcover_crop_resampl <- resample(raster_landcover_crop, raster_chelsa_crop, method="ngb"
plot(raster_landcover_crop_resampl, legend=FALSE, xlab="longitude", ylab="latitude")
```

```
## Warning in plot.window(...): "legend" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "legend" is not a graphical parameter
```

```
## Warning in title(...): "legend" is not a graphical parameter
```

```
## Warning in graphics::rasterImage(z, bb[1], bb[3], bb[2], bb[4], interpolate =
## interpolate, : "legend" is not a graphical parameter
```



Now we check if the two rasters are aligned.

```
res(raster_landcover_crop_resampl)
```

```
## [1] 0.008333333 0.008333333
```

```
res(raster_chelsa)
```

```
## [1] 0.008333333 0.008333333
```

```
origin(raster_landcover_crop_resampl)
```

```
## [1] -0.000139609 -0.000139249
```

```
origin(raster_chelsa)
```

```
## [1] -0.000139609 -0.000139249
```

Now the two rasters are aligned.

4.2.5.3 Point extraction

In this section, we will show you how to extract data from our rasters or spatial data. We will be extracting the values for the locations of our towers. The function to extract values is aptly named `extract()`. First, we specify the raster from which the values should be extracted and then the exact sites at

which the values should be extracted. Here, we combine all this into a neat dataframe.

```
df_sites_temp <- read_csv("./data/fluxnet_site_info_reduced.csv")

df_sites_temp <- extract(raster_chelsa, sp_sites, sp = TRUE) %>%
  as_tibble() %>%
  right_join(df_sites_temp, by = c("lon", "lat")) %>%
  dplyr::rename(temp_chelsa = Chelsa_t_mean_2006.2012)
```

In the lectures, you have already, heard that we must always be aware of modelled data and be aware of its limitations. Therefore, we don't just want to use CHELSA temperature data. It is important to consider different models and compare them. Hence, we'll look at another climate model. We load and plot the provided rasters showing the mean annual temperatures for the Worldclim data for the years 2006-2012. WorldClim is a set of global climate layers (gridded climate data) with a spatial resolution of about 1 km2. We extract values from Worldclim at the same tower sites and add them to the *df_sites* data frame. Now we have a tidy dataframe with the site, latitude and longitude, temperatures for Chelsa and WorldClim at each tower location.

```
raster_worldclim <- raster("./data/WC_mean_t_2006-2012.tif")

df_sites_temp <- extract(raster_worldclim, sp_sites, sp = TRUE) %>%
  as_tibble() %>%
  right_join(df_sites_temp, by = c("lon", "lat")) %>%
  dplyr::rename(temp_wc = WC_mean_t_2006.2012)

head(df_sites_temp)
```

```
## # A tibble: 6 x 5
##   temp_wc   lon   lat temp_chelsa site
##     <dbl> <dbl> <dbl>       <dbl> <chr>
## 1  -0.633  4.52  51.3        2750 BE-Bra
## 2  -1.45   4.75  50.6        2741 BE-Lon
## 3  -3.75   6.00  50.3        2716 BE-Vie
## 4  -4.76   8.41  47.2        2730 CH-Cha
## 5  -4.97   8.54  47.1        2699 CH-Fru
## 6  -5.38   8.36  47.5        2710 CH-Lae
```

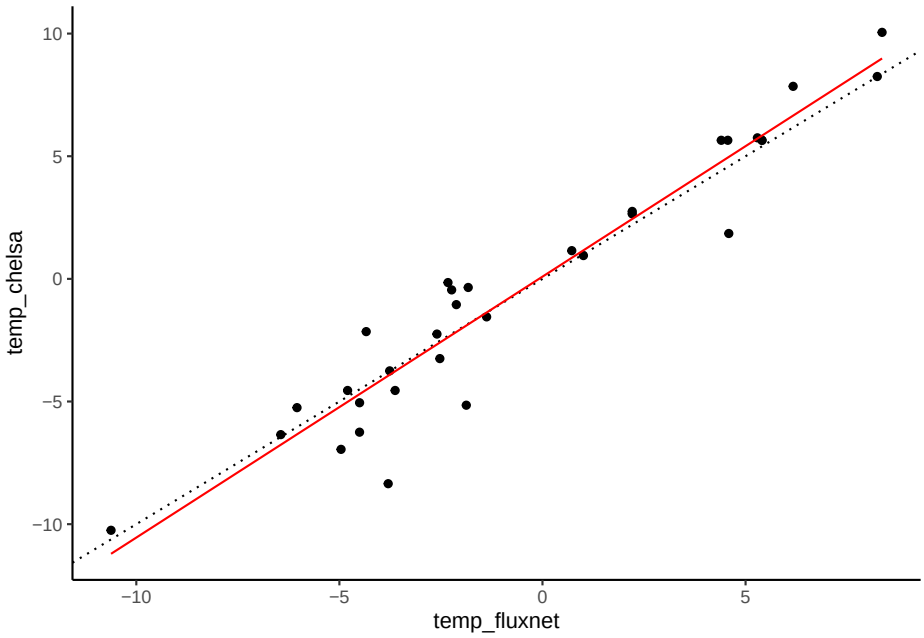
We get the monthly mean temperature of Jan 2006 from FLUXNET data.

```
library(lubridate)
load("./data/ddf_allsites_nested_joined.RData")

df_sites_temp <- ddf_allsites_nested_joined %>%
  dplyr::select(site = siteid, data) %>%
  unnest(data) %>%
```

```
  dplyr::select(site, TIMESTAMP, TA_F) %>%
  mutate(month = month(TIMESTAMP), year = year(TIMESTAMP)) %>%
  filter(month == 1, year == 2006) %>%
  group_by(site) %>%
  summarise(temp_fluxnet = mean(TA_F)) %>%

  # add temp data extracted from spatial files (chelsa and worldclim)
  left_join(df_sites_temp, dplyr::select(site, temp_wc, temp_chelsa),
            by = "site")
```

We compare the climate models by showing the correlations with the tower's data. The dotted line shows what the linear model would look like if the values of our model and the tower data overlapped. The red line shows the actual linear model of the relationship between the climate model and the tower data. The closer the red line to the dotted line the better the model is at predicting the temperatures measured by the towers.

```
df_sites_temp$temp_chelsa <- df_sites_temp$temp_chelsa/10 -273.15

# we remove the sites we removed before
df_sites_temp <- df_sites_temp[-c(16, 34), ]

df_sites_temp %>%
  ggplot(aes(x = temp_fluxnet, y = temp_chelsa), xlim=c(-10,5)) +
  geom_point() +
  geom_abline(intercept=0, slope=1, linetype="dotted") +
  geom_smooth(method='lm', color="red", size=0.5, se=FALSE) +
  theme_classic()
```

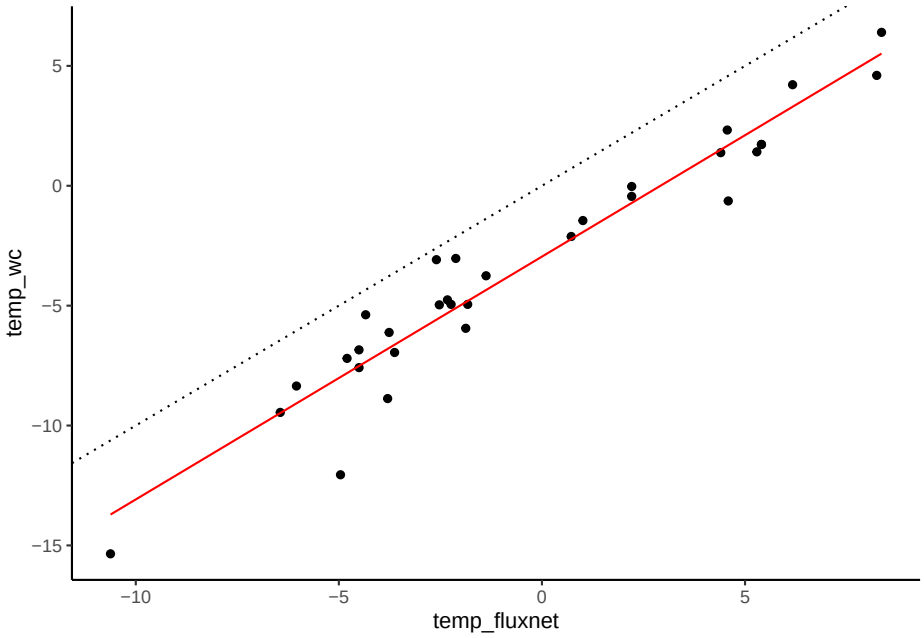


We compute the mean squared error (MSE). The MSE calculates the average of the squares of errors of all the data points from a fitted line or model. Put simply this is the difference between the data points (the actual data) and the estimated points given by the line or model. The values are squared to get positive values even if the difference is negative. Smaller values reflect less variation of the data. In a model you want to minimise the MSE, as this reflects that the model's predicted or estimated points are closer to your actual data.

```
library(Metrics)
mse(df_sites_temp$temp_fluxnet, df_sites_temp$temp_chelsa)
```

```
## [1] 2.314917
```

```
df_sites_temp %>%
  ggplot(aes(x = temp_fluxnet, y = temp_wc), xlim=c(-10,5)) +
  geom_point() +
  geom_abline(intercept=0, slope=1, linetype="dotted") +
  geom_smooth(method='lm', color="red", size=0.5, se=FALSE) +
  theme_classic()
```



We compute the mean squared error.

```
mse(df_sites_temp$temp_fluxnet, df_sites_temp$temp_wc)
```

```
## [1] 10.43716
```

We see that Chelsa correlates better (because the mean squared error is smaller) with the temperature values at the towers. We also see that Worldclim compared to CHELSA tends to predict colder temperatures.

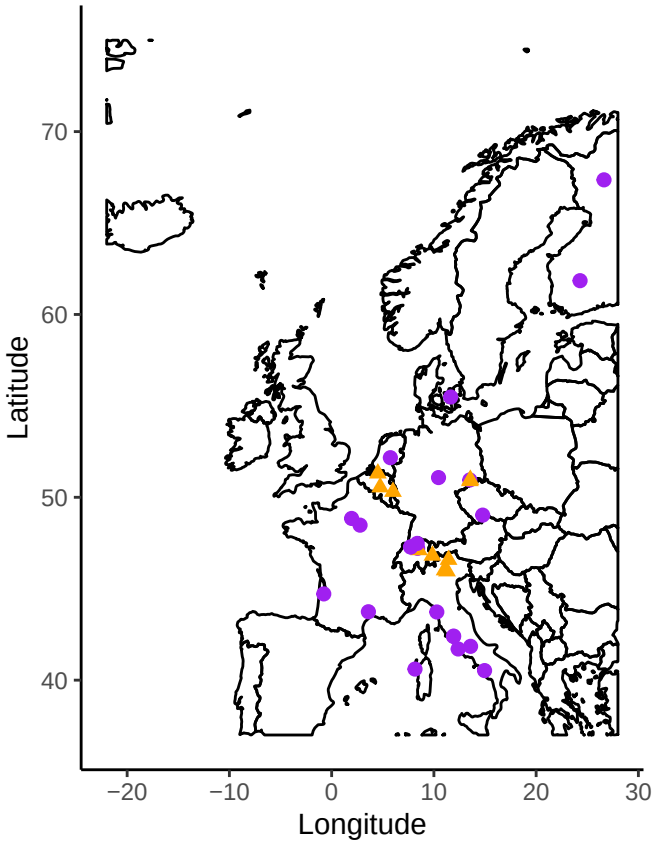
Checkpoint

Above, we compared the CHELSA temperatures with the tower temperatures. Now we want to visually see which of the two data had the higher mean for each tower location. Plot the map of europe and add the tower locations as points. Colour the points so that for a higher CHESLA mean they are one colour and a higher tower temperature another.

Solution

```
# There are more way to do this but we used the function ifelse().
# It takes 3 components: a condition, what to do if the condition applies and what to

# So here if chelsa temperatures are higher we want the point to be purple otherwise o
cols_chelsa_flux <- ifelse(df_sites_temp$temp_chelsa > df_sites_temp$temp_fluxnet,"purp

# We could also do different shapes if we wanted...
pch_chelsa_flux <- ifelse(df_sites_temp$temp_chelsa > df_sites_temp$temp_fluxnet, 19 ,1
```

```
# Plot the result:

# plot(europe_shape)
# points(df_sites_temp$lon,df_sites_temp$lat, pch=pch_diff, col = col_checkpoint)

# OR with ggplot

ggplot() +
  geom_polygon(data = europe_shape, aes(x = long, y = lat, group = group), fill=NA, colour = "bla
  geom_point(data = df_sites_temp, aes(x = lon, y = lat), color = cols_chelsa_flux, shape = pch_c
  labs(x = "Longitude", y = "Latitude") +
  theme_classic()+
  coord_quickmap()
```



4.2.5.4 Correlations with GPP

We already had a look at GPP in Chapter 2. In this section we want to correlate Chelsa temperature and landcover with GPP.

In models we often take multiple variables and correlate them to another variable (here GPP). By seeing if they correlate well we can see if any of the variables make good so-called *predictors*. Let's see how well landcover and temperature correlate with GPP.

We start off by extracting the mean GPP at the tower sites. To do this we group our data by the *siteid* (or tower) and the year. This way we can get the GPP across a year and then an average across all years for each site.

```
GPP <- ddf_allsites_nested_joined %>%
  unnest(data) %>%
  mutate(year = year(TIMESTAMP)) %>%
  group_by(siteid, year) %>%
  summarise(gpp_ann = sum(GPP_NT_VUT_REF)) %>%
  ungroup() %>%
  group_by(siteid) %>%
  summarise(gpp_meanann = mean(gpp_ann)) %>%
  dplyr::select(siteid, gpp_meanann)

# we remove the data we don't have in the other datasets for the correlations
GPP <- GPP[-c(14, 16, 34), ]
```

We then extract landcover data at the tower sites.

```
df_sites_landcover <- extract(raster_landcover, sp_sites, sp = TRUE)
df_sites_landcover <- as.tibble(df_sites_landcover)

# we remove the sites we removed before
df_sites_landcover <- df_sites_landcover[-c(16, 34), ]
```

And create a dataframe with all the extracted data from the tower sites to then make correlations.

```
landcover <- df_sites_landcover$Globcover_EU
chelsa <- df_sites_temp$temp_chelsa
df_corr <- cbind(landcover, chelsa, GPP)
df_corr <- as.data.frame(df_corr)
head(df_corr)
```

```
##   landcover chelsa siteid gpp_meanann
## 1        50   1.85 BE-Bra          NA
## 2        20   0.95 BE-Lon    1348.667
## 3        90  -1.55 BE-Vie    1792.497
## 4       120  -0.15 CH-Cha    2478.429
```

```
## 5       140  -6.95 CH-Dav    1166.543
## 6        50  -3.25 CH-Fru    2106.319
```

Though the landcover column contains numbers, it is actually categorical data. Each number represents a specific landcover type, so we add a column with the corresponding the landcover category name.

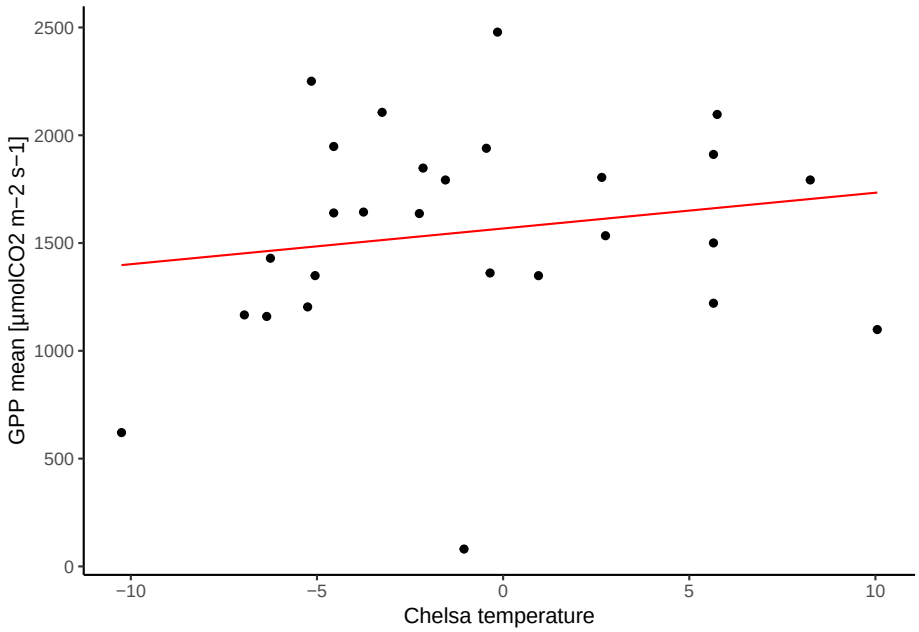
```
df_corr <- df_corr %>%
  mutate(landcover_cat = case_when(landcover == 14 ~ "cropland",
        landcover == 20 ~ "mosaic cropland",
        landcover == 50 ~ "deciduous forest",
        landcover == 70 ~ "evergreen forest",
        landcover == 90 ~ "needleleaved forest",
        landcover == 110 ~ "mosaic forest",
        landcover == 120 ~ "mosaic forest/shrub/grass",
        landcover == 130 ~ "mosaic shrubland",
        landcover == 140 ~ "mosaic grassland",
        landcover == 150 ~ "sparse vegetation"))
```

Back to correlations! Let's start by correlating the Chelsa temperatures with GPP. We do this by building a simple linear model using the function `lm()`.

```
reg1 <- lm(gpp_meanann~chelsa, data = df_corr)
```

To visualise the correlation we plot our temperature data with the linear model.

```
df_corr %>%
  ggplot(aes(x = chelsa, y = gpp_meanann)) +
  geom_point() +
  geom_smooth(method='lm', color="red", size=0.5, se=FALSE) +
  xlab("Chelsa temperature") +
  ylab("GPP mean [µmolCO2 m-2 s-1]") +
  theme_classic()
```

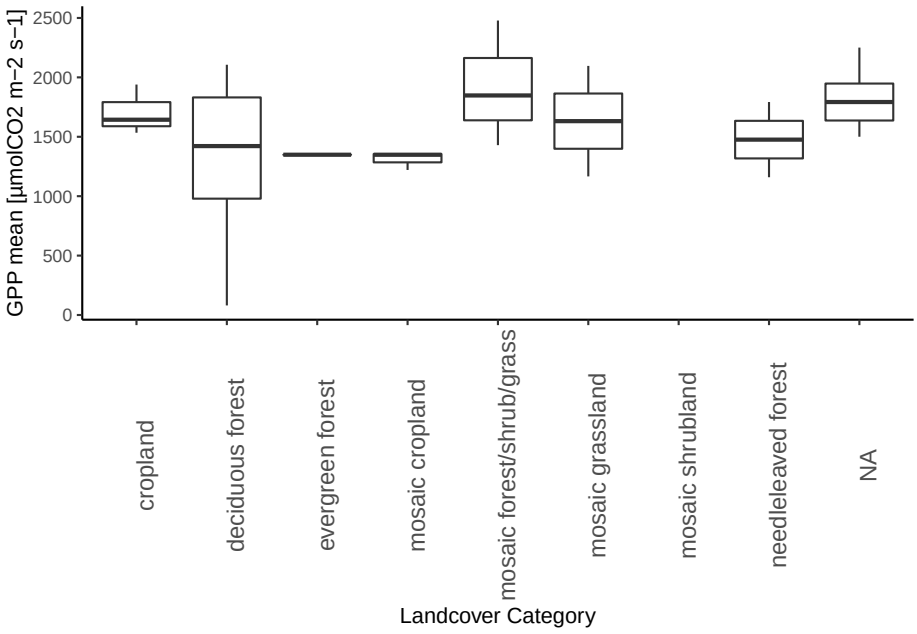


Then we make a linear model for GPP and landcover.

```
reg2 <- lm(gpp_meanann~landcover, data=df_corr)
```

… and again plot it. However, since landcover is categorical data we make a boxplot instead. We've also added a general trendline but remember this could be quite different if we correclated data from a specific landcover category with GPP.

```
ggplot(data=df_corr, aes(x=landcover_cat, y=gpp_meanann, group=as.factor(landcover_cat)
        geom_boxplot() +
        labs(x = "Landcover Category", y = "GPP mean [µmolCO2 m-2 s-1]") +
        theme_classic() +
        theme(axis.text.x = element_text(angle = 90, size=12))
```



We don't see a particularly strong correlation, thus we need to consider other predictors and more complex models. You will discuss that later in the class.

4.2.6 Key points of the tutorial

Integrating remote sensing data: we extract NDVI in the area around our eddy flux towers from the MODISTools package and conver it into a raster format.

- Collapse NDVI measured within an area around the eddy flux towers to calculate the mean across years
- Merge this temporal data with our eddy flux data
- Apply linear regression to test whether the addition of NDVI data to our model of GPP and PPFD_IN improves its accuracy

Points: the eddy flux tower sites are plotted as points on a map of Europe.

- Extract coordinates of the tower sites from the metadata
- Transform the points into SpatialPoints with the sp package and plot them on a map of Europe

Shapefiles: we add country and border information to our tower site plots.

- Extract the information about the spatial polygons in the europe_shape shapefile and apply one coordinate reference system
- Add the SpatialPoints matrix and shapefile data europe_shape to the tower site dataframe
- Filter out the countries with towers and visualize only them

Rasters: we investigate whether the variables landcover and temperature can explain GPP.

- Load landcover data from Globcover temperature data from CHELSA in the form of rasters and plot them around our tower sites with the raster and rasterVis packages
- Apply the same resolution (aggregate) and origin (align) to the climate and landcover rasters
- Compare modeled climate data from CHELSA and Worldclim to our fluxnet data by extracting the temperatures at the tower sites and calculating the MSE
- Finally, landcover and CHELSA modeled temperatures are plugged into a linear regression to see if they are significantly correlated with GPP

Bonus: here we incorporate plant species data from GBIF to determine their effect on GPP with the help of PCA.

- Create a presence-absence matrix for all locations and species
- Check species data for possible sampling biases by visualizing the number of occurrence records
- Plot species richness on the map
- Add species traits to our dataset with a for loop
- Use PCA to determine which traits explain the most variation in GPP

4.2.7 Bonus: Species Occurrence, Trait Data and PCAs

In functional ecology, we focus on the function that a species has in a community. This sub-discipline of ecology represents the intersection between ecological patterns and the mechanisms that underlie them and focuses on traits represented in large number of species. Here, we use plant distribution data around the fluxnet towers to investigate the plant community composition for each location. The community composition will allow us to get insights on which plant traits that are dominant. This is relevant because plant traits are both responsive to local climate and strong predictors of primary productivity. For example, Specific Leaf Area (SLA) and plant height, are linked to GPP and affected by local climate. Functional traits capture core differences in the strategies plants use to acquire and invest resources. Most woody plants have the same basic physiological function and key resource requirements, however species differ considerably in the rates at which resources are acquired, invested into different tissues, and lost via turnover. Traits for example moderate plant responses to light environment and determine shade tolerance. They have been shown to affect growth depending on plant size. In the first step, we use distribution data from the Global Biodiversity Information Facility (GBIF) to obtain the community composition in each location.

GBIF is an international organization that makes biodiversity data publicly available. The data are provided by many institutions from around the world

and GBIF gathers these data and makes them accessible and searchable through their platform. Data available on GBIF are primarily distribution records on plants, animals, fungi, and microbes for the world, and scientific names data. This is very useful information for ecological analysis. With the "rgbif" package we can download occurrence data directly from the GBIF database to R.

4.2.7.0.1 Species Occurrences

We start by to loading the relevant packages.

```
library(rgbif)
library(rgeos)
```

In this first, section we will give you an example of how to download data from gbif for the CH-Lae site.

```
# combine the CH-Lae latitude and longitude
CHLae_lonlat <- as.data.frame(cbind(longitude=lon_lae, latitude=lat_lae))

# make them into 'SpatialPoints', this is an alternative method to the one used above
coordinates(CHLae_lonlat) <- ~longitude + latitude

# assign a projection and coordinate reference system
crs(CHLae_lonlat) <- "+init=epsg:4326 +proj=longlat +datum=WGS84 +no_defs"

# create a roughly 5.5km buffer around the CH-Lae tower
TowerBuffer <- gBuffer(CHLae_lonlat, width=0.05)

# transform the tower buffer into polygons
tower.polygon <- polygons(TowerBuffer)
```

Then transform the polygon to wkt format which is needed for the gbif input:

```
tower.polygon.wkt <- writeWKT(tower.polygon)
```

Now we are ready to download occurrence records of plants around the CH-Lae tower from gbif with `occ_search()` function. We add the hasCoordinate = T, to make sure we only download data with coordinates. The *geometry* argument will be equal to our constructed polygon of the buffer.

```
occ_CHlae <- occ_data(taxonKey = 7707728, hasCoordinate = T,
                      geometry = tower.polygon.wkt, limit = 50000)

# The output is a list so we extract the actual data since this is of interest
occ_CHlae <- as.data.frame(occ_CHlae$data)
```

```
##          key                 scientificName decimalLatitude decimalLongitude
## 1 3028451251 Berberis julianae C.K.Schneid.        47.46482         8.333900
## 2 2646869619            Corylus avellana L.        47.44893         8.356892
```

```
## 3 2649395661             Daphne laureola L.        47.48192         8.317558
## 4 2649628860                Hedera helix L.        47.50389         8.346389
## 5 2878758835       Asplenium trichomanes L.        47.46482         8.333900
## 6 2878758840            Thlaspi montanum L.        47.46482         8.333900
##         issues                           datasetKey
## 1              83fdfd3d-3a25-4705-9fbe-3db1d1892b13
## 2 cdround,cudc 14d5676a-2c54-4f94-9023-1e8dcd822aa0
## 3 cdround,cudc 14d5676a-2c54-4f94-9023-1e8dcd822aa0
## 4 cdround,cudc 14d5676a-2c54-4f94-9023-1e8dcd822aa0
## 5              83fdfd3d-3a25-4705-9fbe-3db1d1892b13
## 6              83fdfd3d-3a25-4705-9fbe-3db1d1892b13
##                       publishingOrgKey                      installationKey
## 1 64ee55c9-570a-42af-b7da-3f13c6b4e5a9 dc9d8bd0-3c20-4fba-b7ff-26d654817ea7
## 2 da86174a-a605-43a4-a5e8-53d484152cd3 fe1a98a6-82b8-4652-86d6-d0f938207f67
## 3 da86174a-a605-43a4-a5e8-53d484152cd3 fe1a98a6-82b8-4652-86d6-d0f938207f67
## 4 da86174a-a605-43a4-a5e8-53d484152cd3 fe1a98a6-82b8-4652-86d6-d0f938207f67
## 5 64ee55c9-570a-42af-b7da-3f13c6b4e5a9 dc9d8bd0-3c20-4fba-b7ff-26d654817ea7
## 6 64ee55c9-570a-42af-b7da-3f13c6b4e5a9 dc9d8bd0-3c20-4fba-b7ff-26d654817ea7
##   publishingCountry    protocol                  lastCrawled
## 1                CH DWC_ARCHIVE 2021-02-02T15:59:13.567+00:00
## 2                FR DWC_ARCHIVE 2020-12-10T14:09:16.521+00:00
## 3                FR DWC_ARCHIVE 2020-12-10T14:09:16.521+00:00
## 4                FR DWC_ARCHIVE 2020-12-10T14:09:16.521+00:00
## 5                CH DWC_ARCHIVE 2021-02-02T15:59:13.567+00:00
## 6                CH DWC_ARCHIVE 2021-02-02T15:59:13.567+00:00
##                      lastParsed crawlId               hostingOrganizationKey
## 1 2021-02-02T16:35:54.759+00:00      17 23e067c0-a255-11da-beae-b8a03c50a862
## 2 2020-12-15T17:21:35.615+00:00      22 da86174a-a605-43a4-a5e8-53d484152cd3
## 3 2020-12-15T17:22:27.362+00:00      22 da86174a-a605-43a4-a5e8-53d484152cd3
## 4 2020-12-15T17:22:28.963+00:00      22 da86174a-a605-43a4-a5e8-53d484152cd3
## 5 2021-02-02T16:36:21.675+00:00      17 23e067c0-a255-11da-beae-b8a03c50a862
## 6 2021-02-02T16:35:36.344+00:00      17 23e067c0-a255-11da-beae-b8a03c50a862
##   extensions     basisOfRecord occurrenceStatus taxonKey kingdomKey phylumKey
## 1       none HUMAN_OBSERVATION          PRESENT  3981543          6   7707728
## 2       none HUMAN_OBSERVATION          PRESENT  2875979          6   7707728
## 3       none HUMAN_OBSERVATION          PRESENT  5420853          6   7707728
## 4       none HUMAN_OBSERVATION          PRESENT  8351737          6   7707728
## 5       none HUMAN_OBSERVATION          PRESENT  8108066          6   7707728
## 6       none HUMAN_OBSERVATION          PRESENT  3045290          6   7707728
##   classKey orderKey familyKey genusKey speciesKey acceptedTaxonKey
## 1      220      399      6673  3033893    3981543          3981543
## 2      220     1354      4688  2875967    2875979          2875979
## 3      220      941      2428  3188453    5420853          5420853
## 4      220     1351      8800  3036021    8351737          8351737
## 5  7228684      392      6625  2650583    8108066          8108066
## 6      220  7225535      3112  3045219    3045261          3045261
```

```
##            acceptedScientificName kingdom       phylum        order
## 1  Berberis julianae C.K.Schneid. Plantae Tracheophyta Ranunculales
## 2             Corylus avellana L. Plantae Tracheophyta      Fagales
## 3              Daphne laureola L. Plantae Tracheophyta     Malvales
## 4                 Hedera helix L. Plantae Tracheophyta      Apiales
## 5        Asplenium trichomanes L. Plantae Tracheophyta Polypodiales
## 6 Noccaea fendleri (A.Gray) Holub Plantae Tracheophyta  Brassicales
##          family     genus               species genericName specificEpithet
## 1 Berberidaceae  Berberis     Berberis julianae    Berberis        julianae
## 2    Betulaceae   Corylus      Corylus avellana     Corylus        avellana
## 3 Thymelaeaceae    Daphne       Daphne laureola      Daphne        laureola
## 4    Araliaceae    Hedera          Hedera helix      Hedera           helix
## 5  Aspleniaceae Asplenium Asplenium trichomanes   Asplenium     trichomanes
## 6  Brassicaceae   Noccaea      Noccaea fendleri     Thlaspi        montanum
##   taxonRank taxonomicStatus coordinateUncertaintyInMeters elevation
## 1   SPECIES        ACCEPTED                           3535     366.5
## 2   SPECIES        ACCEPTED                             23        NA
## 3   SPECIES        ACCEPTED                             65        NA
## 4   SPECIES        ACCEPTED                             NA        NA
## 5   SPECIES        ACCEPTED                           3535     610.0
## 6   SPECIES         SYNONYM                           3535     550.0
##   elevationAccuracy stateProvince year month           eventDate
## 1               1.5            Ag 2021     1 2021-01-01T00:00:00
## 2                NA          <NA> 2020     1 2020-01-18T12:23:07
## 3                NA          <NA> 2020     1 2020-01-06T09:52:11
## 4                NA          <NA> 2020     1 2020-01-18T17:48:09
## 5                NA            Ag 2020     1 2020-01-02T00:00:00
## 6                NA            Ag 2020     1 2020-01-02T00:00:00
##                  lastInterpreted
## 1 2021-02-02T16:35:54.759+00:00
## 2 2020-12-15T17:21:35.615+00:00
## 3 2020-12-15T17:22:27.362+00:00
## 4 2020-12-15T17:22:28.963+00:00
## 5 2021-02-02T16:36:21.675+00:00
## 6 2021-02-02T16:35:36.344+00:00
##                                                 license identifiers facts
## 1 http://creativecommons.org/licenses/by/4.0/legalcode        none  none
## 2 http://creativecommons.org/licenses/by/4.0/legalcode        none  none
## 3 http://creativecommons.org/licenses/by/4.0/legalcode        none  none
## 4 http://creativecommons.org/licenses/by/4.0/legalcode        none  none
## 5 http://creativecommons.org/licenses/by/4.0/legalcode        none  none
## 6 http://creativecommons.org/licenses/by/4.0/legalcode        none  none
##   relations gadm.level0.gid gadm.level0.name gadm.level1.gid gadm.level1.name
## 1      none             CHE      Switzerland         CHE.1_1           Aargau
## 2      none             CHE      Switzerland         CHE.1_1           Aargau
## 3      none             CHE      Switzerland         CHE.1_1           Aargau
```

```
## 4      none             CHE      Switzerland         CHE.1_1           Aargau
## 5      none             CHE      Switzerland         CHE.1_1           Aargau
## 6      none             CHE      Switzerland         CHE.1_1           Aargau
##   gadm.level2.gid gadm.level2.name gadm.level3.gid gadm.level3.name isInCluster
## 1       CHE.1.2_1            Baden    CHE.1.2.23_1        Wettingen       FALSE
## 2       CHE.1.2_1            Baden    CHE.1.2.26_1         Würenlos       FALSE
## 3       CHE.1.2_1            Baden     CHE.1.2.6_1       Ennetbaden       FALSE
## 4       CHE.1.2_1            Baden     CHE.1.2.5_1      Ehrendingen       FALSE
## 5       CHE.1.2_1            Baden    CHE.1.2.23_1        Wettingen       FALSE
## 6       CHE.1.2_1            Baden    CHE.1.2.23_1        Wettingen       FALSE
##   geodeticDatum          class countryCode recordedByIDs identifiedByIDs
## 1         WGS84  Magnoliopsida          CH          none            none
## 2         WGS84  Magnoliopsida          CH          none            none
## 3         WGS84  Magnoliopsida          CH          none            none
## 4         WGS84  Magnoliopsida          CH          none            none
## 5         WGS84 Polypodiopsida          CH          none            none
## 6         WGS84  Magnoliopsida          CH          none            none
##       country rightsHolder              identifier nomenclaturalCode
## 1 Switzerland   Info Flora INFOFLORA-TRAC-10029158               ICN
## 2 Switzerland         <NA>           q-10128206654              <NA>
## 3 Switzerland         <NA>           q-10127072482              <NA>
## 4 Switzerland         <NA>           q-10128246487              <NA>
## 5 Switzerland   Info Flora  INFOFLORA-TRAC-8884922               ICN
## 6 Switzerland   Info Flora  INFOFLORA-TRAC-8885363               ICN
##      georeferencedBy
## 1 observer/collector
## 2              <NA>
## 3              <NA>
## 4              <NA>
## 5 observer/collector
## 6 observer/collector
##
## 1 In order to respect the currently nationally agreed ethical framework while simul
## 2
## 3
## 4
## 5 In order to respect the currently nationally agreed ethical framework while simul
## 6 In order to respect the currently nationally agreed ethical framework while simul
##                                   datasetName     gbifID            occurrenceID
## 1 Swiss National Databank of Vascular Plants 3028451251 INFOFLORA-TRAC-10029158
## 2                                       <NA> 2646869619           q-10128206654
## 3                                       <NA> 2649395661           q-10127072482
## 4                                       <NA> 2649628860           q-10128246487
## 5 Swiss National Databank of Vascular Plants 2878758835  INFOFLORA-TRAC-8884922
## 6 Swiss National Databank of Vascular Plants 2878758840  INFOFLORA-TRAC-8885363
##                           taxonID           catalogNumber       recordedBy
```

```
## 1  infospecies.ch:infoflora:59155 INFOFLORA-TRAC-10029158        anonymous
## 2                            <NA>                    <NA>             <NA>
## 3                            <NA>                    <NA>             <NA>
## 4                            <NA>                    <NA>             <NA>
## 5  infospecies.ch:infoflora:51400  INFOFLORA-TRAC-8884922 Werner Christian
## 6 infospecies.ch:infoflora:418700  INFOFLORA-TRAC-8885363 Werner Christian
##   institutionCode      datasetID
## 1       INFOFLORA INFOFLORA-TRAC
## 2            <NA>           <NA>
## 3            <NA>           <NA>
## 4            <NA>           <NA>
## 5       INFOFLORA INFOFLORA-TRAC
## 6       INFOFLORA INFOFLORA-TRAC
##                                                                  bibliographicCitation
## 1      Info Flora (2021-02-01) INFOFLORA-TRAC-10029158. Plantes vasculaires, Info Flora
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5 Christian Werner (2021-02-01) INFOFLORA-TRAC-8884922. Plantes vasculaires, Info Flora
## 6 Christian Werner (2021-02-01) INFOFLORA-TRAC-8885363. Plantes vasculaires, Info Flora
##                          name projectId individualCount day fieldNumber
## 1 Berberis julianae C.K.Schneid.      <NA>              NA  NA        <NA>
## 2            Corylus avellana L.   queries               1  18        <NA>
## 3             Daphne laureola L.   queries               1   6        <NA>
## 4                Hedera helix L.   queries               1  18        <NA>
## 5       Asplenium trichomanes L.      <NA>              NA   2        <NA>
## 6            Thlaspi montanum L.      <NA>              NA   2        <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rightsHolder
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
## 6                                                                                    <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identifier
## 1                                                                                  <NA>
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5                                                                                  <NA>
## 6                                                                                  <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type
## 1                                                                             <NA>
## 2                                                                             <NA>
## 3                                                                             <NA>
## 4                                                                             <NA>
```

```
## 5                                                                                                         <NA>
## 6                                                                                                         <NA>
##    extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.ca
## 1
## 2
## 3
## 4
## 5
## 6
##    extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publis
## 1                                                                                 <
## 2                                                                                 <
## 3                                                                                 <
## 4                                                                                 <
## 5                                                                                 <
## 6                                                                                 <
##    extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.licens
## 1                                                                               <N
## 2                                                                               <N
## 3                                                                               <N
## 4                                                                               <N
## 5                                                                               <N
## 6                                                                               <N
##    extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                               <N
## 2                                                                               <N
## 3                                                                               <N
## 4                                                                               <N
## 5                                                                               <N
## 6                                                                               <N
##    extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format
## 1                                                                              <NA
## 2                                                                              <NA
## 3                                                                              <NA
## 4                                                                              <NA
## 5                                                                              <NA
## 6                                                                              <NA
##    extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.refere
## 1
## 2
## 3
## 4
## 5
## 6
##    extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                               <N
```

```
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rightsHolder.1
## 1                                                                                      <NA>
## 2                                                                                      <NA>
## 3                                                                                      <NA>
## 4                                                                                      <NA>
## 5                                                                                      <NA>
## 6                                                                                      <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identifier.1
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
## 6                                                                                    <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.1
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.catalogNumbe
## 1                                                                                           <
## 2                                                                                           <
## 3                                                                                           <
## 4                                                                                           <
## 5                                                                                           <
## 6                                                                                           <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publisher.1
## 1                                                                                  <NA>
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5                                                                                  <NA>
## 6                                                                                  <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.license.1
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
```

```
## 6                                                                                  <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                                  <
## 2                                                                                  <
## 3                                                                                  <
## 4                                                                                  <
## 5                                                                                  <
## 6                                                                                  <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.forma
## 1                                                                                 <
## 2                                                                                 <
## 3                                                                                 <
## 4                                                                                 <
## 5                                                                                 <
## 6                                                                                 <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.refere
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.creato
## 1                                                                                  <
## 2                                                                                  <
## 3                                                                                  <
## 4                                                                                  <
## 5                                                                                  <
## 6                                                                                  <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rights
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identi
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.
## 1                                                                               <NA
## 2                                                                               <NA
```

```
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.catalogNumbe
## 1                                                                                             <
## 2                                                                                             <
## 3                                                                                             <
## 4                                                                                             <
## 5                                                                                             <
## 6                                                                                             <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publisher.2
## 1                                                                                   <NA>
## 2                                                                                   <NA>
## 3                                                                                   <NA>
## 4                                                                                   <NA>
## 5                                                                                   <NA>
## 6                                                                                   <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.license.2
## 1                                                                                 <NA>
## 2                                                                                 <NA>
## 3                                                                                 <NA>
## 4                                                                                 <NA>
## 5                                                                                 <NA>
## 6                                                                                 <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.created.2
## 1                                                                                 <NA>
## 2                                                                                 <NA>
## 3                                                                                 <NA>
## 4                                                                                 <NA>
## 5                                                                                 <NA>
## 6                                                                                 <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format.2
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.references.2
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
## 6                                                                                    <NA>
```

```
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.creato
## 1                                                                                   <
## 2                                                                                   <
## 3                                                                                   <
## 4                                                                                   <
## 5                                                                                   <
## 6                                                                                   <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rights
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identi
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.3
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.ca
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publis
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.licens
## 1                                                                                   <
## 2                                                                                   <
## 3                                                                                   <
```

```
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.created.3
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format.3
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.references.3
## 1                                                                                  <NA>
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5                                                                                  <NA>
## 6                                                                                  <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.creator.3
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rightsHolder.4
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
## 6                                                                                    <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identifier.4
## 1                                                                                 <NA>
## 2                                                                                 <NA>
## 3                                                                                 <NA>
## 4                                                                                 <NA>
## 5                                                                                 <NA>
## 6                                                                                 <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.4
```

```
## 1                                                                    <NA>
## 2                                                                    <NA>
## 3                                                                    <NA>
## 4                                                                    <NA>
## 5                                                                    <NA>
## 6                                                                    <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.ca
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publis
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.licens
## 1                                                                      <
## 2                                                                      <
## 3                                                                      <
## 4                                                                      <
## 5                                                                      <
## 6                                                                      <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                      <
## 2                                                                      <
## 3                                                                      <
## 4                                                                      <
## 5                                                                      <
## 6                                                                      <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format
## 1                                                                     <N
## 2                                                                     <N
## 3                                                                     <N
## 4                                                                     <N
## 5                                                                     <N
## 6                                                                     <N
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.refere
## 1
## 2
## 3
## 4
```

```
## 5                                                                      <NA>
## 6                                                                      <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.creator.4
## 1                                                                   <NA>
## 2                                                                   <NA>
## 3                                                                   <NA>
## 4                                                                   <NA>
## 5                                                                   <NA>
## 6                                                                   <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rightsHolder.5
## 1                                                                        <NA>
## 2                                                                        <NA>
## 3                                                                        <NA>
## 4                                                                        <NA>
## 5                                                                        <NA>
## 6                                                                        <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identifier.5
## 1                                                                      <NA>
## 2                                                                      <NA>
## 3                                                                      <NA>
## 4                                                                      <NA>
## 5                                                                      <NA>
## 6                                                                      <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.5
## 1                                                                <NA>
## 2                                                                <NA>
## 3                                                                <NA>
## 4                                                                <NA>
## 5                                                                <NA>
## 6                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.catalogNumbe
## 1                                                                            <
## 2                                                                            <
## 3                                                                            <
## 4                                                                            <
## 5                                                                            <
## 6                                                                            <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publisher.5
## 1                                                                     <NA>
## 2                                                                     <NA>
## 3                                                                     <NA>
## 4                                                                     <NA>
## 5                                                                     <NA>
## 6                                                                     <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.license.5
## 1                                                                   <NA>
```

```
## 2                                                                                 <
## 3                                                                                 <
## 4                                                                                 <
## 5                                                                                 <
## 6                                                                                 <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                                 <
## 2                                                                                 <
## 3                                                                                 <
## 4                                                                                 <
## 5                                                                                 <
## 6                                                                                 <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format
## 1                                                                                <N
## 2                                                                                <N
## 3                                                                                <N
## 4                                                                                <N
## 5                                                                                <N
## 6                                                                                <N
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.refere
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                                 <
## 2                                                                                 <
## 3                                                                                 <
## 4                                                                                 <
## 5                                                                                 <
## 6                                                                                 <
##   dateIdentified modified references http...unknown.org.nick verbatimEventDate
## 1           <NA>     <NA>       <NA>                    <NA>              <NA>
## 2           <NA>     <NA>       <NA>                    <NA>              <NA>
## 3           <NA>     <NA>       <NA>                    <NA>              <NA>
## 4           <NA>     <NA>       <NA>                    <NA>              <NA>
## 5           <NA>     <NA>       <NA>                    <NA>              <NA>
## 6           <NA>     <NA>       <NA>                    <NA>              <NA>
##   verbatimLocality collectionCode http...unknown.org.occurrenceDetails rights
## 1             <NA>           <NA>                                <NA>   <NA>
## 2             <NA>           <NA>                                <NA>   <NA>
## 3             <NA>           <NA>                                <NA>   <NA>
## 4             <NA>           <NA>                                <NA>   <NA>
## 5             <NA>           <NA>                                <NA>   <NA>
```

```
## 6             <NA>           <NA>                                  <NA>   <NA>
##   eventTime identifiedBy identificationID
## 1      <NA>         <NA>             <NA>
## 2      <NA>         <NA>             <NA>
## 3      <NA>         <NA>             <NA>
## 4      <NA>         <NA>             <NA>
## 5      <NA>         <NA>             <NA>
## 6      <NA>         <NA>             <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rightsHolder.6
## 1                                                                                       <NA>
## 2                                                                                       <NA>
## 3                                                                                       <NA>
## 4                                                                                       <NA>
## 5                                                                                       <NA>
## 6                                                                                       <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identifier.6
## 1                                                                                     <NA>
## 2                                                                                     <NA>
## 3                                                                                     <NA>
## 4                                                                                     <NA>
## 5                                                                                     <NA>
## 6                                                                                     <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.6
## 1                                                                               <NA>
## 2                                                                               <NA>
## 3                                                                               <NA>
## 4                                                                               <NA>
## 5                                                                               <NA>
## 6                                                                               <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.catalogNumbe
## 1                                                                                           <
## 2                                                                                           <
## 3                                                                                           <
## 4                                                                                           <
## 5                                                                                           <
## 6                                                                                           <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publisher.6
## 1                                                                                   <NA>
## 2                                                                                   <NA>
## 3                                                                                   <NA>
## 4                                                                                   <NA>
## 5                                                                                   <NA>
## 6                                                                                   <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.license.6
## 1                                                                                 <NA>
## 2                                                                                 <NA>
```

```
## 3                                                                                              <
## 4                                                                                              <
## 5                                                                                              <
## 6                                                                                              <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                                              <
## 2                                                                                              <
## 3                                                                                              <
## 4                                                                                              <
## 5                                                                                              <
## 6                                                                                              <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format
## 1                                                                                             <
## 2                                                                                             <
## 3                                                                                             <
## 4                                                                                             <
## 5                                                                                             <
## 6                                                                                             <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.refere
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                                              <
## 2                                                                                              <
## 3                                                                                              <
## 4                                                                                              <
## 5                                                                                              <
## 6                                                                                              <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rights
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identi
## 1
## 2
## 3
## 4
## 5
## 6
```

```
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.7
## 1                                                                               <NA>
## 2                                                                               <NA>
## 3                                                                               <NA>
## 4                                                                               <NA>
## 5                                                                               <NA>
## 6                                                                               <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.catalogNumbe
## 1                                                                                            <
## 2                                                                                            <
## 3                                                                                            <
## 4                                                                                            <
## 5                                                                                            <
## 6                                                                                            <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publisher.7
## 1                                                                                  <NA>
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5                                                                                  <NA>
## 6                                                                                  <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.license.7
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.created.7
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format.7
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.references.7
## 1                                                                                   <NA>
## 2                                                                                   <NA>
## 3                                                                                   <NA>
```

```
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.creato
## 1                                                                                  <
## 2                                                                                  <
## 3                                                                                  <
## 4                                                                                  <
## 5                                                                                  <
## 6                                                                                  <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rights
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identi
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.8
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.ca
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publis
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.licens
```

```
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.created.8
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format.8
## 1                                                                               <NA>
## 2                                                                               <NA>
## 3                                                                               <NA>
## 4                                                                               <NA>
## 5                                                                               <NA>
## 6                                                                               <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.references.8
## 1                                                                                  <NA>
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5                                                                                  <NA>
## 6                                                                                  <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.creator.8
## 1                                                                               <NA>
## 2                                                                               <NA>
## 3                                                                               <NA>
## 4                                                                               <NA>
## 5                                                                               <NA>
## 6                                                                               <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.rightsHolder.9
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
## 6                                                                                    <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.identifier.9
## 1                                                                                  <NA>
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
```

```
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.type.S
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...rs.tdwg.org.dwc.terms.ca
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.publis
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.licens
## 1                                                                                  <
## 2                                                                                  <
## 3                                                                                  <
## 4                                                                                  <
## 5                                                                                  <
## 6                                                                                  <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.create
## 1                                                                                  <
## 2                                                                                  <
## 3                                                                                  <
## 4                                                                                  <
## 5                                                                                  <
## 6                                                                                  <
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.format
## 1                                                                                 <I
## 2                                                                                 <I
## 3                                                                                 <I
## 4                                                                                 <I
## 5                                                                                 <I
## 6                                                                                 <I
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.refere
## 1
```

```
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.gbif.org.terms.1.0.Multimedia.http...purl.org.dc.terms.creator.9
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.identifier
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.taxonCoverage
## 1                                                                                      <NA>
## 2                                                                                      <NA>
## 3                                                                                      <NA>
## 4                                                                                      <NA>
## 5                                                                                      <NA>
## 6                                                                                      <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.modified
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.type
## 1                                                                                 <NA>
## 2                                                                                 <NA>
## 3                                                                                 <NA>
## 4                                                                                 <NA>
## 5                                                                                 <NA>
## 6                                                                                 <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.subjectPart
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
```

```
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.asso
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.rigl
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.taxo
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.creator
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.tag
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.rigl
## 1
## 2
```

```
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5                                                                                  <NA>
## 6                                                                                  <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.format
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.photoshop.1.0.Credit
## 1                                                                                  <NA>
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5                                                                                  <NA>
## 6                                                                                  <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.creator
## 1                                                                                 <NA>
## 2                                                                                 <NA>
## 3                                                                                 <NA>
## 4                                                                                 <NA>
## 5                                                                                 <NA>
## 6                                                                                 <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.CreateDate
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.caption
## 1                                                                             <NA>
## 2                                                                             <NA>
## 3                                                                             <NA>
## 4                                                                             <NA>
## 5                                                                             <NA>
## 6                                                                             <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.title
## 1                                                                         <NA>
## 2                                                                         <NA>
## 3                                                                         <NA>
## 4                                                                         <NA>
## 5                                                                         <NA>
## 6                                                                         <NA>
```

```
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.acce
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.rights
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.source
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.type
## 1                                                                            <NA>
## 2                                                                            <NA>
## 3                                                                            <NA>
## 4                                                                            <NA>
## 5                                                                            <NA>
## 6                                                                            <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.format
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   infraspecificEpithet
## 1                 <NA>
## 2                 <NA>
## 3                 <NA>
```

```
## 4                 <NA>
## 5                 <NA>
## 6                 <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.identifier.1
## 1                                                                                   <NA>
## 2                                                                                   <NA>
## 3                                                                                   <NA>
## 4                                                                                   <NA>
## 5                                                                                   <NA>
## 6                                                                                   <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.taxonCoverage.
## 1                                                                                          <NA
## 2                                                                                          <NA
## 3                                                                                          <NA
## 4                                                                                          <NA
## 5                                                                                          <NA
## 6                                                                                          <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.modified.1
## 1                                                                                <NA>
## 2                                                                                <NA>
## 3                                                                                <NA>
## 4                                                                                <NA>
## 5                                                                                <NA>
## 6                                                                                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.type.1
## 1                                                                                     <NA>
## 2                                                                                     <NA>
## 3                                                                                     <NA>
## 4                                                                                     <NA>
## 5                                                                                     <NA>
## 6                                                                                     <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.subjectPart.1
## 1                                                                                        <NA>
## 2                                                                                        <NA>
## 3                                                                                        <NA>
## 4                                                                                        <NA>
## 5                                                                                        <NA>
## 6                                                                                        <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.associatedObse
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.rights.UsageTe
```

```
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.taxo
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.creator
## 1                                                                                <N
## 2                                                                                <N
## 3                                                                                <N
## 4                                                                                <N
## 5                                                                                <N
## 6                                                                                <N
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.tag
## 1                                                                              <NA
## 2                                                                              <NA
## 3                                                                              <NA
## 4                                                                              <NA
## 5                                                                              <NA
## 6                                                                              <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.righ
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1
## 1
## 2
## 3
## 4
```

```
## 5                                                                                    <NA>
## 6                                                                                    <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.photoshop.1.0.Credit.1
## 1                                                                                      <NA>
## 2                                                                                      <NA>
## 3                                                                                      <NA>
## 4                                                                                      <NA>
## 5                                                                                      <NA>
## 6                                                                                      <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.creator.1
## 1                                                                                     <NA>
## 2                                                                                     <NA>
## 3                                                                                     <NA>
## 4                                                                                     <NA>
## 5                                                                                     <NA>
## 6                                                                                     <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.CreateDate.1
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
## 6                                                                                    <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.caption.1
## 1                                                                                 <NA>
## 2                                                                                 <NA>
## 3                                                                                 <NA>
## 4                                                                                 <NA>
## 5                                                                                 <NA>
## 6                                                                                 <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.title.1
## 1                                                                            <NA>
## 2                                                                            <NA>
## 3                                                                            <NA>
## 4                                                                            <NA>
## 5                                                                            <NA>
## 6                                                                            <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.accessURI.1
## 1                                                                                   <NA>
## 2                                                                                   <NA>
## 3                                                                                   <NA>
## 4                                                                                   <NA>
## 5                                                                                   <NA>
## 6                                                                                   <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.source.1
## 1                                                                                    <NA>
```

```
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.rights
## 1                                                                                <NA
## 2                                                                                <NA
## 3                                                                                <NA
## 4                                                                                <NA
## 5                                                                                <NA
## 6                                                                                <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.source
## 1                                                                                <NA
## 2                                                                                <NA
## 3                                                                                <NA
## 4                                                                                <NA
## 5                                                                                <NA
## 6                                                                                <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.type.1
## 1                                                                             <NA>
## 2                                                                             <NA>
## 3                                                                             <NA>
## 4                                                                             <NA>
## 5                                                                             <NA>
## 6                                                                             <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.format
## 1                                                                                <NA
## 2                                                                                <NA
## 3                                                                                <NA
## 4                                                                                <NA
## 5                                                                                <NA
## 6                                                                                <NA
##   informationWithheld
## 1                <NA>
## 2                <NA>
## 3                <NA>
## 4                <NA>
## 5                <NA>
## 6                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.identi
## 1
## 2
## 3
## 4
## 5
```

```
## 6                                                                                   <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.taxonCoverage.
## 1                                                                                        <NA
## 2                                                                                        <NA
## 3                                                                                        <NA
## 4                                                                                        <NA
## 5                                                                                        <NA
## 6                                                                                        <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.modified.2
## 1                                                                                 <NA>
## 2                                                                                 <NA>
## 3                                                                                 <NA>
## 4                                                                                 <NA>
## 5                                                                                 <NA>
## 6                                                                                 <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.type.2
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
## 6                                                                                    <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.subjectPart.2
## 1                                                                                       <NA>
## 2                                                                                       <NA>
## 3                                                                                       <NA>
## 4                                                                                       <NA>
## 5                                                                                       <NA>
## 6                                                                                       <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.associatedObse
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.rights.UsageTe
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.taxonCount.2
## 1                                                                                      <NA>
## 2                                                                                      <NA>
```

```
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.creator
## 1                                                                                  <N
## 2                                                                                  <N
## 3                                                                                  <N
## 4                                                                                  <N
## 5                                                                                  <N
## 6                                                                                  <N
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.tag
## 1                                                                               <NA
## 2                                                                               <NA
## 3                                                                               <NA
## 4                                                                               <NA
## 5                                                                               <NA
## 6                                                                               <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.righ
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.photoshop.1
## 1
## 2
## 3
## 4
## 5
## 6
```

```
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.creator.2
## 1                                                                                        <NA>
## 2                                                                                        <NA>
## 3                                                                                        <NA>
## 4                                                                                        <NA>
## 5                                                                                        <NA>
## 6                                                                                        <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.CreateDate.2
## 1                                                                                      <NA>
## 2                                                                                      <NA>
## 3                                                                                      <NA>
## 4                                                                                      <NA>
## 5                                                                                      <NA>
## 6                                                                                      <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.caption.2
## 1                                                                                   <NA>
## 2                                                                                   <NA>
## 3                                                                                   <NA>
## 4                                                                                   <NA>
## 5                                                                                   <NA>
## 6                                                                                   <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.title.2
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.accessURI.2
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
## 5                                                                                    <NA>
## 6                                                                                    <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.source.2
## 1                                                                                       <NA>
## 2                                                                                       <NA>
## 3                                                                                       <NA>
## 4                                                                                       <NA>
## 5                                                                                       <NA>
## 6                                                                                       <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.rights.2
## 1                                                                             <NA>
## 2                                                                             <NA>
## 3                                                                             <NA>
```

```
## 4                                                                                <NA
## 5                                                                                <NA
## 6                                                                                <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.source
## 1                                                                                <NA
## 2                                                                                <NA
## 3                                                                                <NA
## 4                                                                                <NA
## 5                                                                                <NA
## 6                                                                                <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.type.2
## 1                                                                             <NA>
## 2                                                                             <NA>
## 3                                                                             <NA>
## 4                                                                             <NA>
## 5                                                                             <NA>
## 6                                                                             <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.format
## 1                                                                                <NA
## 2                                                                                <NA
## 3                                                                                <NA
## 4                                                                                <NA
## 5                                                                                <NA
## 6                                                                                <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.vari
## 1
## 2
## 3
## 4
## 5
## 6
##   lifeStage continent vernacularName locality samplingProtocol
## 1      <NA>      <NA>           <NA>     <NA>             <NA>
## 2      <NA>      <NA>           <NA>     <NA>             <NA>
## 3      <NA>      <NA>           <NA>     <NA>             <NA>
## 4      <NA>      <NA>           <NA>     <NA>             <NA>
## 5      <NA>      <NA>           <NA>     <NA>             <NA>
## 6      <NA>      <NA>           <NA>     <NA>             <NA>
##   higherClassification footprintWKT ownerInstitutionCode verbatimElevation
## 1                 <NA>         <NA>                 <NA>              <NA>
## 2                 <NA>         <NA>                 <NA>              <NA>
## 3                 <NA>         <NA>                 <NA>              <NA>
## 4                 <NA>         <NA>                 <NA>              <NA>
## 5                 <NA>         <NA>                 <NA>              <NA>
## 6                 <NA>         <NA>                 <NA>              <NA>
##   recordNumber associatedReferences
```

```
## 1         <NA>                <NA>
## 2         <NA>                <NA>
## 3         <NA>                <NA>
## 4         <NA>                <NA>
## 5         <NA>                <NA>
## 6         <NA>                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.variantLiteral
## 1                                                                                          <N
## 2                                                                                          <N
## 3                                                                                          <N
## 4                                                                                          <N
## 5                                                                                          <N
## 6                                                                                          <N
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.variantLiteral
## 1                                                                                          <N
## 2                                                                                          <N
## 3                                                                                          <N
## 4                                                                                          <N
## 5                                                                                          <N
## 6                                                                                          <N
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.identifier.3
## 1                                                                                  <NA>
## 2                                                                                  <NA>
## 3                                                                                  <NA>
## 4                                                                                  <NA>
## 5                                                                                  <NA>
## 6                                                                                  <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.rights.Owner.3
## 1                                                                                        <NA>
## 2                                                                                        <NA>
## 3                                                                                        <NA>
## 4                                                                                        <NA>
## 5                                                                                        <NA>
## 6                                                                                        <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.rights.3
## 1                                                                              <NA>
## 2                                                                              <NA>
## 3                                                                              <NA>
## 4                                                                              <NA>
## 5                                                                              <NA>
## 6                                                                              <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.type.3
## 1                                                                                    <NA>
## 2                                                                                    <NA>
## 3                                                                                    <NA>
## 4                                                                                    <NA>
```

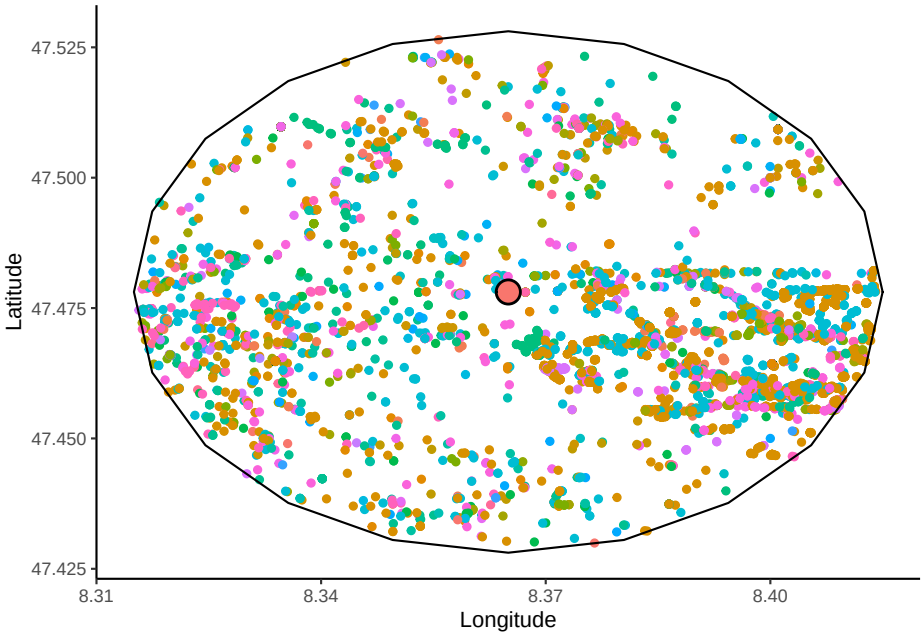
```
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.format
## 1                                                                                <NA
## 2                                                                                <NA
## 3                                                                                <NA
## 4                                                                                <NA
## 5                                                                                <NA
## 6                                                                                <NA
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.vari
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.acce
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.identi
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...ns.adobe.com.xap.1.0.rigl
## 1
## 2
## 3
## 4
## 5
## 6
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.rights
## 1                                                                                <NA
```

```
## 2                                                                                                       <NA>
## 3                                                                                                       <NA>
## 4                                                                                                       <NA>
## 5                                                                                                       <NA>
## 6                                                                                                       <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.type.4
## 1                                                                                             <NA>
## 2                                                                                             <NA>
## 3                                                                                             <NA>
## 4                                                                                             <NA>
## 5                                                                                             <NA>
## 6                                                                                             <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.elements.1.1.creator.4
## 1                                                                                                <NA>
## 2                                                                                                <NA>
## 3                                                                                                <NA>
## 4                                                                                                <NA>
## 5                                                                                                <NA>
## 6                                                                                                <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...purl.org.dc.terms.format.4
## 1                                                                                        <NA>
## 2                                                                                        <NA>
## 3                                                                                        <NA>
## 4                                                                                        <NA>
## 5                                                                                        <NA>
## 6                                                                                        <NA>
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.variantLiteral
## 1                                                                                                   <N
## 2                                                                                                   <N
## 3                                                                                                   <N
## 4                                                                                                   <N
## 5                                                                                                   <N
## 6                                                                                                   <N
##   extensions.http...rs.tdwg.org.ac.terms.Multimedia.http...rs.tdwg.org.ac.terms.accessURI.4
## 1                                                                                              <NA>
## 2                                                                                              <NA>
## 3                                                                                              <NA>
## 4                                                                                              <NA>
## 5                                                                                              <NA>
## 6                                                                                              <NA>
##   institutionKey institutionID county verbatimCoordinateSystem
## 1           <NA>          <NA>   <NA>                     <NA>
## 2           <NA>          <NA>   <NA>                     <NA>
## 3           <NA>          <NA>   <NA>                     <NA>
## 4           <NA>          <NA>   <NA>                     <NA>
## 5           <NA>          <NA>   <NA>                     <NA>
```

```
## 6           <NA>           <NA>   <NA>                       <NA>
```

Let's have a quick look at the orders of the species in the polygon around our tower.

```
# first we make the maine plot of the points coloured by order with the buffer polygon
plot_sp_occ <- ggplot()+
  geom_point(data=occ_CHlae, aes(x=decimalLongitude, y=decimalLatitude, color=order))+
  geom_polygon(data=tower.polygon,
               aes(x=tower.polygon@polygons[[1]]@Polygons[[1]]@coords[,1], y=tower.poly
  theme_classic() +
  xlab("Longitude") +
  ylab("Latitude") +
  theme(legend.position = "none")

# then we add the CH-Lae tower in the middle in red
plot_sp_occ + geom_point(aes(x=lon_lae, y=lat_lae, fill="#FC717F"), size=5, colour="bla
```



To make a species list and calculate the species richness for that tower we simply extract the unique species

```
sp.list <- unique(occ_CHlae$scientificName)
length(unique(occ_CHlae$scientificName))
```

```
## [1] 1254
```

Normally at this point a species name filtration step would be done. That means the plant names from our species list would be compared to that of 'The Plant

List' (TPL) using the package **Taxonstand**. This is quite time consuming, so we have already done it for the species lists of each tower. Below is the example code:

```
# We retrieve name from taxonstand. To specify how much difference in the names we want to allow
library(Taxonstand)
name <- TPL(sp.list.final, diffchar = 4, drop.lower.level = T)

# select columns of interest
name <- name[, c("Taxon", "New.Genus", "New.Species", "New.Taxonomic.status", "Typo")]

# create column with new species name
name$autocorrected <- paste(name$New.Genus, name$New.Species)

# drop new.species and new.genus column
name <- name[, c("Taxon", "autocorrected", "New.Taxonomic.status", "Typo")]

# change column names
colnames(name) <- c("submittedname", "autocorrected", "taxo_status", "typo")

# Finally, we can drop all species for which we did not find a matching scientific name.
names.accepted <- unique(name[name$taxo_status=="Accepted",])
```

```
# read in the resolved species names .csv
species_list_full <- read.csv("./data/resolved_species_names.csv",sep=";")
```

```
# check how many species are in the list
dim(species_list_full)
```

```
## [1] 7480    1
```

Next, we can use this information to build a presence-absence matrix for all locations and species, this will allow us to see where each species is found. This can be the basis for many community ecology calculations or species distribution models linking the environmental variables to species' occurrences to see which are most correlated with certain factors.

```
# select all files with resolved in their name
paths <- list.files(file.path("./data/", "species_occurrences"), pattern='resolved_*', recursive

# create a function to read them all in
read_species = function(file_path){
  return(read.csv(file_path,sep=";"))
}

# read all the files in paths in using the function read_species
species_occurrences <- lapply(paths, read_species)
```

```
# bind them into a df and give an identifier with .id=
species_occ_df <- bind_rows(species_occurrences, .id="id")

# make a species occurrence (or presence-absence) matrix
species_occ_mat <- species_occ_df %>%
  group_by(id)%>%
  distinct() %>%
  mutate(present = 1) %>%
  pivot_wider(names_from = names.accepted.autocorrected, values_from = present) %>%
  mutate_all(funs(ifelse(is.na(.), 0, .)))
```

```
head(species_occ_mat)
```

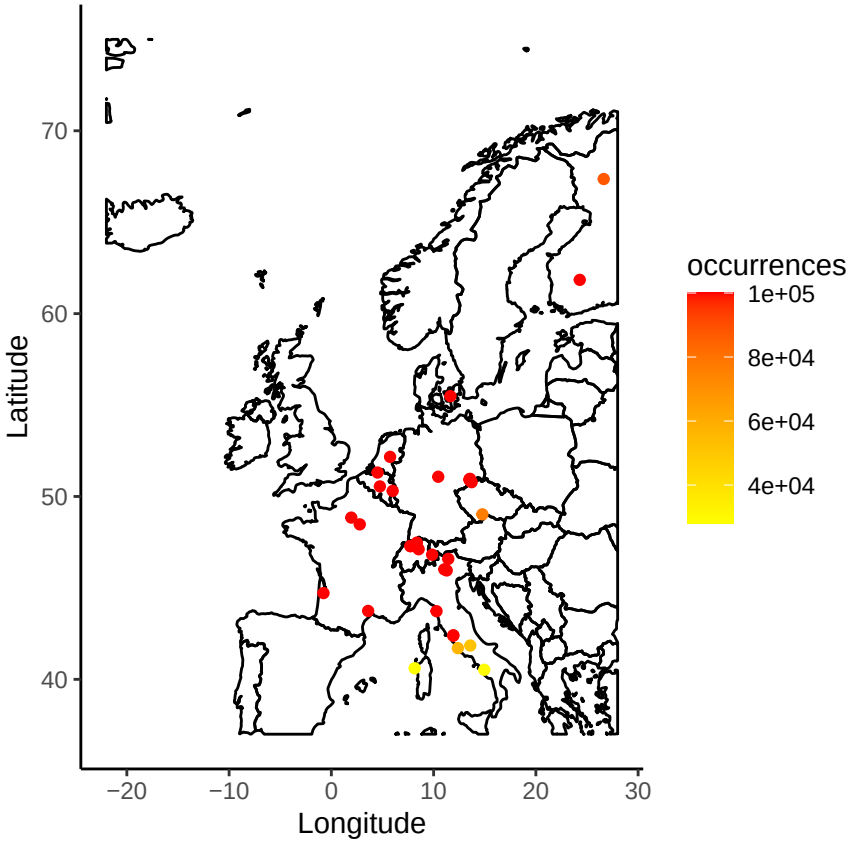
```
## # A tibble: 6 x 6,651
## # Groups:   id [6]
##   id    `Hedera helix` `Arum italicum` `Asplenium ruta~ `Ceterach offic~
##   <chr>          <dbl>           <dbl>            <dbl>            <dbl>
## 1 1                  1               1                1                1
## 2 2                  1               1                1                1
## 3 3                  1               1                1                1
## 4 4                  1               1                1                1
## 5 5                  1               1                1                1
## 6 6                  1               1                1                1
## # ... with 6,646 more variables: `Artemisia vulgaris` <dbl>, `Corylus
## #   avellana` <dbl>, `Teucrium scorodonia` <dbl>, `Glechoma hederacea` <dbl>,
## #   `Alliaria petiolata` <dbl>, `Plantago lanceolata` <dbl>, `Ficaria
## #   verna` <dbl>, `Trifolium pratense` <dbl>, `Erodium cicutarium` <dbl>,
## #   `Urtica dioica` <dbl>, `Achillea millefolium` <dbl>, `Cirsium
## #   vulgare` <dbl>, `Stellaria media` <dbl>, `Betula pendula` <dbl>,
## #   `Symphoricarpos albus` <dbl>, `Equisetum telmateia` <dbl>, `Euphorbia
## #   peplus` <dbl>, `Aphanes arvensis` <dbl>, `Geranium robertianum` <dbl>,
## #   `Clematis vitalba` <dbl>, `Lamium purpureum` <dbl>, `Lamium album` <dbl>,
## #   `Bellis perennis` <dbl>, `Geum urbanum` <dbl>, `Juncus effusus` <dbl>,
## #   `Chelidonium majus` <dbl>, `Ilex aquifolium` <dbl>, `Daucus carota` <dbl>,
## #   `Lonicera ligustrina` <dbl>, `Senecio vulgaris` <dbl>, `Oxalis
## #   pes-caprae` <dbl>, `Asplenium scolopendrium` <dbl>, `Galium aparine` <dbl>,
## #   `Ranunculus repens` <dbl>, `Calluna vulgaris` <dbl>, `Solanum
## #   dulcamara` <dbl>, `Carex pendula` <dbl>, `Cerastium glomeratum` <dbl>,
## #   `Dipsacus pilosus` <dbl>, `Buddleja davidii` <dbl>, `Pseudosasa
## #   japonica` <dbl>, `Epipactis helleborine` <dbl>, `Lunaria annua` <dbl>,
## #   `Dryopteris carthusiana` <dbl>, `Senecio inaequidens` <dbl>, `Eranthis
## #   hyemalis` <dbl>, `Aesculus hippocastanum` <dbl>, `Geranium molle` <dbl>,
## #   `Polypodium vulgare` <dbl>, `Humulus lupulus` <dbl>, `Arum
## #   maculatum` <dbl>, `Alnus glutinosa` <dbl>, `Tanacetum vulgare` <dbl>,
## #   `Verbascum thapsus` <dbl>, `Capsella bursa-pastoris` <dbl>, `Jasminum
## #   nudiflorum` <dbl>, `Galanthus nivalis` <dbl>, `Acer platanoides` <dbl>,
```

```
## #   `Crataegus monogyna` <dbl>, `Pinus sylvestris` <dbl>, `Prunus
## #   laurocerasus` <dbl>, `Quercus robur` <dbl>, `Scrophularia nodosa` <dbl>,
## #   `Erophila verna` <dbl>, `Cardamine hirsuta` <dbl>, `Epilobium
## #   angustifolium` <dbl>, `Rumex obtusifolius` <dbl>, `Anthriscus
## #   sylvestris` <dbl>, `Claytonia perfoliata` <dbl>, `Sonchus oleraceus` <dbl>,
## #   `Cirsium arvense` <dbl>, `Borago officinalis` <dbl>, `Poa annua` <dbl>,
## #   `Alnus rubra` <dbl>, `Viburnum tinus` <dbl>, `Phragmites australis` <dbl>,
## #   `Larix decidua` <dbl>, `Veronica beccabunga` <dbl>, `Lepidium
## #   virginicum` <dbl>, `Mercurialis annua` <dbl>, `Euphorbia
## #   helioscopia` <dbl>, `Cytisus scoparius` <dbl>, `Tripleurospermum
## #   inodorum` <dbl>, `Castanea sativa` <dbl>, `Veronica persica` <dbl>, `Rumex
## #   acetosa` <dbl>, `Sambucus nigra` <dbl>, `Phacelia tanacetifolia` <dbl>,
## #   `Veronica hederifolia` <dbl>, `Taxus baccata` <dbl>, `Silene dioica` <dbl>,
## #   `Asplenium trichomanes` <dbl>, `Oenothera biennis` <dbl>, `Fraxinus
## #   excelsior` <dbl>, `Chenopodium album` <dbl>, `Aucuba japonica` <dbl>,
## #   `Euonymus europaeus` <dbl>, `Taraxacum campylodes` <dbl>, `Solidago
## #   gigantea` <dbl>, `Rhododendron ponticum` <dbl>, ...
```

GBIF is a great source for species occurrences. However, we have to keep in mind that the data collection is biased by regions. For example, we find much fewer plant records in Africa than in Europe. But also within Europe, there are some biases. Therefore we need to be careful with the interpretation.

```
nr_occurrences <- read_csv("./data/flx_tower_occurrences.csv")

occurrences <- nr_occurrences$nr_occurrences
```

```
ggplot() +
  geom_polygon(data = europe_shape,
               aes(x = long, y = lat, group = group), fill = NA, colour = "black") +
  geom_point(aes(colour = occurrences, x = df_sites$lon,
                              y = df_sites$lat)) +
  scale_colour_gradient(low = "yellow", high = "red", na.value = NA) +
  theme_classic() + xlab("Longitude") + ylab("Latitude") + coord_quickmap()
```

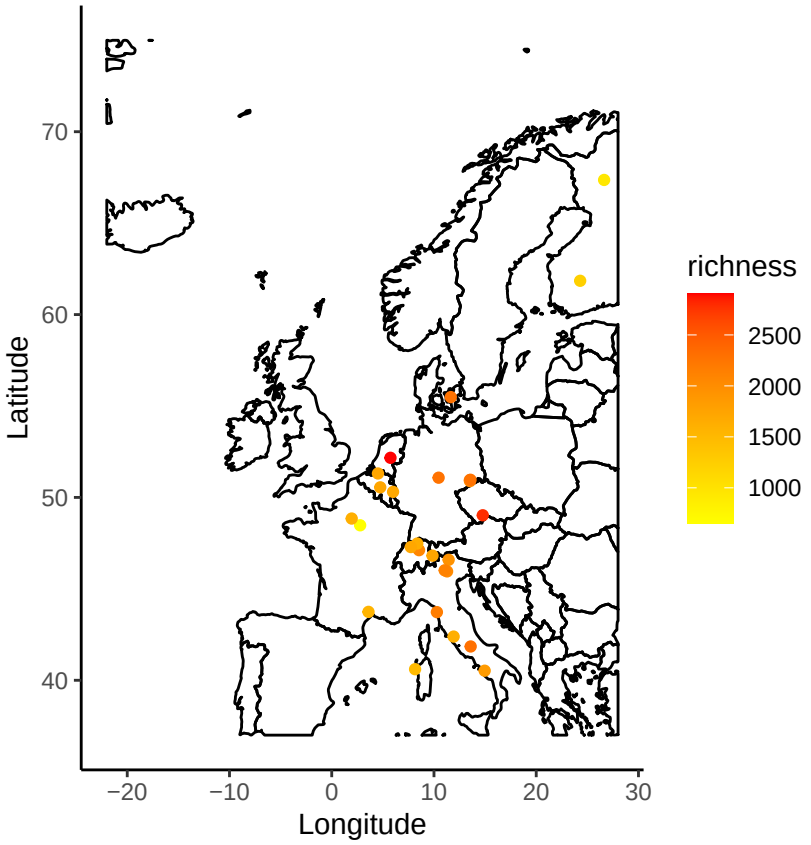


We see that the numbers of occurrence records in southern Europe are lower than in central Europe. This suggests there is a sampling bias, with a higher sampling effort in central European countries.

Knowing the species richness of the different sites can help us learn more about the ecological context the towers are located in. We made the occurrence matrix above and now we'll use that to plot the richness. The richness is the number of species at each site. Since each row in the matrix is a site we can just sum the rows and we'll get the number of species per site. We don't have data for all our towers in *df_sites* so we will make a new dataframe with only the relevant sites and call it *df_sp_sites*.

```
# remove the sites for which we have no species lists
df_sp_sites <- df_sites %>%
  filter(!(site %in% c("CH-Oe1", "DE-Obe", "FR-LBr", "IT-Cpz", "IT-Ro1")))

# the first column is the 'id', so we exclude that
richness <- rowSums(species_occ_mat[,-1])

ggplot() +
```

```
  geom_polygon(data = europe_shape,
               aes(x = long, y = lat, group = group), fill = NA, colour = "black") +
  geom_point(aes(colour = richness, x = df_sp_sites$lon,
                                y = df_sp_sites$lat)) +
  scale_colour_gradient(low = "yellow", high = "red", na.value = NA) +
  theme_classic() + xlab("Longitude") + ylab("Latitude") + coord_quickmap()
```



4.2.7.1 Species Traits

In the next step, we will study the relationship between plant traits, GPP, and climatic variables for our fluxnet towers. We obtained data from the TRY database, which is a database containing plant trait data. You can make data requests on their website. You can select the traits that you are interested in and the species for which you would like to look at the traits. For this tutorial we already requested a dataset, because usually, it takes time to receive the data. We downloaded data on 9 different traits for all species available. We extracted the trait data for all the species we have in our species list (and therefore also

in our species occurrence matrix).

Let's load the data.

```
trait_df <- read.csv("./data/Trait_df.csv")
head(trait_df)
```

```
##   X.1 X         SpeciesName
## 1   1 1      Acer campestre
## 2   2 2      Acer campestre
## 3   3 3 Acer pseudoplatanus
## 4   4 4 Acer pseudoplatanus
## 5   5 5     Alnus glutinosa
## 6   6 6     Alnus glutinosa
##                                                    TraitName TraitID
## 1                                              Leaf thickness      46
## 2 Leaf texture (sclerophylly, physical strength, toughness)       2
## 3                                              Leaf thickness      46
## 4 Leaf texture (sclerophylly, physical strength, toughness)       2
## 5                                              Leaf thickness      46
## 6 Leaf texture (sclerophylly, physical strength, toughness)       2
##   OrigValueStr OrigUnitStr OrigUncertaintyStr UncertaintyName
## 1   0.14400000          mm         0.00400000  Standard Error
## 2   0.04511078   (N/10)/mm         0.01426528  Standard Error
## 3   0.17800000          mm         0.01655295  Standard Error
## 4   0.06038498   (N/10)/mm         0.01909541  Standard Error
## 5   0.13600000          mm         0.00509902  Standard Error
## 6   0.06667092   (N/10)/mm         0.02108320  Standard Error
```

Some examples of traits are are leaf texture, leaf thickness, eaf chlorophyll content per leaf area, specific leaf area, etc.

As a next step, we can combine the species occurrence data with the trait data. The presence/absence matrix (*species_occ_mat*) we made earlier, gives us the species present at each site. From that we can then extract these species and their respective trait values from the trait dataframe (*trait_df*). Then we go on to summarise this data by calculating the mean value for each trait across all the species at the tower sites.

We start by doing this for one site.

```
ba <- trait_df

# select all the species (species names= column names) which are present (so ==1)
species_present <- colnames(species_occ_mat)[which(species_occ_mat[1, ] == 1)]

# filter out the trait data for the species present at that site
trait_subset <- ba[ba$SpeciesName %in% species_present,]
```

```
# calculate the mean of each trait across all species present
bac <- aggregate(trait_subset$OrigValueStr, list(trait_subset$TraitID), mean, na.rm=T)
```

To finish preparing the data for the PCA, we do the above steps for all the remaining sites by using a *'for loop'*.

```
df_pca <- bac$x
for(i in c(2:28)){
    species_present <- colnames(species_occ_mat)[which(species_occ_mat[i, ] == 1)]
    trait_subset <- ba[ba$SpeciesName %in% species_present,]
    bac <- aggregate(trait_subset$OrigValueStr, list(trait_subset$TraitID), mean, na.rm=T)
    df_pca <- rbind(df_pca, bac$x)
}
```

For the PCA we use only the traits with traitIDs: 46, 47, 3106 and 3117. This is because the other traits are just slightly modified versions of these four traits. These four traitIDs correspond to the traits: *leaf thickness*, *leaf dry matter content (LDMC)*, *plant height* and *specific leaf area (SLA)*. *Leaf dry matter content* is calculated as leaf dry mass per leaf fresh mass and *specific leaf area* is defined as leaf area per leaf dry mass.

```
# we select the columns corresponding to our desired traits
df_pca <- cbind(df_pca[,2],df_pca[,3],df_pca[,5],df_pca[,9])

# to make the data easier to understand we name the columns after the trait
colnames(df_pca) <- c("leaf thickness", "leaf dry matter content (LDMC)", "plant height vegetativ

head(df_pca)
```

```
##        leaf thickness leaf dry matter content (LDMC) plant height vegetative
## df_pca       18.38967                       142.1853                80.59372
##              19.11832                       144.8715                80.23070
##              19.32090                       176.7112                80.68073
##              21.47128                       144.4585                73.76227
##              22.30091                       148.8697                70.61044
##              22.17279                       144.7945                73.61253
##        specific leaf area (SLA)
## df_pca                 77.28056
##                        78.31965
##                        81.82512
##                        78.28065
##                        76.57826
##                        76.61992
```

Principal Component Analysis (PCA) is one of famous techniques for dimension reduction, feature extraction, and data visualization. In general, PCA is defined as the transformation of a high dimensional vector space to a low dimensional

space. Just imagine having to visualise data with 10 dimensions… we often stuggle already with three. It's almost impossible to effectively show the shape of such a high dimensional data distribution. PCA provides an efficient way to reduce the dimensionalty (i.e. from 10 to 2), so it is much easier to visualize the shape of data distribution. In a nutshell, this reduction happens by the PCA taking the two axes that best explain the variation in the data.

PCA is also useful in the modeling a robust classifier where considerably small number of high dimensional training data is provided. By reducing the dimensions of learning data sets, PCA provides an effective and efficient method for data description and classification. For a more mathematical description of PCA you can check https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples.

So now let's reduce the dimensions of our trait data as an example… For this we use the function `prcomp()`.

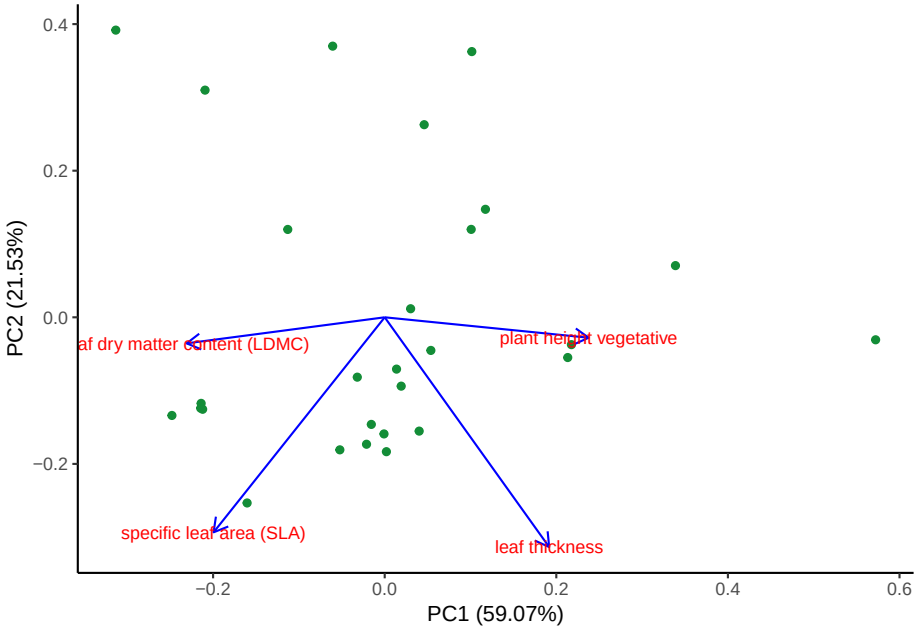
```
traits_pca <- prcomp(df_pca, center = TRUE, scale = TRUE)
summary(traits_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4
## Standard deviation     1.5372 0.9280 0.6972 0.53830
## Proportion of Variance 0.5907 0.2153 0.1215 0.07244
## Cumulative Proportion  0.5907 0.8060 0.9276 1.00000
```

From the information above we can see how much of the variation in the data is explained by each axis or dimensionality. Ideally, you want the first to axes to explain the vast majority of the variation, while the subsequent ones are much less important. But what do all these numbers mean and how does the data look. It's time to visualise the PCA.

```
library(ggfortify)
autoplot(traits_pca, data = df_pca, col ="#138D37", loadings = TRUE, loadings.colour =
```

```
## Warning: `select_()` is deprecated as of dplyr 0.7.0.
## Please use `select()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



We have successfully reduced the dimensionality of our data! We started with four traits and now we have reduced the variation to two main axes. We see that PC1 is mainly associated to leaf dry matter content, and plant height, while PC2 is associated with specific leaf area and leaf thickness.

It can be interesting to see how the points relate spatially, this could, for example, allow us to identify clusters. Therefore, we will visualise our tower sites coloured by the corresponding position (positive or negative) on the PC1 axes.

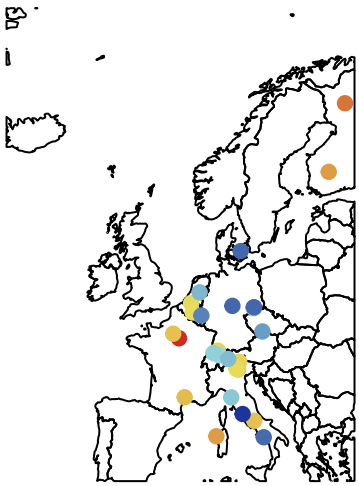
```
# start by designating trait and axis values to vector objects
tr1 <- df_pca[,1] # these are the traits
tr2 <- df_pca[,2] # these are the traits

pc1 <- traits_pca$x[,1] # these are the axis values
pc2 <- traits_pca$x[,2] # these are the axis values
```

```
# specify the colour palettes for positive and negative values
pospal <- colorRampPalette(c("#E9E15C", "#CF2B1E"))
negpal <- colorRampPalette(c("#1E3498", "#95D7DB"))

# we separate positive from negative values for visualization
neg_pc1 <- pc1
neg_pc1[neg_pc1 > 0 ] <- NA
pos_pc1 <- pc1
pos_pc1[pos_pc1 < 0 ] <- NA

datColpos <- pospal(100)[as.numeric(cut(pos_pc1,breaks = 100))]
```

```
datColneg <- negpal(100)[as.numeric(cut(neg_pc1,breaks = 100))]
```

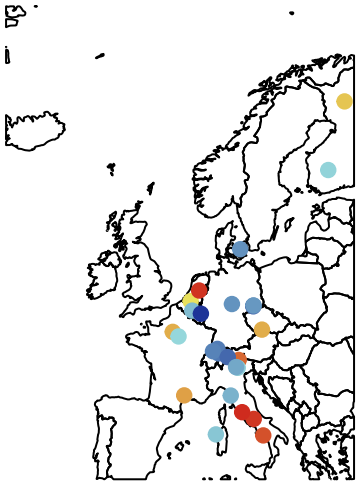
Once the colours are defined, we can plot PC1.

```
#PC1
plot(europe_shape)
points(df_sp_sites$lon, df_sp_sites$lat, col=datColpos,cex=1, pch = 19)
points(df_sp_sites$lon, df_sp_sites$lat, col=datColneg,cex=1, pch = 19)
```



And then repeat the process for PC2:

```
# PC2
neg_pc2 <- pc2
neg_pc2[neg_pc2 > 0 ] <- NA
pos_pc2 <- pc2
pos_pc2[pos_pc2 < 0 ] <- NA

datColpos <- pospal(100)[as.numeric(cut(pos_pc2,breaks = 100))]
datColneg <- negpal(100)[as.numeric(cut(neg_pc2,breaks = 100))]

# and plot it on the map
plot(europe_shape)
points(df_sp_sites$lon, df_sp_sites$lat, col=datColpos,cex=1, pch = 19)
points(df_sp_sites$lon, df_sp_sites$lat, col=datColneg,cex=1, pch = 19)
```

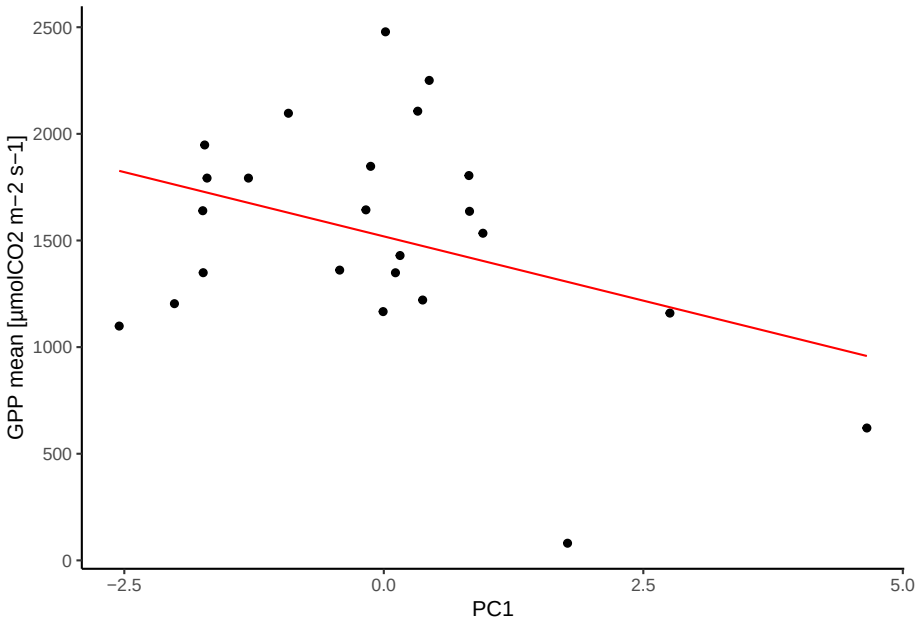


The last step in our trait data PCA is to compare the results with GPP. Since GPP has more sites than we used for this analysis we start by removing these sites.

```
GPP_pca <- GPP %>%
  filter(!(siteid %in% c("CH-Oe1", "DE-Obe", "FR-LBr", "IT-Cpz", "IT-Ro1")))
```

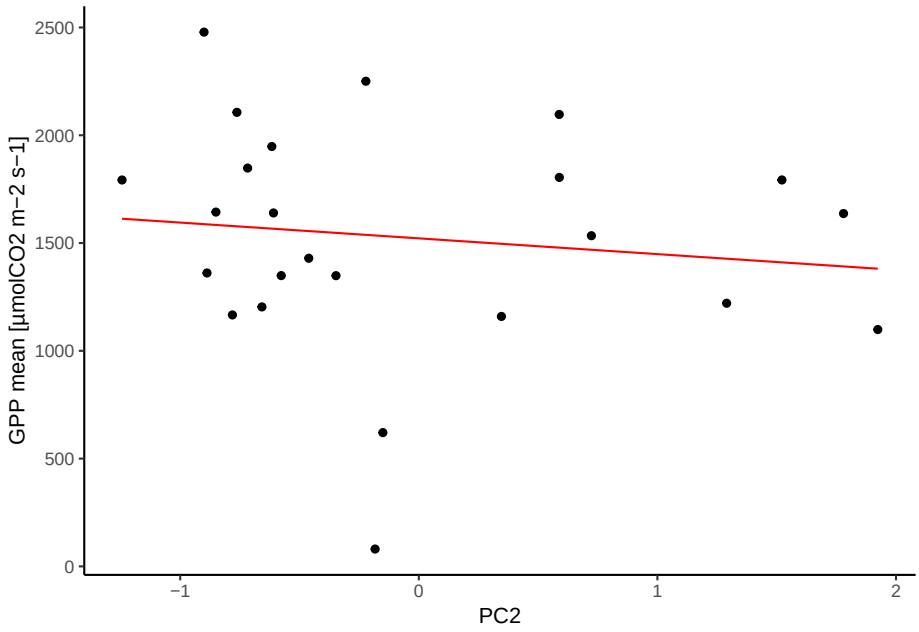
Then we make a linear model for PC1 and GPP and plot it.

```
reg_gpp_pc1 <- lm(gpp_meanann~pc1, data = GPP_pca)

GPP_pca %>%
  ggplot(aes(x = pc1, y = gpp_meanann)) +
  geom_point() +
  geom_smooth(method='lm', color="red", size=0.5, se=FALSE) +
  xlab("PC1") +
  ylab("GPP mean [µmolCO2 m-2 s-1]") +
  theme_classic()
```



We repeat this for GPP and PC2.

```
reg_gpp_pc2 <- lm(gpp_meanann~pc2, data = GPP_pca)

GPP_pca %>%
  ggplot(aes(x = pc2, y = gpp_meanann)) +
  geom_point() +
  geom_smooth(method='lm', color="red", size=0.5, se=FALSE) +
  xlab("PC2") +
  ylab("GPP mean [µmolCO2 m-2 s-1]") +
  theme_classic()
```



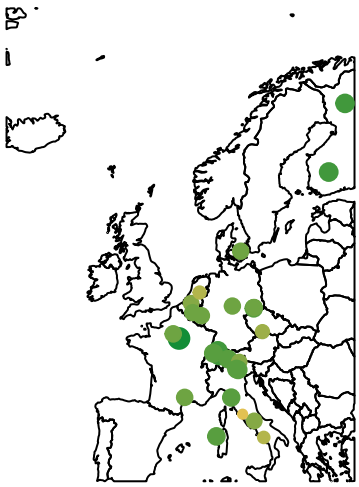
Once again we see weak correlations with GPP. Therefore, we can conclude that this further search for predictors that explain the variation in GPP has not yielded any good results. We will have to continue looking for better predictors, which explain this variation in GPP between the sites.

Checkpoint

Plot the spatial distribution of the trait 'leaf thickness', make the colour or size related to the leaf thickness values. Describe what you can gather from this plot.

Solution

```
leaf_t <- df_pca[,1]

leavepal <- colorRampPalette(c("#E2C151","#138D37"))

datColleaves <- leavepal(10)[as.numeric(cut(leaf_t,breaks = 10))]

plot(europe_shape)
points(df_sp_sites$lon, df_sp_sites$lat, col=datColleaves,cex=leaf_t/18, pch = 19)
```



The distribution of leaf thickness seems to depend on latitude. In Southern regions, leaves are generally thinner than in Northern regions. To confirm this we would have to make a model and test our observation.

Chapter 5

Data Scraping

5.1 Introduction

This chapter covers *data scraping* in R and yes, this is not a typo. *Scraping* means to gather or to extract whereas *scrapping* means to get rid of something. The goal of the sections below is to gather and convert online data into a structured format that can be easily accessed and modified in R. As a case study, we will scrape data from a website containing various information about fish species, called FishBase.

We will first learn how to access the website and how to extract useful useful information from plain text. Then, we will look at how to access a table that is embedded in the website and how to generate a new table in R which holds all information we want to extract. Next up is cleaning up the scraped data for visualization and analysis. At the end we will conduct a case study where we create a model to make predictions about fish species richness.

5.2 Theory

Let us first have a brief review of the main concept of the lecture. The web is the largest source of information and often free to access. In some cases the information is already presented in a nice format and can be easily copied into an excel spreadsheet. However, this is often not the case and data-sets are only presented in a format that is not easy to download or modify. Manually copying

data from different sub pages and sections is a tedious, slow and error prone approach. We therefore need a more automated and efficient technique to access such data.

Web scraping is a popular technique to extract information directly from a website by accessing its underlying HTML code. Scraping allows us to gather this unstructured data from many websites and put it all together in a ready-to-use data format. This structured data can then be further used as training, validation or test data sets for our machine learning algorithms.

Important Concepts of Websites

- **HTTP**: The **H**yper**t**ext **T**ransfer **P**rotocol is a widely used protocol for information systems. Hypertext documents include hyperlinks that can be clicked to access other resources. Put simply, HTTP is the foundation of how information, i.e., data, is communicated in the world wide web.
- **HTML**: **H**yper**t**ext **M**arkup **L**anguage is the coding language in which most websites are written. This includes elements like formatting or structuring and is often combined using CSS or JavaScript. HTML is organized using tags, which are surrounded by `< >`. Each HTML document is made of elements that are specified using tags. HTML elements and HTML tags are often confused. Tags are used to open and close the object, whereas elements include both tags and its content.

  Let's consider an example with the `<h1>` tag: `<h1> Title of the document </h1>` is an element, and `<h1>`, `</h1>` - are the tags enclosing it. The `<>` symbol is used to open a tag or an element and `</>` is used to close it. HTML documents have a hierarchical or tree like structure with different types of nodes as described in 5.1. The rectangular boxes are referred to as nodes. The *Text* node is also called as a child node of the *Element* node and also a leaf node as it only contains text and no links to further nodes. Both of the *Element* nodes that are attached to the *Root Element* node are called sibling nodes of the *Root Element* node. When we do web scraping we go through such a hierarchical structure to get the content (here text in the *Test* node). Keep in mind that every HTML document can vary with respect to its structure. This example is to just to provide you some knowledge about HTML document and its structure.
- **XML**: The **Ex**tensible **M**arkup **L**anguage has some similarities to HTML but the format is generally easier to read for machines. While HTML has a number of pre-defined tags, one can create ("extend") new tags as needed . This allows to define and control the meaning of the elements contained in a document or text.
- **API**: An **A**pplication **P**rogramming **I**nterface is a set of procedures and communication protocols that provide access to the data of an application, operating system or other services. Both APIs and web scraping are used

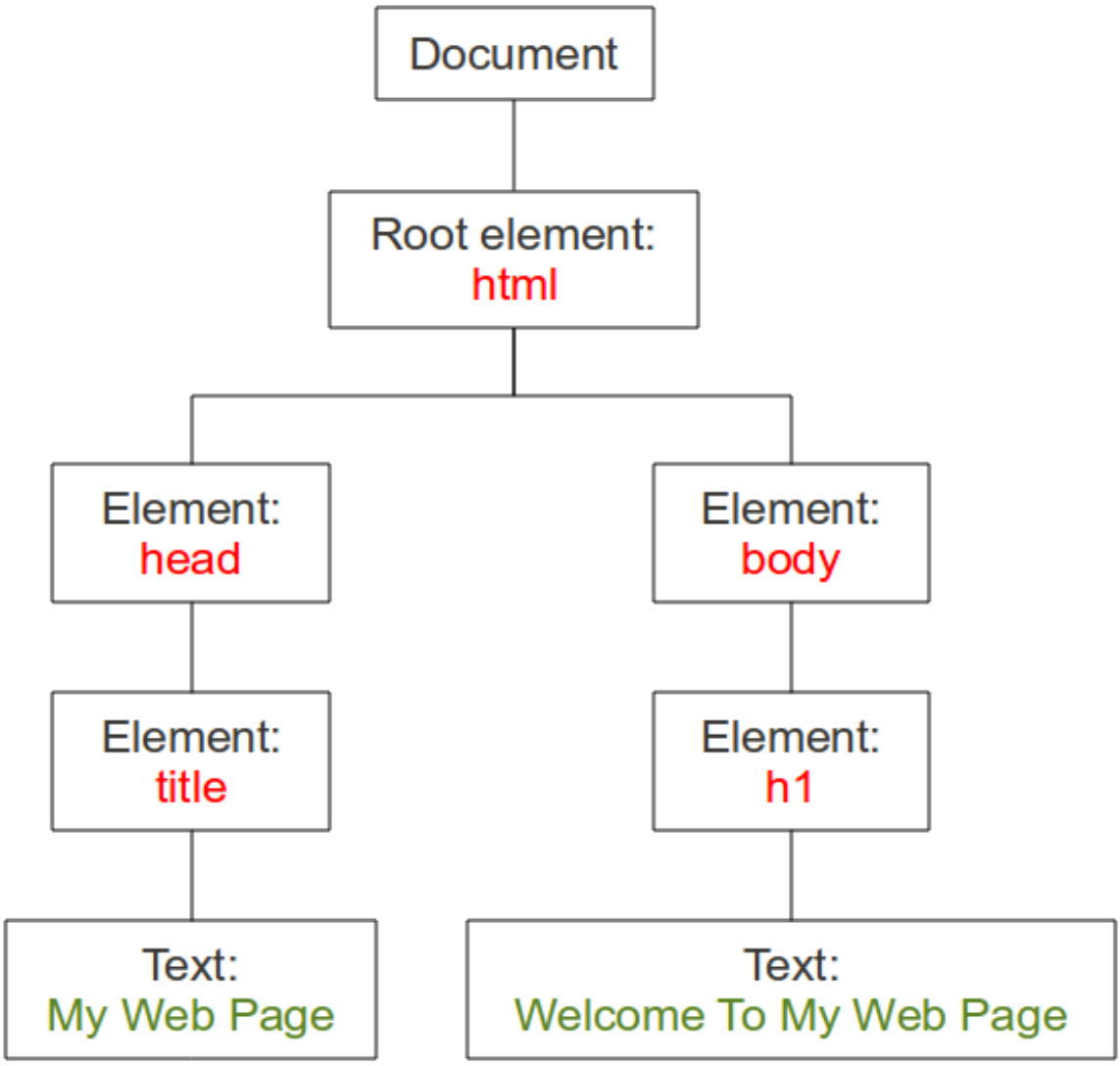


Figure 5.1: Visualization of the structure of an HTML document.

to retrieve data from websites, but their methodology differs substantially. APIs give us direct access to the data we would want, but they are limited to the corresponding website. As a result, we might find us in a scenario where there might not be an API to access the data we want. In these scenarios, web scrapping would allow us to access the data as long as it is available on a website. Hence APIs are very source/website specific and we can only do what is already implemented but in a clean fashion, while scrapping is more flexible and can be applied (nearly) everywhere but we have to handle all the formatting, inconsistencies, extraction, etc. by yourself.

Next, we will demonstrate the principles of web scraping using a simple case study. We will extract fish occurrence data from an online database and perform some basic correlations with the obtained data.

5.3 Web Scraping Applied

5.3.1 R-Packages and Functions

We will now load all packages necessary to perform web scraping on FishBase, namely `RCurl` and `XML`. `RCurl` provides functions to allow us to compose general HTTP requests and provides convenient functions to fetch URLs via get and post requests and process the results returned by the web server. `XML` give us approaches for both reading (get request) and creating (post request) XML and HTML documents, both locally and on the web via HTTP. Later in this tutorial, we will create some spatial visualization and load the packages `raster`, `sf` and `rgdal` to do so. Before we proceed further to accessing FishBase, we will have a quick look at the useful functions `lapply()` and `sapply()`.

```
lib_vec <- c("RCurl", "XML", "raster", "rgdal", "rfishbase", "tidyverse", "sf")
sapply(lib_vec, library, character.only = TRUE) # See below for how sapply() works
help(package = "XML") # Use this code line in the RStudio console to learn more about
```

```
help(package = "XML") # Use this code line in the RStudio console to learn more about
```

Sidenote: For data scraping in R one can also use 'rjson' to convert R objects into JSON objects and vice-versa. Another option we will use later are APIs. To get information about the loaded packages we can type the following command in the console:

lapply()

`lapply()` allows us to apply the same function to a vector of objects. The respective arguments are `X` (the vector or object we give as input) and `FUN` (the function which is applied to each element of `X`). The output of `lapply()` is a list of the same length as `X` where each element of `X` is the result of applying `FUN` to the corresponding element of `X`. `X` is a vector (atomic, list or data frame) or an expression object. The `l` in `lapply()` thus stands for list.

A simple example is to change the string value of a matrix to lower case with the R-function `tolower()` function. First, we set up a vector with string objects that we want to apply the function on. Then we create a pipe that feeds this vector into `lapply()` which applies `tolower`. The output is a list of the original two string objects but now written in lower case.

```
fish <- c("FAMILY","SPECIES")
fish %>% lapply(tolower) # Note that tolower() must not be written with brackets
```

```
## [[1]]
## [1] "family"
##
## [[2]]
## [1] "species"
```

sapply()

`sapply()` takes a list, vector or data frame as input and gives a vector or matrix as output. This is very useful if we have to use another function which only accepts a vector as input but does not accept a list. Note that we can apply the same function using `lapply()` or `sapply` but their outputs are not the same.

As an example we can have a look at the cars data where the speed and respective stopping distance are saved. We can apply the `min()` function to both features of the data set to get their minimal values. If we are using `lapply()`, we get a list that is accessible using `$`. With `sapply()` we get a named vector which can be seen at the headings `speed` and `dist` in the output below (those headings are of course the same as the variable names in the outputted list of `lapply()`).

```
(df <- head(cars))
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```
df %>% sapply(min)
```

```
## speed  dist
##     4     2
```

```
df %>% lapply(min)
```

```
## $speed
## [1] 4
##
## $dist
## [1] 2
```

5.3.2 The FishBase website

FishBase is a global species database of fish species. It provides data of various fish species, including information on taxonomy, geographical distribution, biometrics and morphology, behaviour and habitats, ecology and population dynamics as well as reproductive, metabolic and genetic data. Different tools, such as trophic pyramids, identification keys, biogeographical modelling and fishery statistics can be accessed on the website. Furthermore, direct species level links to information in other databases such as LarvalBase, GenBank, the IUCN Red List and the Catalog of Fishes exist. As of November 2018, FishBase included descriptions of 34,000 species and subspecies.

As shown in Figure 5.2, the website contains lots of information for all the different species and this information is stored in various different data types. A few examples of these data types are:

- Numbers in different formats and units (temperature ranges, latitudinal distribution)
- Text blocks (description of the distribution)
- Tables (fecundity, larvae information)
- Pictures and videos (of the species, embedded into HTML code)

For a better understanding of the following R code, you are strongly encouraged to have a look at the FishBase website in your browser. In the next sections you will learn how to deal with these different complex data types. A careful approach is required when extracting and downloading such information. Understanding the structure of the website is an important first step and it will save you coding time. So, always identify the relevant information you want to extract first and then write a suitable extraction code which fits the needs of your machine learning algorithm. *** ### Accessing FishBase

We are now ready to extract data from the *Coregonus lavaretus* website on FishBase. For this we first create a string object which holds the name of the

Coregonus lavaretus (Linnaeus, 1758)
European whitefish

Upload your photos and videos
Pictures | Videos | Stamps, Coins Misc. | Google image

Coregonus lavaretus
Picture by Østergaard, T.

Classification / Names Common names | Synonyms | Catalog of Fishes (gen.,

Actinopterygii (ray-finned fishes) > Salmoniformes (Salmons) > Salmonidae (Sal
Etymology: *Coregonus:* Greek, kore = pupils of the eye + Greek, gonia = angle (R
Linnaeus.

Environment: milieu / climate zone / depth range / distribution range

Freshwater; brackish; demersal; pH range: 7.0 - 7.5; dH range: 20 - ?; anadromous
4°C - 16°C (Ref. 2059); 73°N - 40°N

Distribution Countries | FAO areas | Ecosystems | Occurrences | P

Europe: Native to Lake Bourget (France) and Geneva (Switzerland, France). Popu
(France) apparently introduced, but a 'lavaret' had already been recorded from ther
59043). Other authors assume it to be a superspecies occurring in Great Britain an
Europe. Has been stocked into many other places in Europe outside its native rang
wrong scientific names for this species in use because of the problems in classifyi
7495). Appendix III of the Bern Convention (protected fauna). Asia: introduced to

Figure 5.2: Screenshot of the FishBase webpage for the species *Coregonus lavaretus*, a member of the family Salmonidae. As can be read in the distribution paragraph, it is widespread in freshwater systems from central and northwest Europe to Siberia.

fish profile we want to access. Using a variable to do this, adds flexibility in functions as will be shown below. If we want to get data about other species we can simply assign a new value to the object `x`with the desired species name.

```
x <- "Coregonus-lavaretus"
x
```

```
## [1] "Coregonus-lavaretus"
```

Next, we use the function `paste()` to create an object holding the URL to the fish profile. Attaching strings like this is called concatenate and will be used quite often in later tutorials, so remember this wording. To create the URL we do not have to add any separation between the arguments, so we use `sep = ""`. Using the flexibility of using `x` allows us here to add any species name directly into `paste()` instead of deleting and adding another Latin name every time.

```
url <- paste("http://www.fishbase.de/summary/", x ,".html",sep="")
url
```

```
## [1] "http://www.fishbase.de/summary/Coregonus-lavaretus.html"
```

For `url` we could also directly use *http://www.fishbase.de/summary/Coregonus-lavaretus* but then we would loose flexibility if we want to look for information about other species.

Checkpoint Try to do the same for the URL of other species by changing the code above.

Solution

```
# Change the object x_end to desired species name
x_ex<- "Salvelinus alpinus"

# Concatenate the URK
url_ex <- paste("http://www.fishbase.de/summary/",x_ex,".html",sep="")
```

To access the URL and its HTML documents, we will use various functions from the `XML` package. First, we want to get the URL content using `getURLContent()` which takes a URL as input argument. Then, we use `htmlParse()` to read the HTML document from the URL content into an R object. To get help on functions we can always use `help("htmlParse")`.

```
fishbase <- url %>% getURLContent(followlocation = TRUE) %>% htmlParse()
```

The HTML document saved in `fishbase` is quite long to show it entirely but a snapshot is displayed in Figure 5.3.

Somewhere in this gigantic object we can find the relevant information we are

```
## </h1>
##              <div class="smallSpace">
##              <span>
##                  Not Evaluated               </span>
##              </div>
## <br>
## </div>
##          <!--End Mod 10/16/17 Shelumiel Ortiz-->
##
##          <!-- Threat to humans -->
##          <div class="rlalign sonehalf">
##              <h1 class="slabel bottomBorder">
##                  Threat to humans  <br>
## </h1>
##              <div class="smallSpace">
##              <span>
##                  <span class="box" style="background-color:#3cb371"> </span>
##              </div>
##          </div>
## <br><br><br><br><br><!-- Start human uses--><p>
##              </p>
## <h1 class="slabel bottomBorder">
##                  Human uses                  </h1>
##              <div class="smallSpace">
##              <span>
##                  Fisheries: commercial; aquaculture: commercial; gamefish: y
##              </div>
```

Figure 5.3: Extract from the HTML document saved in the variable ‘fishbase‘.

looking for. For example, Figure 5.4 below shows the extract where the maximal lenght of the fish is documented. Check out the online profile to double check whether the value is correct.

```
##             <div class="smallSpace">
##             <span>
##                 Maturity: L<sub>m</sub><a href="../Reproduction/MaturityList.php?ID=232">27.1</a>, range
##             </div>
##
```

Figure 5.4: Extract of 'fishbase' documenting maximal length of *Coregonus lavaretus*.

Highlighted in red are the opening and closing tags of the element where the information on the maximal length is stored. Knowing these tags, we can identify where our information is stored and how to extract it. In this example, we need to target either the `span` or `\div` tags. In order to extract the information inside these two tags, we use the function `getNodeSet()`. This function finds XML nodes that match a particular criterion. `span` is used for a small chunk of HTML inside a line whereas `div` is used to group larger chunks of code.

```
fishbase_div <- fishbase %>% getNodeSet("//div ")
fishbase_span <- fishbase %>% getNodeSet("//span ")
```

In the next two Figures 5.5 and 5.6 we can see the differences of what is stored in variable `fishbase` and `fishbase_div`. The difference to `fishbase_span` is similar. We see that instead of having this one long unstructured extract in Figure 5.5, we have now a list of objects that we easily access. In Figure 5.6, the list can be identified by the `[[22]]` in the top left corner which marks the position of the object within the list. In short, `getNodeSet()` identifies all the sections for a given tag (i.e., nodes), separates them and gathers them in a list.

```
##             <div class="smallSpace">
##             <span>
##             Spawning takes place at night.                    </span>
##             </div>
```

Figure 5.5: 'fishbase' before the command 'getNodeSet()'

5.3.3 Scraping Numbers

Next, we now want to extract the maximal body length of *Coregonus lavaretus* out of the list of nodes we created in the previous section. For this, we will use the function `xmlValue()` which allows us to access the text in the nodes

```
## [[22]]
## <div class="smallSpace">&#13;
##               <span>&#13;
##               Spawning takes place at night.                    </span>&#13;
##               </div>
##
```

Figure 5.6: 'fishbase' after the command 'getNodeSet()'

by converting them into strings. To proceed further, we briefly have a look at regular expressions.

A *regular expression* is basically a sequence of characters (think of a certain word or a number) that can be searched for in a string (which is nothing else but a sequence of characters). The `regexec()` (for **reg**ular **ex**pression) function allows us to search for a pattern within a string. The output of `regexec()` gives you different information. For now it enough to know that it returns a list where the first object holds the position of the pattern in the string or simply a `-1` if the pattern is not found. In our case, we are going to look for the pattern "Max length" because after this pattern, the value we are looking for is saved (see Figure 5.7).

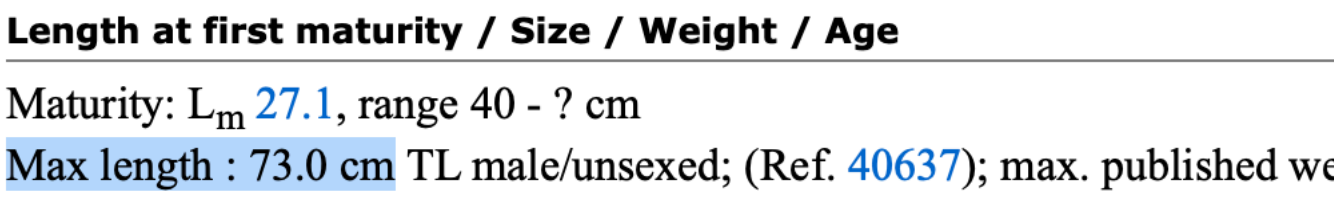


Figure 5.7: In blue highlighted is the regular expression on the FishBase website which we are looking for.

The code to identify the position where the regular expression *Max Length* is located is rather complex, so here is a step-by-step breakdown. You can run each part of the pipe below one-by-one to better understand what is happening.

```
pos <- fishbase_span %>% lapply(xmlValue) %>%  # 1. Convert all list objects in fishbase_span int
  regexec(pattern =" Max length") %>%          # 2. Search strings for pattern
  which.max()                                  # 3. Return position in list where highest number
                                               # (just one way to identify the node with relevant
pos
```

```
## [1] 42
```

Now we know that our regular expression *Max length* can be found in node number 42. To get access to the text in this node we simply access `fishbase_span`

via the list notation `[[ ]]` and turn the object into a string and save it as `fish_length`. As you can see in the output, the information on max length has been found correctly.

```
fish_length <- fishbase_span[[pos]] %>% xmlValue()
fish_length
```

```
## [1] "\r\n\t\t\t\t\tMaturity: Lm27.1, range 40 - ? cm Max length : 73.0 cm TL male/u
```

We now use the function `substr()`, which you already encountered in previous chapters. It allows to extract a section of a string that lies between two characters. Think of this as extracting a number interval in a series of numbers. Unfortuntaly, identifying this start and end has to be done somewhat by hand. In Figure 5.7 you can see that, starting from the *'M'* in *Max Length*, there are 13 characters until the value begins with a *7* and 16 characters until it ends with a *0*. Keep in mind that every blank space is also counted as character. Knowing this, we can first use `regexec()` to get the character position where *Max Length* starts and just add 13, respectively 16 to the value.

Two things to note here: (i) Keep in mind that every blank space is also counted. (ii) Note that the output of `regexec()` is a list where the first object is the character position where the pattern begins in the string. Thus, we need to access this position number by using `[[1]][1]`. Another way instead of using `regexec()` is just by typing in some numbers into `substr()` and find the relevant characters by trial and error.

```
start_M <- fish_length %>%  regexec(pattern= "Max length")
max_length <- fish_length %>% substr(start_M[[1]][1] + 13,
                                     start_M[[1]][1] + 16)
max_length
```

```
## [1] "73.0"
```

At last, we have to convert the value of the maximal length (which is a string of characters) into a number using the function `as.double()`. This converts the value into a machine readable format which makes things easier later. We see from the output below that we correctly exctracted a maximal length of *Coregonus lavaretus* of 73 cm. Now, we can start thinking about looping this approach to extract the maximal length of various species and collect them in a nicely formatted data frame.

To demonstrate fexibility in data reading: This entire approach can also be used to for example to identify any numeric value that is followed by any one or enclosed by any two characters.

```
max_length <- as.double(max_length)
max_length
```

```
## [1] 73
```

5.3.4 Scraping Text Snippetrs

International Union for Conservation of Nature

Another interesting information on FishBase is the IUCN (International Union for Conservation of Nature) status of each species. The IUCN Red List of threatened Species has evolved to become the world's most comprehensive information source on the global extinction risk status of animal, fungus and plant species.

The IUCN Red List is a critical indicator of world's biodiversity and has been established in 1994. It contains explicit criteria and categories to classify the conservation status of individual species on the basis of their probability of extinction. After a species is evaluated by the IUCN, it is placed into one of eight categories based on its current conservation status as shown in the figure.

Far more than a list of species and their status, it is a powerful tool to inform and catalyze action for biodiversity conservation and policy change, critical to protecting the natural resources we depend on for survival. The IUCN also provides information about the range, population size, habitat and ecology, use and/or trade, threats, and conservation actions that will help inform necessary conservation decisions.

Extracting IUCN Status

Now that we learned how to extract values, we can move on to extracting text snippets! In this part, we are going to get the IUCN Status of *Coregonus lavaretus*. In contrary to above, we are going to use the function `which()` (not `which.max()`) to find the position of the elements we are looking for. We use `regexec()` to search for matches of our pattern within each element of a character vector. In this case, the pattern is simply *IUCN*. As for the maximal length, we can look up this pattern on the website we are scraping from.

```
IUCN_pos <- which(                  # which() gives us the numbers of nodes where pattern was fou
  fishbase_div %>%                  # pipe from above to find pattern put into which()
    lapply(xmlValue) %>%
    regexec(pattern = "IUCN")       # end of pipe
  > 0                               # check for which() to only give positive numbers for nodes
)                                   # we do this because if pattern is not found, regexec returns

IUCN_pos                            # pattern can be found in nodes 5, 14 and 24
```

```
## [1]  5 14 24
```

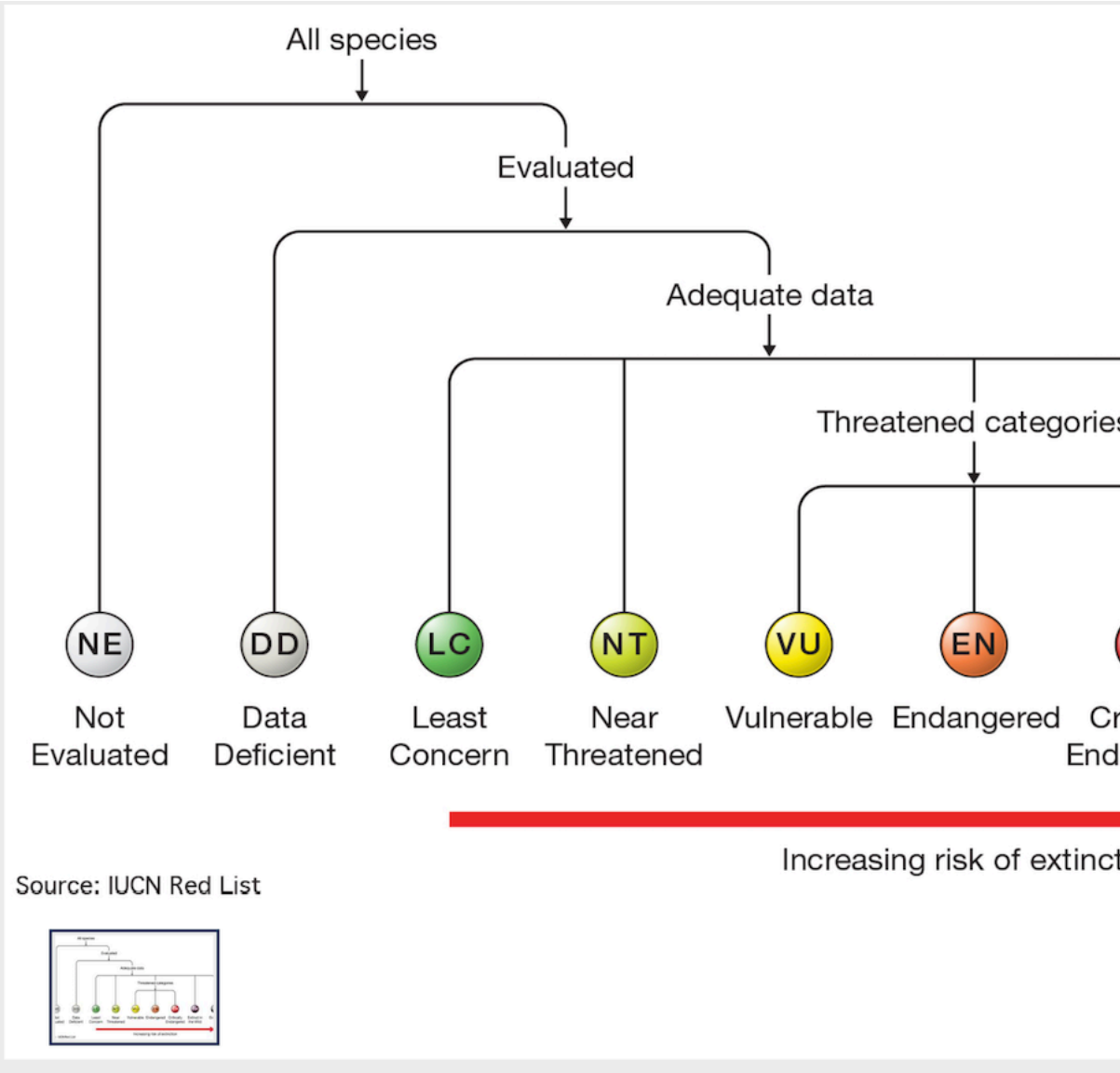


Figure 5.8: Structure of IUCN categories.

If the pattern cannot be found in `fishbase_div`, the variable `w_IUCN` will be empty (we use `fishbase_div` here because the pattern was not found in `fishbase_span`). To avoid complications later, we better do a quick check and set the variable to `NA` if it is empty and else use `xmlValue()` to get the value at the last node where the pattern was found.

```
if(length(IUCN_pos)==0){                # if which() from above returned 0 (no node with >0)
  IUCN_stat = "NA"                      # set status to NA
} else {                                # else
  IUCN_xml <- fishbase_div[[            # access fishbase_div nodes with [[
    IUCN_pos[length(IUCN_pos)]]] %>%    # access last node where pattern was found, length(IUCN_pos
    xmlValue()                          # feed node into xmlValue() to return readable string
}

IUCN_xml
```

```
## [1] "\r\n\t\t\t\t\r\n\t\t\t\t\tIUCN Red List Status   (Ref. 120744)\r\n\t\t\t\t\r\n\t\t\t\t\r\
```

The `IUCN_xml` looks a little confusing and we only need a tiny part of it, namely *VU*. We can see that in this node *VU* has a unique character pattern which does not appear twice: The closing bracket `)`is right after `U`, an alphabetical character. We can use this pattern to extract *VU*! Here, the `str_extract()` function from tidyverse comes in very handy. We can directly specify the pattern we are looking for and extract the substring from the original string. Note that for generalization, we need to specify `[[:alpha:]]` for any alphabetical character `+[)]` for the closing bracket.

However, if we do so, we still extract the bracket as well. To get rid of it, we can simply remove it by replacing it with "nothing" using `str_replace()`. Instead of specifying the closing bracket, we use `[:punct:]` which generalizes punctuations like `.` `,` `:` `;` `?` `!` etc. As we see in the output below, the IUCN status *VU* has been correctly extracted - nice! Plus, we don't need to convert it and can use it as string later on.

```
IUCN_sta <- IUCN_xml %>%
  str_extract(pattern = "[:alpha:]+[)]") %>%
  str_replace(pattern = "[:punct:]", replacement = "")

IUCN_sta
```

```
## [1] "VU"
```

5.3.5 Scraping Tables

The next step is to read a table from a website. Here, we are going to get information on the eggs of *Coregonus lavaretus*. Have a look at its profile.

In section **Life cycle and mating behavior** you can see that the table for information on eggs is a link and not a table directly. Foruntatley for us, we can use the function `getHTMLLinks()` to retrieve links within an HTML document or the collection of names of external files referenced in an HTML document. In Figure 5.9, an extract is shown of the list with more than 100 links found on the profile page.

We see that either we could directly access link number 100 or search for the substring *FishEggInfoSummary* to look for other links. In fact, we will find two links with this substring (see Figure 5.10. For this reason, we quickly check if the links are identical of if they lead to different pages. We can do this by using the logic operator `==`. The link comparison gives `TRUE` as output meaning that both links are completely identical and it does not matter which one we use. Let us save the first one in `egg_link`.

```
egg_link <- fishbase %>% getHTMLLinks() %>%
  str_subset(pattern = "FishEggInfoSummary")

egg_link[1] == egg_link[2]
```

```
## [1] TRUE
```

```
egg_link <- egg_link[1]
```

```
##   [99] "/Reproduction/SpawningList.php?ID=232&GenusName=C
##  [100] "../Reproduction/FishEggInfoSummary.php?ID=232&Gen
##  [101] "/Reproduction/FecundityList.php?ID=232&GenusName=
```

Figure 5.9: Extract of the list of links found on the FishBase profile of *Coregonus lavaretus*.

```
## [1] "../Reproduction/FishEggInfoSummary.php?ID=232&GenusNam
## [2] "../Reproduction/FishEggInfoSummary.php?ID=232&GenusNam
```

Figure 5.10: List of links that have the substring *FishEggInfoSummary*.

Checkpoint Call the variable *egg_link*, you will notice the link begins with '..' (see above 5.10). Try to remove these dots so that the link begins with '/Reproduction/..'. As a hint, you can use the function `str_replace()` as was done above. You will need this code later, so make sure that you run it.

```
# your code
```

Solution

```
egg_link <- egg_link %>% str_replace("..", "" )
egg_link
```

```
## [1] "/Reproduction/FishEggInfoSummary.php?ID=232&GenusName=Coregonus&SpeciesName=lavaretus&fc=
```

As you can see, this is no proper URL yet. Thus, we first need to create a working URL to access its content. Similarly to what we did previously, we do this using the function `getURLContent()`. Since we know that this link leads to a table, we can pipe the URL content into `readHTMLTable()`. The output here is a list and we have access the first object as done below.

```
egg_tab <- paste("http://www.fishbase.de/", egg_link, sep="") %>%
  getURLContent(followlocation=TRUE, .encoding="CE_UTF8") %>%
  readHTMLTable()

egg_tab <- egg_tab[[1]] # To save the list object as a data frame
egg_tab
```

```
##               Main Ref.
## 1  Place of Development
## 2          Shape of Egg
## 3            Attributes
## 4         Color of Eggs
## 5  Color of Oil Globule
## 6 Additional Characters
## 7    Get Information on Scirus
```

Now we can extract information from this table by using the function `which()`. We want to find the position in this table, where *Shape of Egg* is defined. To identify the right row, we look in the first column for the respective string and extract the value stored in the second column of this row. To automatize, we add a check to see whether any information was extracted at all from the table. If not (as it is the case here), we just set `egg_shape` to NA.

```
egg_shape <- egg_tab[                      # Accessing rows of egg_tab (remember: df[rows, cols])
  which(egg_tab[, 1] == "Shape of Egg"),   # Get number of row where "Shape of Egg" appears in the
  2]                                       # Access the second column

egg_shape
```

```
## [1] ""
```

```
if(egg_shape == "") {egg_shape = "NA"}
egg_shape
```

```
## [1] "NA"
```

Now we can put all the information into a data frame for the entry *Coregonus lavaretus* by using the function `tibble()`. Having done this entire data scraping for one species, we can start to automatize the process for multiple species and simply append the data to our data frame.

```
species_df <- tibble(Species = "Coregonus-lavaretus", Length = max_length, IUCN = IUCN_
species_df
```

```
## # A tibble: 1 x 4
##   Species             Length IUCN  Egg
##   <chr>                <dbl> <chr> <chr>
## 1 Coregonus-lavaretus     73 VU    NA
```

Checkpoint Try to create and add a new variable *egg_color* to the data frame and feed it with the respective information from our `egg_table` (follow the same process used for 'egg_shape').

```
# your code
```

Solution

```
egg_color = egg_tab[which(egg_tab[, 1] == "Color of Eggs"), 2]  # extract information
if(egg_color == ""){egg_color = "NA"}                           # set missing informat
species_df <- tibble(species_df, Color = egg_color)             # Add new variable to
species_df
```

```
## # A tibble: 1 x 5
##   Species             Length IUCN  Egg   Color
##   <chr>                <dbl> <chr> <chr> <chr>
## 1 Coregonus-lavaretus     73 VU    NA    NA
```

5.3.6 The Fishbase Package

Using an API

As we saw in the previous sections, web scraping can be tedious. In this subsection, we want to get data using an API which is a much easier way. We use the R package `rfishbase` to get data from https://www.fishbase.de. This package makes it very easy to look up for information on the most well-known fish species. It simplifies the data extraction process but also has some limits

as we will see below. You ca have a look at the functions built into `rfishbase` by typing `help(package = "rfishbase")` into your RStudio console or have a look at the RDocumentation. Most functions have straight forward names and have a sring as input which holds the Latin name of the wanted fish species.

Let us get some information on *Coregonus lavaretus* with this new package! We see that the super short code `species("Coregonus lavaretus")`provides us with 101 variables with entries for our fish species - think about the hand-written web scraping code this would require! To directly asses the maximal length of the species, we can directly use the `$` notation. As you might recall, this is the same length as we got above.

```
CL <- species("Coregonus lavaretus")
CL$Length
```

```
## [1] 73
```

In the next step we are interested to get information about the diet of *Coregonus lavaretus* by using the function `diet_items()`. It allows us to access the table on the food items of the chosen species. To extract information on food, we can save the diet items in a table and use the tidyverse notation plus the respective variables *FoodI*, *FoodII*:

```
food <- diet_items("Coregonus lavaretus") %>% as_tibble
food %>% dplyr::select(FoodI, FoodII) %>% head() # We're using dplyr:: here to avoid package conf
```

```
## # A tibble: 6 x 2
##   FoodI      FoodII
##   <chr>      <chr>
## 1 zoobenthos mollusks
## 2 nekton     finfish
## 3 zoobenthos benth. crust.
## 4 zoobenthos worms
## 5 zoobenthos benth. crust.
## 6 zoobenthos benth. crust.
```

Checkpoint Now your next task is to get information on predators (hint: the function is exactly named like that).

```
# your code
```

Solution

```
# use function 'predators()' for Coregonus lavaretus
pre <- predators("Coregonus lavaretus")
pre %>% dplyr::select(PredatorName) %>% head()
```

```
## # A tibble: 6 x 1
```

```
##   PredatorName
##   <chr>
## 1 Coregonus peled
## 2 Esox lucius
## 3 Salmo trutta trutta
## 4 Sander lucioperca
## 5 Phryganea sp.
## 6 Coregonus lavaretus
```

Limits of APIs

One of the major limitations of an API is that we can only use already implemented functions. For example, in our case, we cannot get the IUCN Status of a given species because it is not built into `rfishbase`. In order to obtain the IUCN Status we must use another API, namely the package `rredlist`. Unfortunately, to have access to the *rredlist* API we need an authentication key which we only get for a small fee. As it is with other things in life, either you try to put in the work yourself or you can pay someone to do so (except for the amazing open source community).

5.3.7 Summary

- In this third section we learned how to extract data from a website.
- We saw that this can be done either with web scraping or using APIs.
- We saw that web scraping can be tedious, but we learned about some limitations of APIs.

5.4 Case Study: Species Richness and Red List species proportions

Please note that the code below is written in base R and not tidyverse which makes the codes a bit more difficult to read. But do not worry about this and simply just enjoy the beauty of what R is capable of.

Next, we look at another case study on species richness and red list species proportions. To proceed with the case study we need to prepare our dataset. First, we need to get a list of species for the dataset. In this subsection we will see how to get all the species in a given family.

5.4.1 Creating List of Species

In this section, we want to get all the species of the family of *Salmonidae*. As we did above, for flexibility we will create an object x with the name of the family and use `paste()` to get the link. We then use `getURLContent()` to get the content of the link `url` and save it in `con_sal`.

```
x <- "Salmonidae"
url <- paste("http://www.fishbase.de/Nomenclature/FamilySearchList.php?Family=", x, sep="")
con_sal <- getURLContent(url, followlocation = TRUE)
```

Next we create a dataframe using `data.frame()` and get a list of variables. We will extract the variables with the same number of rows and unique row names. The function `readHTMLTable()` (from earlier) helps to extract data from HTML tables in an HTML document. Then we can extract the species from the given family with `as.character()`. We use `z[,1]` to get the first column which is the column with the scientific names of species. Now we can print the first element of the column with the scientific names.

```
df <- data.frame(readHTMLTable(con_sal))
sp_per_family <- as.character(df[,1])
sp_per_family[1]
```

```
## [1] "Brachymystax lenok"
```

Using `str_replace()` function we can substitute the empty space between the Genus and the Species with a `"-"`. Finally we can print the first element of *sp_per_family* again.

```
sp_per_family <- str_replace(sp_per_family, " ", "-")
sp_per_family[1]
```

```
## [1] "Brachymystax-lenok"
```

5.4.2 Extracting IUCN Status for all species

In this section, we are going to get the IUCN Status for a given List of species using a 'for loop'. We will extract the IUCN Status of all the species from the Netherlands. First, we are going to upload the dataset containing the list of the species and some other data that is going to be useful for the next sections. We will do that by using the function `read_csv()`. This data is taken from this nature article by cropping it to Western Europe. As we only need data related to Netherlands, we will use the function `grep()` and pass *Netherlands* as an argument.

```
dataset <- read_csv("./data/dataset2.csv")
subset <- dataset[grep("Netherlands", dataset$Country),]
```

Let us first briefly discuss how to construct a 'for loop', since it's been a while since you used it in previous chapters. To get the IUCN Status of a list of species we always change the value of x (the species) and run the code, but if we have to do that for many species it will be very long and tedious. In this case 'for loops' come in handy. In a 'for loop' the variable x runs over the vector (here each species) and returns the IUCN Status. Before getting the IUCN status we will go over an easy example, such as printing the integers from 1 to 10 using a 'for loop'. In this case, we iterate over the vector 1:10.

```
for(j in 1:10) {
print(j) # this prints the value of j for that given loop
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

Now we can make a 'for loop' in order to get the IUCN status of all the species in the above subset (Netherlands) of the initial dataset. We are going to iterate over the column of the dataset with the valid FishBase species names. The code lines inside the loop are exactly a copy-paste of what we had for the *Coregonus lavaretus*, but in this case, we have to look for other species. In the last line of the code below, we created a new column in the data frame *subset* in order to save the IUCN Status. The 'i' in `subset$IUCN[i]` is used to save the IUCN status of the species we are iterating over in the 'for loop'. It will save the results of the species one by one.

```
i <- 0
for(x in subset$X6.Fishbase.Valid.Species.Name) {
  i <- i + 1

  url <- paste("http://www.fishbase.de/summary/",x,".html",sep="")        # we call the
  fish_species <- htmlParse(getURLContent(url, followlocation=TRUE))      # get the con
  fish_species_div <-getNodeSet(fish_species, "//div ")                   # get the nod
  w_IUCN  <- which(sapply(lapply(fish_species_div,xmlValue),function(x)   # look for th
    {regexec(pattern="IUCN", x)[[1]][1]})>0)
```

```
  if(length(w_IUCN)==0){                                          # NA if no IUCN status
    IUCN_status="NA"
} else {                                                          # else access informatio
    d1_IUCN  <- xmlValue(fish_species_div[[w_IUCN[length(w_IUCN)]]])
    IUCN <- unlist(regmatches(d1_IUCN,gregexpr(pattern= "[[:alpha:]]+)",
    d1_IUCN)))
    IUCN_status <- sub(pattern="[[:punct:]]",replacement="",IUCN[1] )
}

print(IUCN_status)
subset$IUCN[i] <- IUCN_status # make a new column in 'subset' containing the IUCN status
}
```

```
## [1] "LC"
## [1] "LC"
## [1] "LC"
## [1] "LC"
## [1] "LC"
## [1] "LC"
## [1] "VU"
## [1] NA
## [1] NA
```

We can see which IUCN statuses are present in Netherlands by using the `unique()` function. There are LC for least concern, VU for vulnerable and of course NA for unknown values.

```
unique(subset$IUCN)
```

```
## [1] "LC" "VU" NA
```

5.4.3 Proportion of species in Netherlands

We will now plot the proportion of species in each category for the Netherlands. So let us calculate the number of species in each category (from above we just have 2 outputs,* *LC* & *NT*). In general in other countries, we also have other IUCN Status, for example, VU. In this section, we will focus only on the two listed statuses. Using the function `length()` we can obtain the number of species in each category.

```
number_lc <- length(which(subset$IUCN == "LC"))
number_vu <- length(which(subset$IUCN == "VU"))
number_na <- length(which(is.na(subset$IUCN)))
```

```
# print the values for NT, VU and NA
paste0("LC:", " ", number_lc)
```

```
## [1] "LC: 6"
```

```
paste0("VU:", " ", number_vu)
```

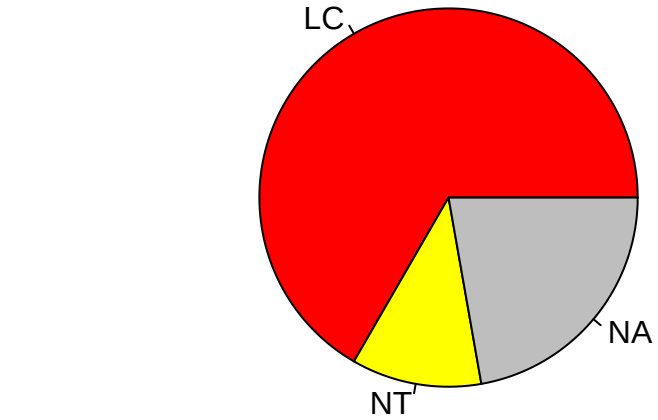
```
## [1] "VU: 1"
```

```
paste0("NA:", " ", number_na)
```

```
## [1] "NA: 2"
```

We are ready to plot this as pie chart.

```
slices <- c(number_lc, number_vu, number_na)
lbls <- c("LC","NT", "NA")
pie(slices, labels = lbls, font.main = 1,
main = "Proportion of species per IUCN Status in Netherlands", col=c("red", "yellow", "
```



*** ### Cleaning data with IUCN Status

In this part of the tutorial, we have to clean the data that we will use for the correlations. We have provided the dataset already with the IUCN status since the code needs a lot of time to run, but the procedure is exactly the same as we did in the previous sections. So, let us load the dataset, it's called *datasetIUCN*.

In the previous sections, we saw that for some species we did not have information on the IUCN status. In these cases, we set the IUCN status as NA. We can check for possible NA values by calling the `unique()` function.

```
dataset_IUCN <- read_csv("./data/datasetIUCN.csv")
unique(dataset_IUCN$IUCN)
```

```
##  [1] "LC" "CR" "lc" "VU" NA   "EX" "NT" "DD" "EN" "EW"
```

Checkpoint Now your task is to remove the row with the IUCN status as NA. You can use the function `subset()` to get the subset of the dataset with information about the IUCN status.

```
# your code
```

Solution

```
# subset the data without NAs (!=NA; not equal to NA), so this effectively removes NAs
dataset_IUCN_NA <- subset(dataset_IUCN, dataset_IUCN$IUCN != "NA")

# check if it worked
unique(dataset_IUCN_NA$IUCN)
```

```
## [1] "LC" "CR" "lc" "VU" "EX" "NT" "DD" "EN" "EW"
```

Next, we will load the shapefile of glacial basins using the package `rgdal` and the function `readOGR()`. We fetch the data from different fish basins across Europe in the `basin_shapefile` from the data stored in the basins folder. In order to plot the basins on the map we will use `fortify()` function on the `basin_shapefile`. This function helps to convert a lines-and-points object into a data frame for ggplot. We will store this dataframe as `fort_basin`.

```
basin_shapefile <- readOGR("./data/basins")
```

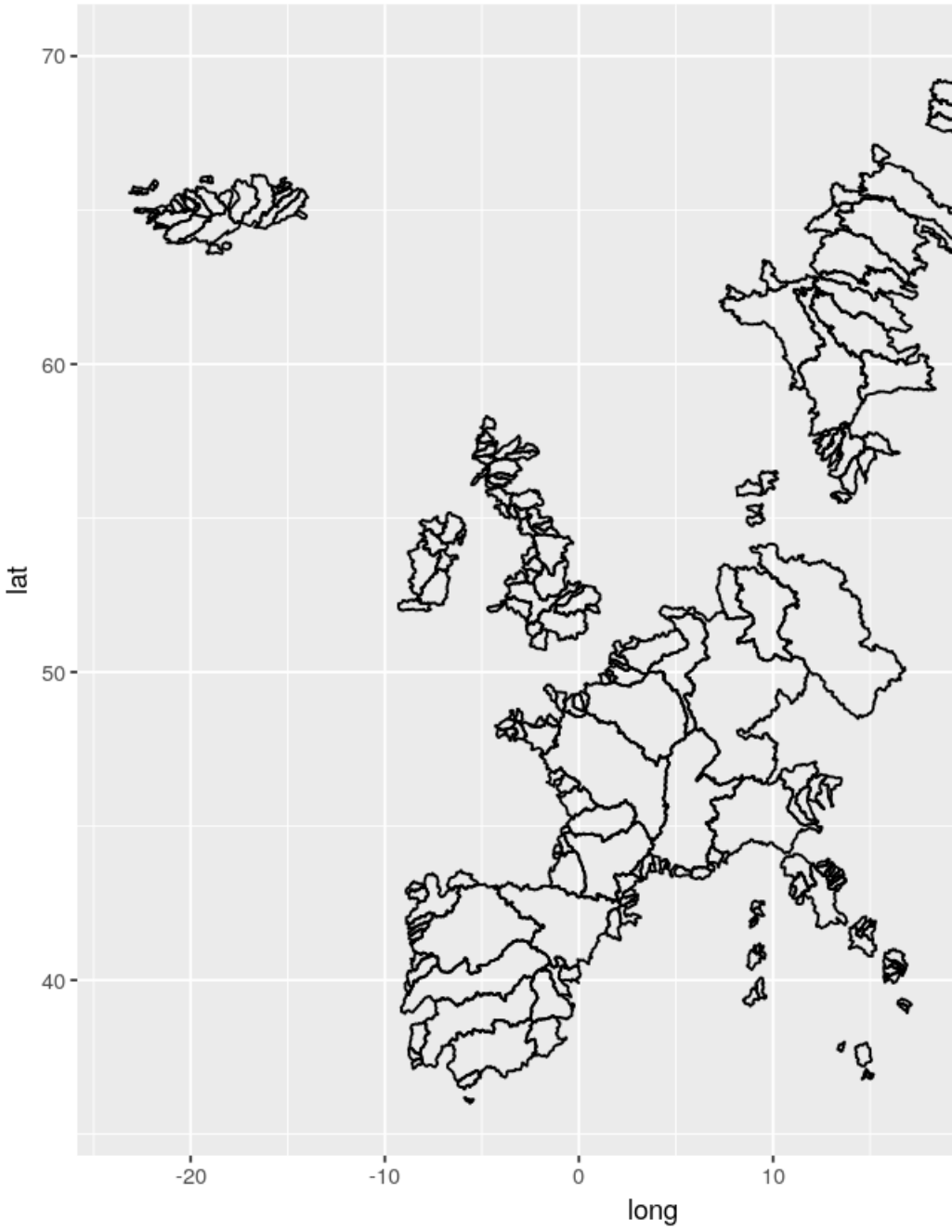
```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/pascalschneider/Polybox/Shared/Data Science Lecture Planning - shared folder/4
## with 3119 features
## It has 9 fields
```

```
fort_basin <- fortify(basin_shapefile)
head(fort_basin)
```

```
##        long       lat order  hole piece id group
## 1 -43.00000 -22.55000     1 FALSE     1  0   0.1
## 2 -43.03750 -22.69583     2 FALSE     1  0   0.1
## 3 -43.05325 -22.69523     3 FALSE     1  0   0.1
## 4 -43.05507 -22.68816     4 FALSE     1  0   0.1
## 5 -43.06888 -22.68693     5 FALSE     1  0   0.1
## 6 -43.07279 -22.68390     6 FALSE     1  0   0.1
```

We can visualise our `basin_shapefile` using `ggplot()`.

```
ggplot() + geom_polygon(data = fort_basin, aes(x = long, y = lat, group = group), colour = "black
```



\begin{figure}
\caption{Plot of `fort_basin`.} \end{figure}

5.4.4 Maps of species richness and Red List species proportions

Preparing data

We now want to map the species richness and Red List species proportions. Let us first calculate the species richness and proportion of Red List species per basin. Species richness is the number of species pro basin and the proportion of red list species is the ratio between the number of species in the red list and the number of species pro basin. We will iterate over the vector with the basin names and we will use the function `nrow()` to find the number of occurrences.

```
for (x in as.character(basin_shapefile$BasinName)) {

  # we now restrict to dataset to the basin x
  dataset_basins <- dataset_IUCN[dataset_IUCN$X1.Basin.Name==x,]

  # number of species in the given country x
  n1 <- nrow(dataset_basins)

  # we now restrict to dataset to the country x and Red List
  dataset_c_IUCN <- dataset_basins[grep("VU|EN|EX|EW|CR", dataset_basins$IUCN),]

  # number of species in the given country x with given IUCN Status
  n2 <- nrow(dataset_c_IUCN)

  # compute the proportion
  basin_shapefile$proportion[basin_shapefile$BasinName==x] <- n2/n1
  basin_shapefile$richness[basin_shapefile$BasinName==x] <- n1
}
```

We can see the newly computed data for the columns *'proportion'* and *'richness'* with respect to each basin in the `basin_shapefile`.

```
head(basin_shapefile@data)
```

```
##       BasinName   Country Ecoregion Endorheic Out_Longit  Out_Latit Med_Longit
## 0  Cachoeirinha    Brazil Neotropic      <NA>  -43.10344 -22.693658  -43.06899
## 1      Comprido    Brazil Neotropic      <NA>  -47.07734 -24.451571  -47.15300
## 2 Arroyo.Walker Argentina Neotropic      <NA>  -62.29922 -40.628898  -62.59625
## 3     Aconcagua     Chile Neotropic      <NA>  -71.53604 -32.912822  -70.65645
## 4        Amazon    Brazil Neotropic      <NA>  -52.23409  -1.619426  -64.57286
## 5      Andalien     Chile Neotropic      <NA>  -73.09136 -36.664541  -72.81968
##    Med_Latit    Surf_area proportion richness
## 0 -22.595111     228.8151        NaN        0
## 1 -24.465831     204.7977        NaN        0
## 2 -40.507344     969.1561        NaN        0
```

```
## 3 -32.770645    7318.8768        NaN        0
## 4  -6.714857 5888416.9156        NaN        0
## 5 -36.824925     767.4691        NaN        0
```

Next we get the map of Europe. We will read the data in `continent_shapefile` and then will extract the continent 'Europe' map. If you call the variable `europe` you will see that the dataframe is empty. This is because it extracts only the map data which we can simply plot by passing the europe as an argument in `plot()` function. We will do it in the upcoming code cells.

```
continents <- readOGR('./data/continent_shapefile')
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/pascalschneider/Polybox/Shared/Data Science Lecture Planning - share
## with 8 features
## It has 1 fields
```

```
europe <- continents[continents$CONTINENT == 'Europe',]
```

```
Warning message in readOGR("../data/continent_shapefile"):
"First layer europe_map read; multiple layers present in
/work/04_data_scraping/data/continent_shapefile, check layers with ogrListLayers()"


OGR data source with driver: ESRI Shapefile
Source: "/work/04_data_scraping/data/continent_shapefile", layer: "europe_map"
with 53 features
It has 94 fields
```

Plotting data

So now let us plot the proportions of Red List species. First, we will create a new column *'proportion_colour'* and in this column, we will store the colours. Then we will break the proportions into 10 different parts by grouping the values in the proportion column into 10 using the `cut()` function. Then we get rid of the index vector using `as.numeric()` and get all the values as numeric values. We used the `rev()` function to reverse the colours, red colour indicates the species that are getting distinct and have very less proportion pro basin and yellow indicates the species with comparatively more proportion in the basin. Finally, we store these colours to the column proportion_colour. The colour indicates the proportion of Red List species occurring in the corresponding basin. We do the same for the species richness.

```
basin_shapefile$proportion_colour <- rev(heat.colors(11))[as.numeric(cut(basin_shapefil
```

Now we create a new column in the *fort_basin* dataframe and map the values of *'proportion_colour'* into the new color column. We are doing this to have the colour (species proportion divided into 10 parts) and lat-long values in one

datframe which helps to plot the graph using `ggplot()`. Remember, above we extracted the continent *'europe'* from the `continent_shapefile`, here we will use `fortiy()` on the europe dataset to plot it. In detail, we will plot the `fort_europe` and `fort_basin` data on the map.

```
fort_basin$color <- fort_basin$id   # create a new column color

for (i in as.numeric(unique(fort_basin$id))){   # map the values into color column by id
   fort_basin$color[fort_basin$id == i] <- basin_shapefile@data$proportion_colour[i+1]
}

fort_europe <- fortify(europe)
```

```
ggplot(fort_basin, aes(x = long, y = lat, group = group)) +
  geom_polygon(data = fort_europe, aes(x = long, y = lat, group = group), colour = 'white') +
  geom_polygon(fill = fort_basin$color, colour = "black")+
  xlim(-25, 28)
```

From the Figure 5.11, we see that the proportion of Red List species is highest in South-Western Europe. The aim of the Red List is to inform decision-makers about potentially endangered species, i.e. species whose population size has been rapidly declining during the last decades or the species that only occur in small numbers at present. The proportion of species on the Red List of each region, therefore, gives an indication of the risk of species going extinct in a region and represents an important tool for conservation strategies.

Repeat for mapping species richness

We will repeat the above steps all together for plotting the species richness on map.

```
# break the richness  of the species into 10 parts and then we assign colors to each part
basin_shapefile$richness_colour <- rev(heat.colors(11))[as.numeric(cut(basin_shapefile$richness,

fort_basin$rich_color <- fort_basin$id    # create a new column rich_color

# mapping the values from basin_shapefile to fort_basin

for (i in as.numeric(unique(fort_basin$id)))
{
   fort_basin$rich_color[fort_basin$id == i] <- basin_shapefile@data$richness_colour[i+1]
}

# plot
ggplot(fort_basin, aes(x = long, y = lat, group = group)) +
  geom_polygon(data = fort_europe, aes(x = long, y = lat, group = group), colour = 'white') +
```

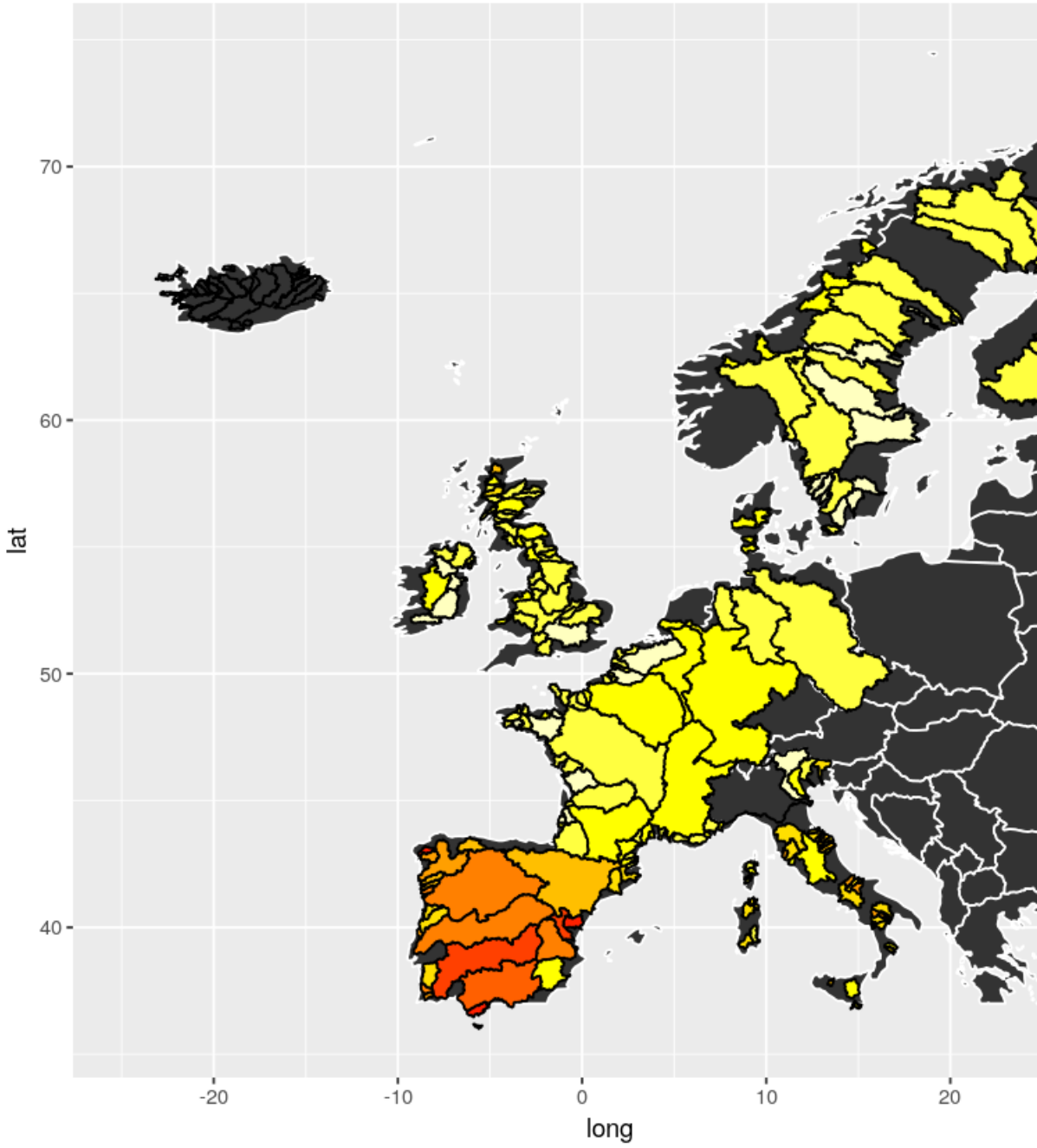


Figure 5.11: Proportions of Red List species in Europe.

```
  geom_polygon(fill = fort_basin$rich_color, colour = "black")+
  xlim(-25, 28)
```

5.4.5 Relation of basin size and species richness

In the previous section, you observed the spatial patterns of fish diversity across Europe. In this section, we will try to explain these patterns. To achieve this, we will correlate the fish species richness of each basin to the surface area of the corresponding area to see how the species richness varies with respect to the surface area of the basin. We begin with plotting a simple scatterplot. We will log transform the data we have on surface area as it helps to make data conform to normality and also helps to deal with the outliers and skeweness in the data. Then we will plot this data against species richness. Since the data-deficient basins show zero observations in the dataframe, we will remove those first.

Now we will create a new dataframe with the *surface area* and *richness*. We'll rename the columns as *'Basin_area'* and *'Species_richness'* respectively. We make a simple scatterplot with a regression line to visualise the relationship.

```
basin_shapefile <- basin_shapefile[basin_shapefile$richness!=0,]

bs_sr <- tibble(basin_shapefile@data$Surf_area, basin_shapefile@data$richness )
names(bs_sr)[1]<- 'Basin_area'
names(bs_sr)[2]<- 'Species_richness'

ggplot(bs_sr, aes(x = log(Basin_area), y = Species_richness)) +
geom_point() +
geom_smooth(method='lm', color="red", size=0.5, se=FALSE)+
xlab("Basin Area") +
ylab("Species Richness") +
theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

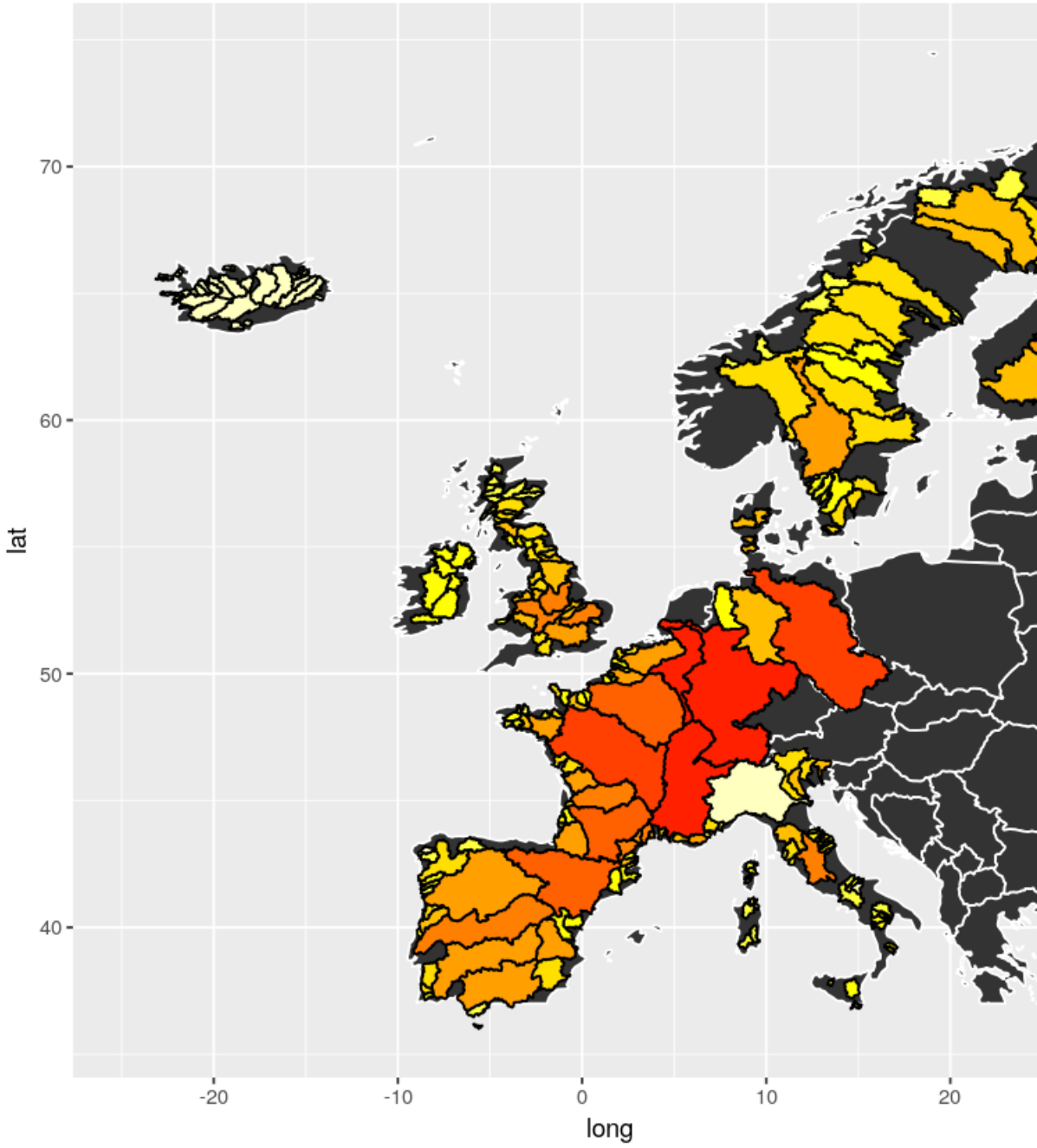
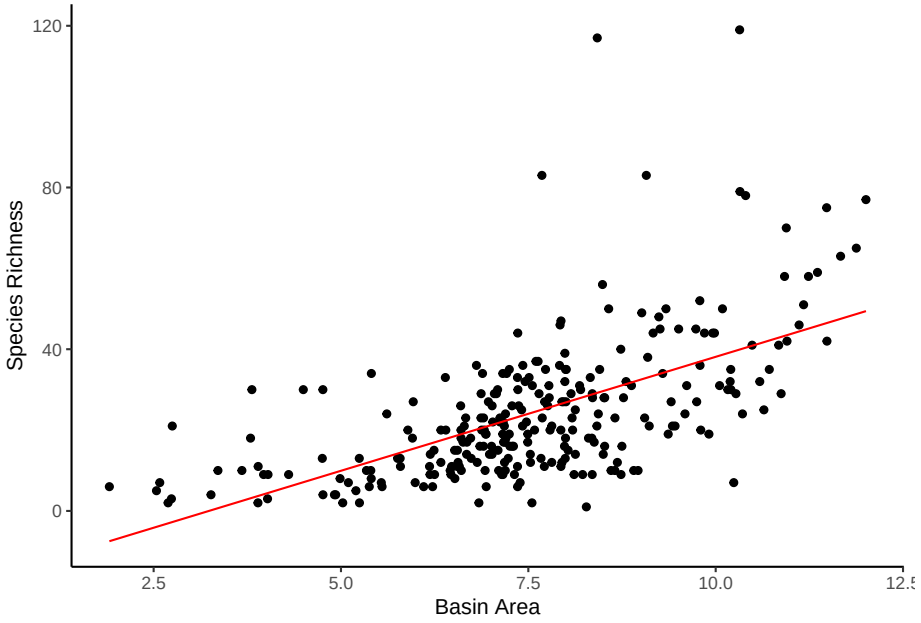


Figure 5.12: Species richness in Europe.



The plot shows a positive correlation between basin area and fish richness, meaning that we expect to see a higher richness in larger basins. This pattern is commonly observed in ecology and one explanation for this is that larger areas provide different habitat types (niches), which allows more species to co-exist.

As a next step, we will create a simple model of this relationship. This allows us to make predictions on the richness of fish species in other basins based on the basin area.

```
richness_model <- lm(richness~log(Surf_area), data=basin_shapefile@data) #create a linear model
```

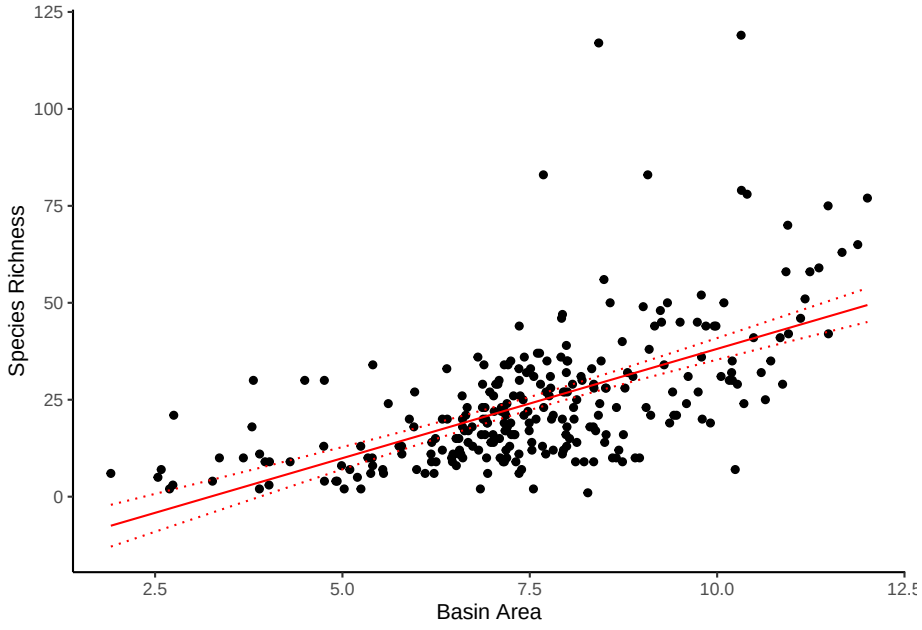
We now want to calculate the confidence intervals of the model to have a better idea of the uncertainty of the model. We will then coerce the basin surface, the model fit and the confidence interval into a new data frame called *model_df*.

```
#calculate the confidence intervals
model_df <- as.data.frame(cbind(log(basin_shapefile$Surf_area), predict(richness_model, interval=
names(model_df)[1] <- 'log_area'
head(model_df)
```

```
##     log_area      fit      lwr      upr
## 174 8.274449 28.39050 26.57731 30.20369
## 291 9.960755 37.89576 35.13808 40.65343
## 335 7.949240 26.55739 24.82776 28.28701
## 340 5.953703 15.30907 13.07901 17.53914
## 392 9.074369 32.89944 30.71907 35.07981
## 393 5.538369 12.96794 10.47659 15.45930
```

The final step is to plot the model fit and prediction intervals over the data. To get nice lines in the plot, the *model_df* dataframe needs to be ordered by the basin area first.

```
ggplot(bs_sr, aes(x = log(Basin_area), y = Species_richness)) +
geom_point() +
geom_line(aes(x = model_df$log_area, y= model_df$fit ), color = "red")+
geom_line(aes(x = model_df$log_area, y= model_df$lwr ), color = "red", linetype = "dott
geom_line(aes(x = model_df$log_area, y= model_df$upr ), color = "red", linetype = "dott
xlab("Basin Area") +
ylab("Species Richness") +
theme_classic()
```

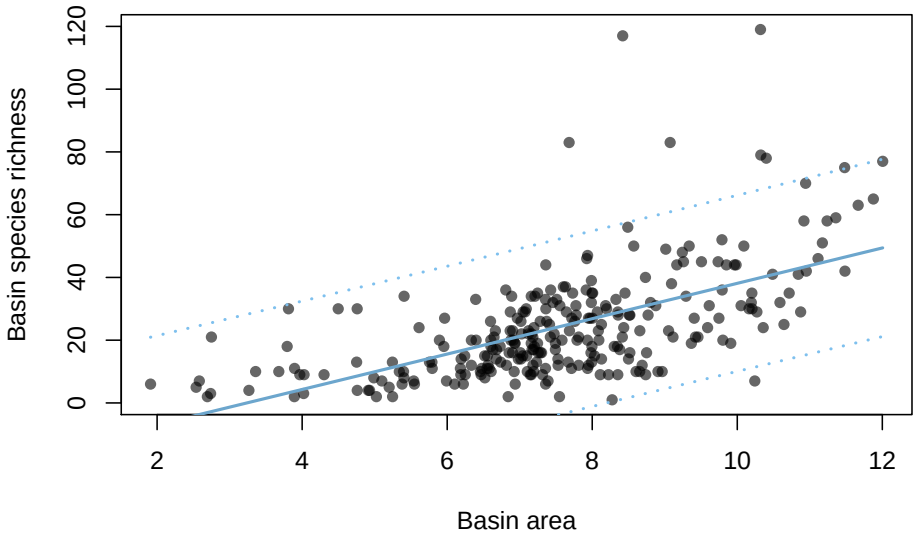


Checkpoint You have now seen how you can calculate and plot the confidence interval of your data. Based on your model you can also create a so-called **prediction interval** which gives an estimate of the range in which the model will most likely predict the y-values (for a given x-value). In our case the prediction interval would indicate in which range the model would expect the fish richness to be for a given basin surface area. Do you think that these prediction intervals will be broader or narrower than the confidence interval? Try to write the code for calculating and plotting the prediction intervals by yourself.

```
# your code
```

Solution

```
#prediction intervals

# make a dataframe of area and richness model predictions
model_df_prediction <- as.data.frame(cbind(log(basin_shapefile$Surf_area), predict(richness_model
names(model_df_prediction)[1] <- 'log_area'

# order the basins by size
model_df_prediction <- model_df_prediction[order(model_df_prediction$log_area),]

# plot the model
plot(log(basin_shapefile$Surf_area), basin_shapefile$richness, pch=16, cex=1, col=rgb(0,0,0,0.6),
lines(model_df_prediction$log_area, model_df_prediction$fit, col='skyblue3', lwd=2)
lines(model_df_prediction$log_area, model_df_prediction$lwr, col='skyblue2', lwd=2, lty=3)
lines(model_df_prediction$log_area, model_df_prediction$upr, col='skyblue2', lwd=2, lty=3)
```



5.5 References

- Automated Data Collection with R, S. Munzert, C. Rubba, P. Meißner and D. Nyhuis
- XML and Web Technologies for Data Sciences with R, D. Nolan, D. Temple Lang
- http://www.columbia.edu/~cjd11/charles_dimaggio/DIRE/styled-4/styled-6/code-13/
- https://ourcodingclub.github.io/tutorials/webscraping/

- https://www.earthdatascience.org/courses/earth-analytics/get-data-using-apis/use-twitter-api-r/

5.6 Exercise

For this week's exercise open up the Rstudio environment. Remember to save all your changes to this notebook using git status, git add , git commit -m "your comment", git push.

Today's exercise is about getting data from the web and extracting useful insights from it.

Get in touch with your teaching assistant if you have any further questions.

Chapter 6

Catch-up

TBC

Chapter 7

Supervised machine learning basics I

7.1 Introduction

7.1.1 Learning objectives

After this learning unit, you will be able to …

- Differentiate machine learning from classical programming
- Describe the different variants of machine learning
- Conceptualize model training as an optimization problem
- Describe overfitting and how it can be measured (training vs. validation error).
- Formulate a model in R.
- Discuss why, when, and how to pre-process data.
- Measure and minimize loss for regression and classification (video)
- Describe the fundamentals of gradient descent (video)

7.1.2 Important points from the lecture

Machine learning refers to a class of algorithms that automatically generate statistical models of data.

There are two main types of machine learning:

Unsupervised machine learning: A class of algorithms that automatically detect patterns in data without using labels or 'learning without a teacher'. Examples are: PCAs, clustering, autoencoders, self-organizing maps, etc.

Supervised machine learning: A class of algorithms that automatically learn an input-output function based on example input-output pairs.
Examples include: support vector machines, random forests, decision trees, neural networks, etc. Supervised machine learning requires three ingredients: (1) Input data (2) Output data (3) A measure of model performance (a.k.a. "loss")

Loss is a concept central to many supervised machine learning algorithms. It measures how well our predicted model values fit the actual observed model values, a higher value indicates a higher loss, essentially meaning the model fits less well. Ideally, loss should be minimised. Loss can be used to update our model parameters in the next iteration of model training, this is called learning.

Supervised machine learning be used for regression (predict a continuous label) or classification (predict a categorical label).

Essentially, supervised machine learning is an optimization problem, which searches a vast and high dimensional solution space.

7.2 Tutorial

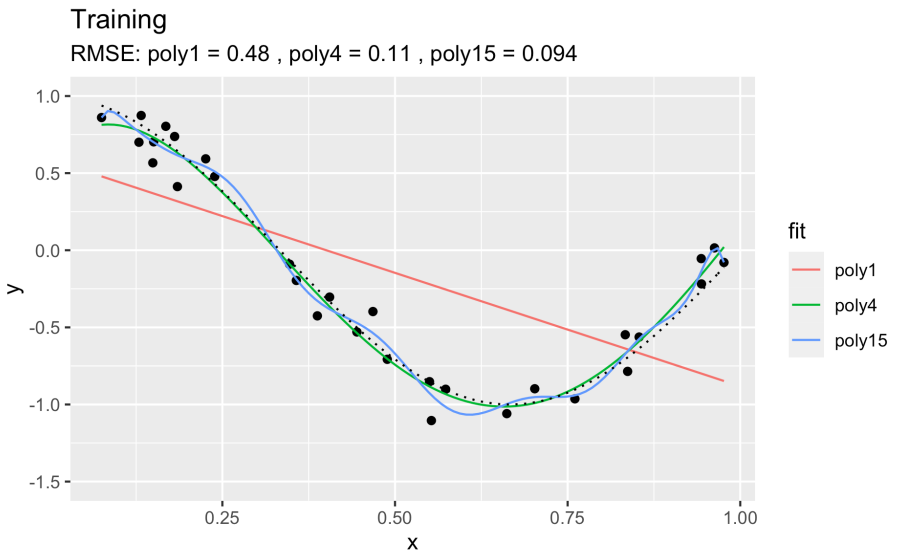
Machine learning (ML) may appear magical. The ability of ML algorithms to detect patterns and make predictions is fascinating. However, several challenges have to be met in the process of formulating, training, and evaluating the models. In this practical we will discuss some basics of supervised ML and how to achieve best predictive results.

In general, the aim of supervised ML is to find a model $\hat{Y} = f(X)$ that is *trained* (calibrated) using observed relationships between a set of *features* (also known as *predictors*, or *labels*, or *independent variables*) X and the *target* variable Y. Note, that Y is observed. The hat on $\hat{Y}$ denotes an estimate. Some algorithms can even handle predictions of multiple target variables simultaneously (e.g., neural networks). ML algorithms consist of (more or less) flexible mathematical models with a certain structure and set of parameters. At the simple extreme end of the model spectrum is the univariate linear regression. You may not want to call this a ML algorithm because there is no iterative learning involved. Nevertheless, also univariate linear regression provides a prediction $\hat{Y} = f(X)$, just like other (proper) ML algorithms do. The functional form of a linear regression is not particularly flexible (just a straight line for the best fit between predictors and targets) and it has only two parameters (slope and intercept). At the other extreme end are, for example, deep neural networks. They are extremely flexible, can learn highly non-linear relationships and deal with interactions between a large number of predictors. They also contain very large numbers of parameters, typically on the order of thousands. You can imagine that this allows these types of algorithms to very effectively learn from the data, but also bears the risk of *overfitting*.

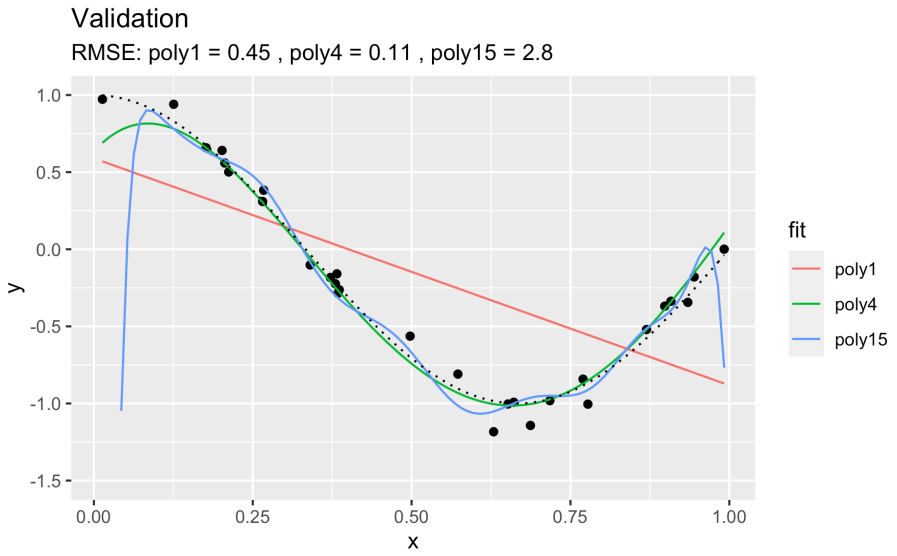
What is overfitting? The following example illustrates it. Let's assume that there is some true underlying relationship between a predictor x and the target variable y. We don't know this relationship (in the code below, this is `true_fun()`) and the observations contain a (normally distributed) error (`y = true_fun(x) + 0.1 * rnorm(n_samples)`). Based on our training data (`df_train`), we fit three polynomial models of degree 1, 4, and 15 to the observations. A polynomial of degree N is given by:

$$y = \sum_{n=0}^{N} a_n x^n$$

a_n are the coefficients, i.e., model parameters. The goal of the training is to get the coefficients a_n. From the above definition, the polynomial of degree 15 has 16 parameters, while the polynomial of degree 1 has two parameters (and corresponds to a simple linear regression). You can imagine that the polynomial of degree 15 is much more flexible and should thus yield the closest fit to the training data. This is indeed the case.

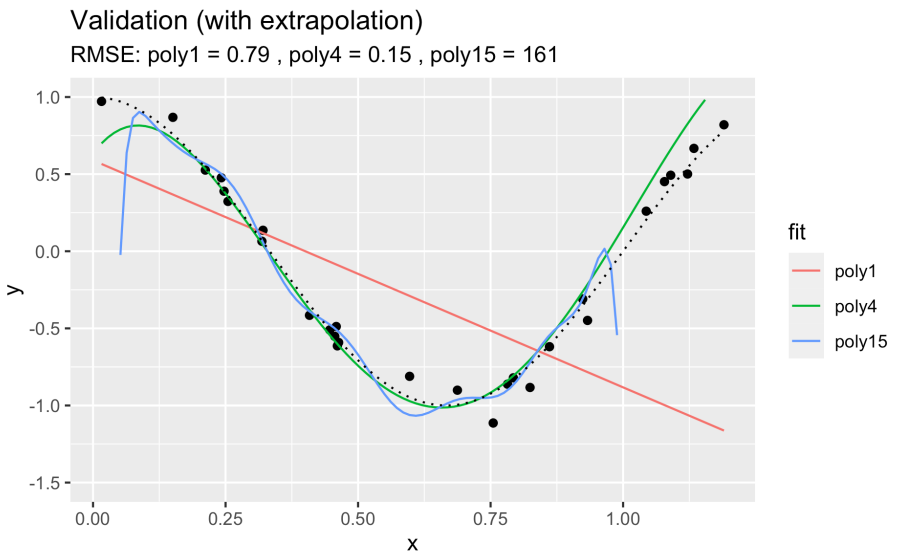


We can use the same fitted models on unseen data - the *validation data*. This is what's done below. Again, the same true underlying relationship is used, but we sample a new set of data points in x and add a new sample of errors on top of the true relationship.



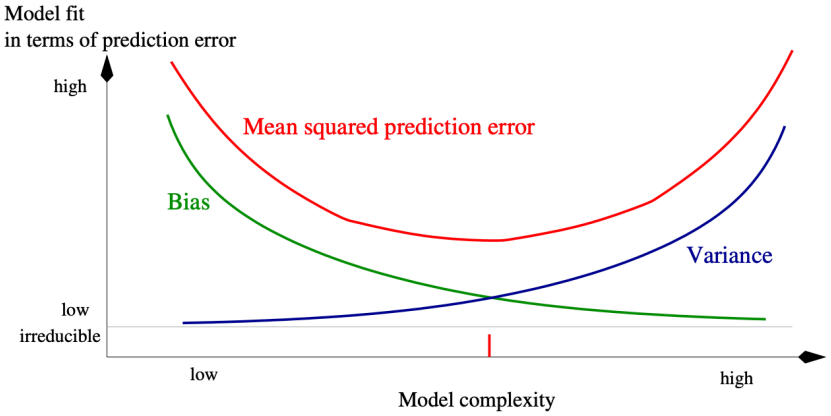
You see that, using the validation set, we find that "poly4" actually performs the best - it has a much lower RMSE that "poly15". Apparently, "poly15" was overfitted. Apparently, it indeed used its flexibility to fit not only the shape of the true underlying relationship, but also the observation errors on top of it. This has obviously the implication that, when this model is used to make predictions for data that was not used for training (calibration), it will yield misguided predictions that are affected by the errors in the training set. In the above pictures we can also conclude that "poly1" was underfitted.

It gets even worse when applying the fitted polynomial models to data that extends beyond the range in x that was used for model training. Here, we're extending just 20% to the right.



You see that the RMSE for "poly15" literally explodes. The model is hopelessly overfitted and completely useless for prediction, although it looked like it fit the data best when we considered at the training results. This is a fundamental challenge in ML - finding the model with the best *generalisability*. That is, a model that not only fits the training data well, but also performs well on unseen data.

The phenomenon of fitting/overfitting as a function of the model "flexibility" is also referred to as *bias vs. variance trade-off*. The bias describes how well a model matches the training set (average error). A model with low bias will match the data set closely and vice versa. The variance describes how much a model changes when you train it using different portions of your data set. "poly15" has a high variance, but much of its variance is the result of misled training on observation errors. On the other extreme, "poly1" has a high bias. It's not affected by the noise in observations, but its predictions are also far off the observations. In ML, we are challenged to balance this trade-off. In the next figure you can see a schematic illustration of the bias–variance trade-off.



This chapter introduces the methods to achieve the best model generalisability and find the sweet spot between high bias and high variance. The steps to get there include the preprocessing of data, splitting the data into training and testing sets, and model training that "steers" the model towards what is considered a good model fit in terms of its generalisation power.

You have learned in video 6a about the basic setup of supervised ML, with input data containing the features (or predictors) X, predicted ($\hat{Y}$) and observed target values (Y, also known as *labels*). In video 6b (title 6c: loss and it's minimization), you learned about the loss function which quantifies the agreement between Y and $\hat{Y}$ and defines the objective of the model training. Here, you'll learn how all of this can be implemented in R. Depending on your application or research question, it may also be of interest to evaluate the relationships embodied in $f(X)$ or to quantify the importance of different predictors in our model. This is referred to as *model interpretation* and is introduced in the respectively named subsection. Finally, we'll get into *feature selection* in the next Application session.

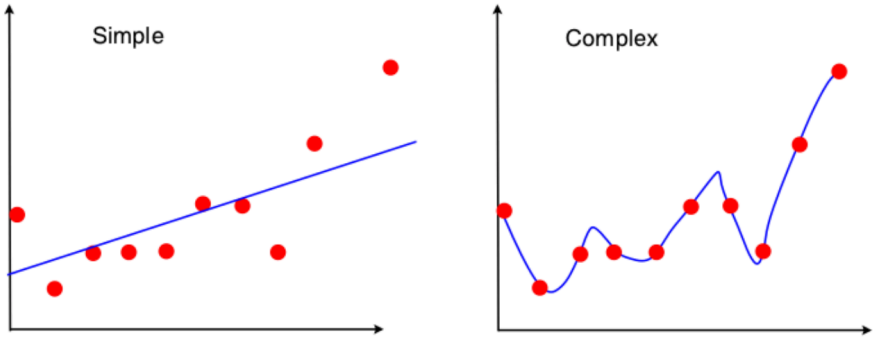
The topic of supervised machine learning methods covers enough material to fill two sessions. Therefore, we split this part in two. Model training, implementing the an entire modelling workflow, model evaluation and interpretation will be covered in the next session's tutorial (Supervised Machine Learning Methods II).

Of course, a plethora of algorithms exist that do the job of $Y = f(X)$. Each of them has its own strengths and limitations. It is beyond the scope of this course to introduce a larger number of ML algorithms. Subsequent sessions will focus primarily on Artificial Neural Networks (ANN) - a type of ML algorithm that has gained popularity for its capacity to efficiently learn patterns in large data sets. For illustration purposes in this and the next chapter, we will briefly introduce two simple alternative "ML" methods, linear regression and K-nearest-neighbors. They have quite different characteristics and are therefore great for

illustration purposes in this chapter.

Checkpoint

What can you say about the bias and the variance of the following graphs?



Solution

In the righthand figure we can see a low bias and a high variance. While in the left figure we see a high bias and a low variance.

We can now load some packages that we will use in this tutorial.

```
library(tidyverse)
library(ggplot2)
library(modelr)
library(forcats)
library(yardstick)
library(recipes)
library(caret)
library(broom)
```

7.2.1 Linear regression

Theory

In its simplest form, the univariate linear regression, we assume a linear relationship between X and Y:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad i = 1, 2, ...n \ ,$$

where Y_i is the i-th observation of the target variable, and X_i is the i-th value of the (single) predictor variable. The errors ϵ_i are assumed to be independent from each other (no autocorrelation), normally distributed, have mean of zero and a constant variance. β_0 and β_1 are constant coefficients (model parameters).

Fitting a linear regression is finding the values for β_0 and β_1 so that the sum of the square errors is minimized, that is:

$$\sum_i \epsilon_i^2 = \sum_i (Y_i - \beta_0 - \beta_1 X_i)^2 = \min.$$

Since the expected value of ϵ is zero (because it's normally distributed with mean zero), predictions of a linear regression model are obtained by $Y_{\text{new}} = \beta_0 + \beta_1 X_{\text{new}}$.

It's not hard to imagine that the univariate linear regression can be generalized to a multivariate linear regression, where we assume that the target variable is a linear combination of p predictor variables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ ... \ + \beta_p X_p + \epsilon \ .$$

Note that here, X, Y, and ϵ are vectors of length corresponding to the number of observations in our data set (n - as above). Analogously, calibrating the p coefficients $\beta_1, \beta_2, ..., \beta_p$ is to minimize the sum of square errors $\sum_i \epsilon_i^2$. While the regression is a line in two-dimensional space for the univariate case, it is a plane in three-dimensional space for bi-variate regression, and so on.

Implementation

To fit a univariate linear regression model in R, we can use the `lm()` function. Already in Chapter 2, we made linear models by doing:

```
ddf_ch_lae <- read_csv("./data/ddf_ch_lae.csv") # loads 'ddf_ch_lae'

df <- ddf_ch_lae %>%
  dplyr::select(-NEE_VUT_REF_QC, -TIMESTAMP) %>%  # not numeric features
  drop_na()
```

```
linmod1 <- lm(GPP_NT_VUT_REF ~ PPFD_IN, data = df)
```

Here, `GPP_NT_VUT_REF` is Y, and `PPFD_IN` is X. We can include multiple predictors for a multivariate regression, for example as:

```
linmod2 <- lm(GPP_NT_VUT_REF ~ PPFD_IN + VPD_F + TA_F, data = df)
```

or all available features in our data set (all columns other than `GPP_NT_VUT_REF` in `df`) as:

```
linmod3 <- lm(GPP_NT_VUT_REF ~ ., data = df)
```

`linmod` is now a model object of class `"lm"`. It is a list containing the following components:

```
ls(linmod1)
```

```
##  [1] "assign"        "call"          "coefficients"  "df.residual"
```

```
##  [5] "effects"       "fitted.values" "model"         "qr"
##  [9] "rank"          "residuals"     "terms"         "xlevels"
```

Enter `?lm` in the console for a complete documentation of these components.

R offers a set of generic functions that work with this type of object. The following returns a human-readable report of the fit.

```
summary(linmod1)
```

```
##
## Call:
## lm(formula = GPP_NT_VUT_REF ~ PPFD_IN, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1392  -1.9090  -0.1295   1.7981  14.9231
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.9201659  0.0813886   11.31   <2e-16 ***
## PPFD_IN     0.0144372  0.0002275   63.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.929 on 3631 degrees of freedom
## Multiple R-squared:  0.5258, Adjusted R-squared:  0.5257
## F-statistic:  4027 on 1 and 3631 DF,  p-value: < 2.2e-16
```

We can also extract coefficients β with

```
coef(linmod1)
```

```
## (Intercept)     PPFD_IN
##   0.9201659   0.0144372
```

Under the assumption of normally distributed errors ϵ with mean zero and constant variance σ^2, the magnitude of the variance can be estimated by

$$\hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^{n} \epsilon_i^2$$

- $\hat{\sigma}^2$ is also referred to as the *mean square error* (MSE), and its root, - $\hat{\sigma}$ is the *root mean square error* (RMSE).

The RMSE can be extracted from the linear regression model object by:

```
sigma(linmod1)
```

```
## [1] 2.929148
```

The RMSE is also reported in the output of `summary()` as the `Residual standard error`. The MSE can be calculated accordingly as:

```
sigma(linmod1)^2
```

```
## [1] 8.579907
```

Although `summary()` provides a nice, human-readable output, you may find it unpractical to work with. A set of relevant quantities are returned in a tidy format using `tidy()` from the broom package:

```
library(broom)
tidy(linmod1)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   0.920   0.0814        11.3 3.75e-29
## 2 PPFD_IN       0.0144  0.000228      63.5 0.
```

Model advantages and concerns

An advantage of linear regression is that the coefficients provide information that is straight-forward to interpret. We've seen above, that `GPP_NT_VUT_REF` increases by 0.014 for a unit increase in `PPFD_IN`. Of course, the units of the coefficients depend on the units of `GPP_NT_VUT_REF` and `PPFD_IN`. This has the advantage that the data does not need to be normalised. That is, a linear regression model with the same predictive skills can be found, irrespective of whether `GPP_NT_VUT_REF` is given in g $Cm^{-2}s^{-1}$ or in kg $Cm^{-2}s^{-1}$.

Another advantage of linear regression is that it's much less prone to overfit than other algorithms. We've seen this in the overfitting example above. But this can also be a disadvantage. Indeed, we found that the linear model "poly1" is rather under-fitting. It's not able to capture non-linearities in the observed relationship and exhibits a poorer performance than "poly4" also on the validation data set.

A further limitation is that least squares regression requires $n > p$. In words: the number of observations must be greater than the number of predictors. If this is not given, one can resort to step-wise forward regression, where predictors are sequentially added based on which predictor adds the most additional information at each step. You'll encounter stepwise regression in the Application session 8.

When multiple predictors are linearly correlated, then linear regression cannot discern individual effects and individual predictors may appear statistically insignificant when they would be significant if covarying predicotrs were not included in the model. Such instability can get propagated to predictions. Again, stepwise regression can be used to remedy this problem. However, when one predictor covaries with multiple other predictors, this may not work. For many applications in environmental sciences, we deal with limited numbers of predic-

tors. We can use our own knowledge to examine potentially problematic covariations and make an informed pre-selection rather than throwing all predictors we can possibly think of at our models. Such a pre-selection can be guided by the model performance on a validation data set (more on that below).

An alternative strategy is to use *dimension reduction* methods. Principal Component regression reduces the data to capture only the complementary axes along which our data varies and therefore collapses covarying predictors into a single one that represents their common axis of variation. Partial Least Squares regression works similarly but modifies the principal components so that they are maximally correlated to the target variable. You can read more on their implementation in R here.

Checkpoint

Considering the univariate linear regression model `GPP_NT_VUT_REF ~ PPFD_IN`, how do coefficients β_0 and β_1 change when you add 1 to all values `GPP_NT_VUT_REF`?

Solution

```
lm(GPP_NT_VUT_REF ~ PPFD_IN, data = df %>% mutate(GPP_NT_VUT_REF = GPP_NT_VUT_REF + 1))
```

```
##
## Call:
## lm(formula = GPP_NT_VUT_REF ~ PPFD_IN, data = df %>% mutate(GPP_NT_VUT_REF = GPP_NT_VUT_REF +
##     1))
##
## Coefficients:
## (Intercept)      PPFD_IN
##     1.92017      0.01444
```

```
lm(GPP_NT_VUT_REF ~ PPFD_IN, data = df)
```

```
##
## Call:
## lm(formula = GPP_NT_VUT_REF ~ PPFD_IN, data = df)
##
## Coefficients:
## (Intercept)      PPFD_IN
##     0.92017      0.01444
```

Of course, this simply corresponds to an upward shift of all points by 1, and thefore an upward shift of the regression line by 1. The slope β_1 doesn't change, but the y-axis intercept β_0 increases by 1.

7.2.2 K-nearest neighbours

As the name suggests, the K-nearest neighbour (KNN) uses the k observations that are "nearest" to the new record for which we want to make a prediction. It then calculates their average (in regression) or most frequent value (in classification) as the prediction. "Nearest" is determined by some distance metric evaluated based on the values of the predictors. In our example (`GPP_NT_VUT_REF ~ .`), KNN would determine the k days where conditions, given by our set of predictors, were most similar (nearest) to the day for which we seek a prediction. Then, it calculates the prediction as the average (mean) GPP value of these days. Determining "nearest" neighbors is commonly based on either the Euclidean or Manhattan distances between two data points x_a and x_b, considering all p predictors j.

$$\text{Euclidean: } \sqrt{\sum_{j=1}^{p}(x_{a,j} - x_{b,j})^2} \text{Manhattan: } \sum_{j=1}^{p}|x_{a,j} - x_{b,j}|$$

In two-dimensional space, the Euclidean distance measures the length of a straight line between two points (remember Pythagoras!). The Manhattan distance is called this way because it measures the distance you would have to walk to get from point a to point b in Manhattan, New York, where you cannot cut corners but have to follow a rectangular grid of streets. $|x|$ is the positive value of x ($|-x| = x$).

KNN is a simple algorithm that uses knowledge of the "local" data structure for prediction. A drawback is that the model training has to be done for each prediction step and the computation time of the training increases with $x \times p$. KNNs are used, for example, to impute values (fill missing values) and have the advantage that predicted values are always within the range of observed values of the target variable.

To use the KNN algorithm for prediction, we have to apply two essential methods in machine learning applications: *pre-processing* and *hyperparameter tuning*. These are described in respective sections below.

Checkpoint Compute the Euclidean metric and Manhattan distances between two points $x_a = (1, 1)$ and $x_b = (4, 5)$.

Solution

$$d_{\text{Euclidean}}(x_a, x_b) = \sqrt{\sum_{j=1}^{2}(x_{a,j} - x_{b,j})^2} = \sqrt{(4-1)^2 + (5-1)^2} = \sqrt{25} = 5 d_{\text{Manhattan:}}(x_a, x_b) = \sum_{j=1}^{2}|x_{a,j}$$

7.2.3 Essential methods for the modelling process

Building a ML model is an iterative process and it is typically not possible to know which ML algorithm will perform best when you start with the process. It is also essential that you understand your data in order to apply appropriate pre-processing steps, splitting your data wisely into training and testing sets, building a robust model, and achieving an informative model evaluation. This section introduces essential methods of the modelling process that allow you to walk a safe (and hopefully still spectacular) path. This section is inspired by the fantastic tutorial *Hands On Machine Learning in R* by Boehmke & Gladwell.

7.2.3.1 Model formulation

Remember that the aim of supervised ML is to find a model $\hat{Y} = f(X)$ so that $\hat{Y}$ agrees well with observations Y. We typically start with a research question where Y is given - naturally - by the problem we are addressing and we have a data set at hand where one or multiple predictors (or features) X are recorded along with Y. With the data set that you have become familiar with in preceeding chapters, we have information about how GPP (ecosystem-level photosynthesis) depends on set of abiotic factors, mostly meteorological measurements. In R, it is common to use the *formula* notation to specify the target and predictor variables. You have encountered formulas before, e.g., for a linear regression using the `lm()` function. To specify a linear regression model for `GPP_NT_VUT_REF` with three predictors `PPFD_IN`, `VPD_F`, and `TA_F`, we write:

```
lm(GPP_NT_VUT_REF ~ PPFD_IN + VPD_F + TA_F, data = df)
```

Actually, the way we formulate a model is independent of the algorithm, or *engine* that takes care of fitting $f(X)$. The R package **caret** provides a unified interface for using different ML algorithms implemented in separate packages. In other words, it acts as a *wrapper* for multiple different model fitting, or ML algorithms. This has the advantage that it unifies the interface (the way arguments are provided). caret also provides implementations for a set of commonly used tools for data processing (which we won't use here), model training, and evaluation. We'll use caret for model training with the function `train()` (more on model training in Chapter 7). Note however, that using a specific algorithm, which is implemented in a specific package outside caret, also requires that the respective package be installed and loaded. Using caret for specifying the same linear regression model as above, using the base-R `lm()` function, can be done with caret in a generalized form as:

```
library(caret)
```

```
train(
  form = GPP_NT_VUT_REF ~ .,
  data = df,
```

```
  method = "lm"
)
```

Of course, this is an overkill compared to just writing `lm(...)`. But the advantage of the unified interface is that we can simply replace the `method` argument to use a different ML algorithm. For example, to use KNN, we can write:

```
train(
  form = GPP_NT_VUT_REF ~ .,
  data = df,
  method = "knn"
)
```

Another common way of model formulation is not to use the formula notation, but to provide an argument `x` containing columns for all the predictors, and an argument `y` containing a vector of the target variable (with `length(y) = nrow(x)`). E.g., the caret function `train()` also takes such specifications:

```
train(
  x = dplyr::select(df, PPFD_IN, VPD_F, TA_F),
  y = df$GPP_NT_VUT_REF,
  method = "knn"
  )
```

7.2.3.2 Data splitting

The introductory example impressively demonstrated the importance of validating the fitted model with data that was not used for training. Thus, we can test the model's generalisability. The essential step that enables us to assess the model's *generalization error* is to hold out part of the data from training, and set it aside (leaving it absolutely untouched) for *testing* (*validation*).

There is no fixed rule for how much data are to be used for training and testing, respectively. We have to balance the trade-off between:

- Spending too much data for training will leave us with too little data for testing and the test results may not be robust. In this case, the sample size for getting robust validation statistics is not sufficiently large and we don't know for sure whether we are safe from an overfit model.
- Spending too much data for validation will leave us with too little data for training. In this case, the ML algorithm may not be successful at finding real relationships due to insufficient amounts of training data.

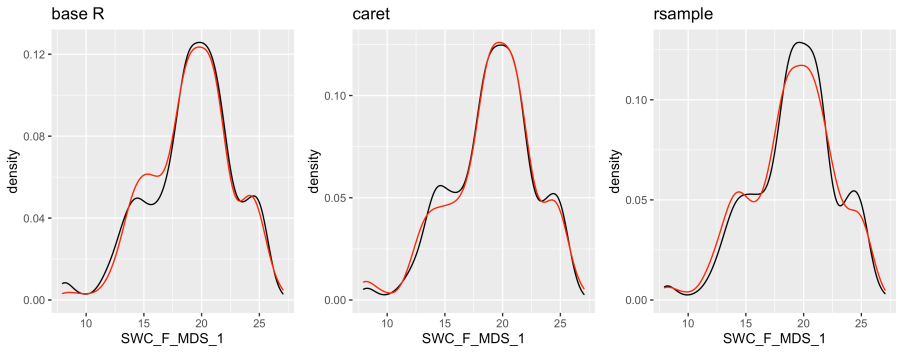
Typical splits are between 60-80% for training. However, in cases where the number of data points is very large, the gains from having more training data are marginal, but come at the cost of adding to the already high computational burden of model training.

In environmental sciences, the number of predictors is often smaller than the sample size ($p < n$), because its typically easier to collect repeated observations of a particular variable than to expand the set of variables being observed. Nevertheless, in cases where the number p gets large, it is important, and for some algorithms mandatory, to maintain $p < n$ for model training.

An important aspect to consider when splitting our data is to make sure that all "states" of the system for which we have data are approximately equally represented in training and testing sets. This is to make sure that the algorithm learns relationships $f(X)$ also under rare conditions X, for example meteorological extreme events.

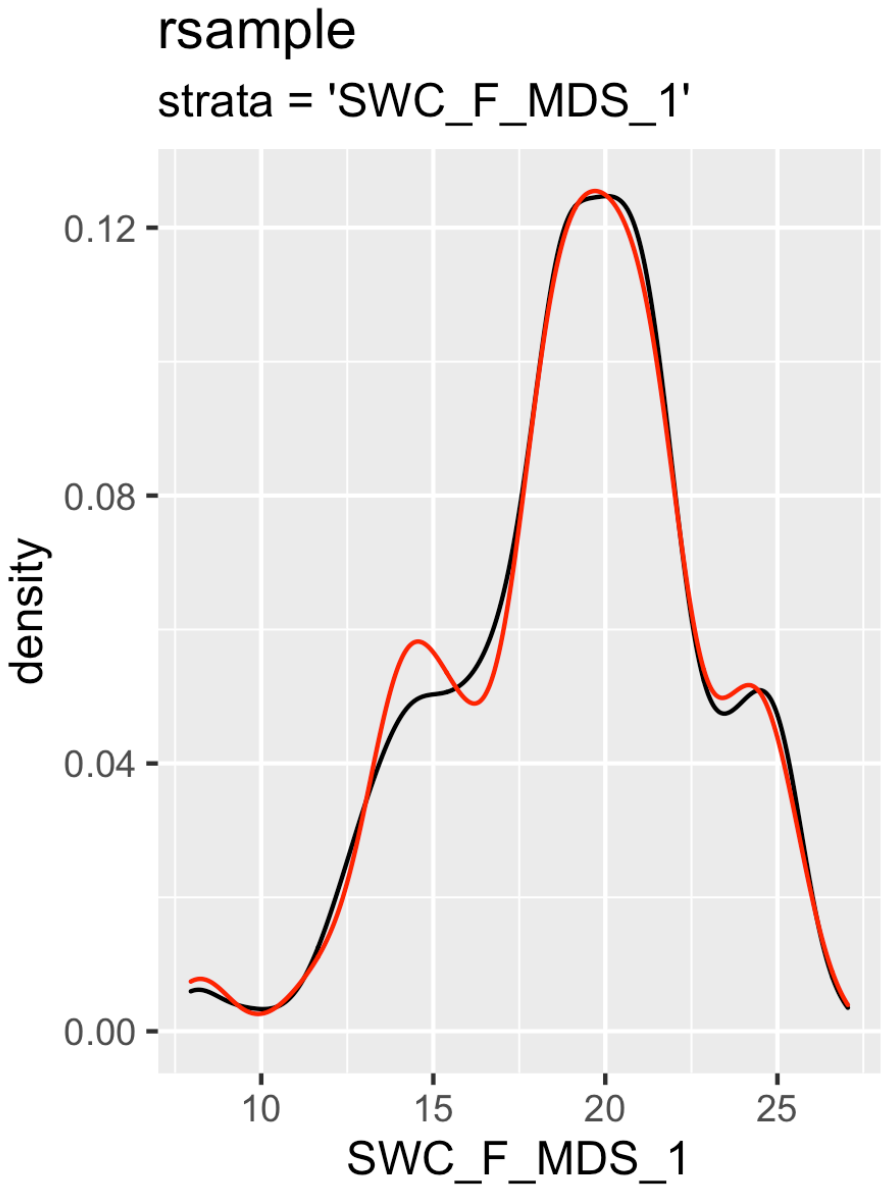
Several alternative functions for the data splitting step are available from different packages in R. Below is an example for their implementation using base R (`sample`), caret (`createDataPartition`), and rsample (`initial_split`, combined with `training` and `testing`) and their resulting distribution of soil water content values (`SWC_F_MDS_1`) in training and testing sets.

```
## using base-R
set.seed(123) # for reproducibility
index_base <- sample(1:nrow(df), round(nrow(df) * 0.7))
df_train_base <- df %>%
  slice(index_base)
df_test_base <- df %>%
  slice(-index_base)

## using caret
library(caret)
set.seed(123)  # for reproducibility
index_caret <- createDataPartition(df$SWC_F_MDS_1, p = 0.7, list = FALSE)
df_train_caret <- df %>%
  slice(index_caret)
df_test_caret <- df %>%
  slice(-index_caret)

## using rsample
library(rsample)
set.seed(123)  # for reproducibility
split_rsample <- initial_split(df, prop = 0.7)
df_train_rsample <- training(split_rsample)
df_test_rsample <- testing(split_rsample)
```



In this example, you can see that not in all cases, data with very low soil water content values are sufficiently represented in the testing set (see red curve for 'base R'). It may be critical to have extreme values both in the testing and training sets, e.g., if we are interested to capture responses of our target variable to very dry soil conditions. A way to control for this aspect and achieve a more similar representation of data in training and testing sets across all values, we can apply a *stratified sampling*. Here, we use the `initial_split` function from the rsample package again.

```
split_rsample_strat <- initial_split(df, prop = 0.7, strata = "SWC_F_MDS_1")
df_train_rsample_strat <- training(split_rsample_strat)
df_test_rsample_strat <- testing(split_rsample_strat)
```



7.2.4 Bonus

To generalize this and achieve a sort of "stratification" with respect to all predictors, we can use the `maxDissim` function from the **caret** package (uses the **proxy** package). It samples the most dissimilar data points after an initial

pick and thus achieves a good representation of the entire feature space in our training set.

```
library(proxy)
set.seed(123)
startSet <- sample(1:nrow(df), 5)   # initial pick of five
samplePool <- df[-startSet,]
start <- df[startSet,]
index_caret <- maxDissim(start, samplePool, n = floor(0.7 * nrow(df))-5 )
```

7.2.4.1 Pre-processing

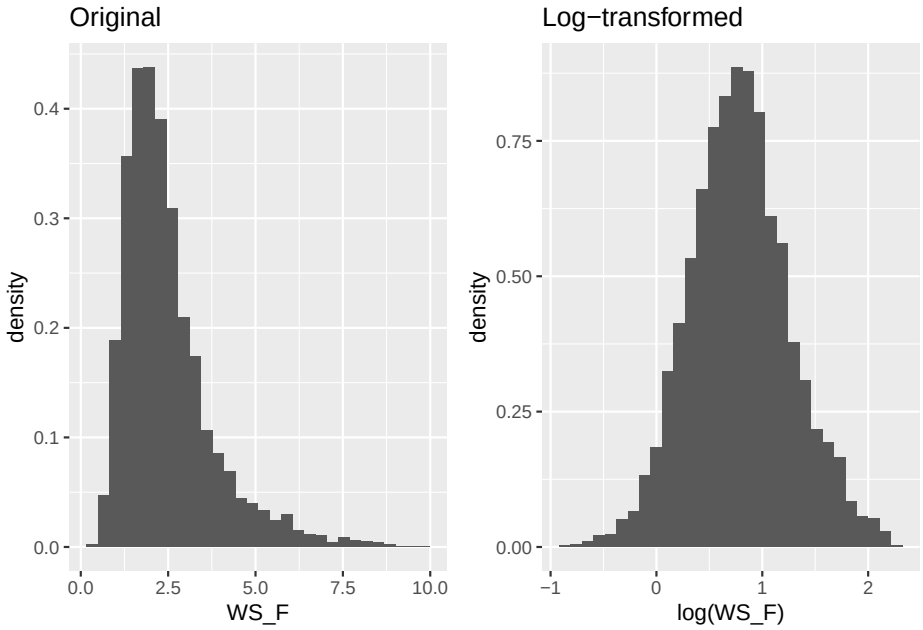
Skewed data, outliers, and values covering multiple orders of magnitude can create difficulties for certain ML algorithms, e.g., or KNN or neural networks. Other algorithms, like tree-based methods (e.g., Random Forest) are more robust against such issues. This section introduces pre-processing methods that enable improved ML models and prepare the data for specific needs by the ML algorithm. There is a difference between data wrangling which we learned about in Chapter 2 and pre-processing as part of the modelling workflow. Data wrangling can be considered to encompass the steps to get raw data into a format that is usable for modelling with any sort of ML algorithm. Data wrangling are the steps to prepare the data set with variable selection, removal of bad or missing data, complementing variables from different sources, and aggregating to the desired resolution or granularity (e.g., averaging over all time steps in a day, or over all replicates in a sample). In contrast, *pre-processing* refers to the additional steps that are either required by the ML algorithm (examples will be given below) or the transformation of variables guided by the resulting improvement of the predictive power of the ML model.

Target engineering

Target engineering refers to pre-processing of the target variable. Its application can enable improved predictions, particularly for models that make assumptions about errors (e.g., normally distributed errors in linear regression) and when the target variable follows a "special" distribution (e.g., heavily skewed distribution, or where the target variable is a fraction that is naturally bounded by 0 and 1). A simple log-transformation of the target variable can often resolve issues with skewed distributions. An implication of a log-transformation is that errors in predicting values in the upper end of the observed range do not affect the model disproportionally compared to errors in the lower range.

In our data set of half-hourly ecosystem flux and meteorological measurements, the variable `WS_F` (wind speed) is skewed. The target variable that we have considered so far (`GPP_NT_VUT_REF`) is not skewed. In a case where we would consider `WS_F` to be our target variable, we would thus consider applying a log-transformation.

```
library(patchwork)  # to combine two plots into separate panels of a single plot

gg1 <- df %>%
  ggplot(aes(x = WS_F, y = ..density..)) +
  geom_histogram() +
  labs(title = "Original")

gg2 <- df %>%
  ggplot(aes(x = log(WS_F), y = ..density..)) +
  geom_histogram() +
  labs(title = "Log-transformed")

gg1 + gg2   # the + is from the patchwork library
```



When defining any pre-processing step, it should be specified as a "recipe" or "blueprint", and not actually executed on the data itself before we start with the model training. Such a "recipe" can then be applied to any new data, while the parameters of the data pre-processing transformations are different each time. The reason for not actually executing the data transformation is *data leakage*. It happens when information from the validation data somehow finds its way into the training step. Don't worry if this sounds incomprehensible. We'll learn more about it below. For now, you can focus on the types of pre-processing steps, what they do, and how they are implemented as part of a pre-processing "recipe" in R.

The **recipes** package offers a powerful way to specify pre-processing steps in

R and is gaining traction as part of the tidymodels ecosystem. It allows us to sequentially build a pre-processing pipeline using the pipe (`%>%`) operator.

```
library(recipes)
recipe(WS_F ~ ., data = df) %>%   # it's of course non-sense to model wind speed like
  step_log(all_outcomes())
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor         10
##
## Operations:
##
## Log transformation on all_outcomes()
```

The formula, specified as an argument in the `recipe()` function, defines variable "roles" (in recipes-speak). What's left of the `~` is interpreted as an "outcome" (the target variable). By writing `~ .`, we specify all remaining variables in the data frame as "predictors". As an argument in the `step_log()` function, we write `all_outcomes()` to declare that the log transformation is applied only to the "outcome" variables we specified with the formula before.

A log-transformation doesn't necessarily result in a perfect normal distribution of transformed values. The *Box-Cox* can get us closer. It can be considered a generalization of the log-transformation. Values are transformed according to the following function:

$$y(\lambda) = \begin{cases} \frac{Y^{\lambda}-1}{\lambda}, & y \neq 0 \\ \log(Y), & y = 0 \end{cases}$$

λ is treated as a parameter that is fitted such that the resulting distribution of values y approaches the normal distribution. To specify a Box-Cox-transformation as part of the pre-processing, we can use `step_BoxCox()` from the recipes package. How do transformed values look like? The strict separation of the pre-processing recipe from its execution on the data also forces us to be explict about what part of the data (training vs. validation/testing subsets) are used for determining pre-processing parameters. Let's be clear about this in the next example and apply the initial data split first here and use the training subset for the recipe.

```
library(rsample)
split <- initial_split(df, prop = 0.5)
df_train <- training(split)
df_test <- testing(split)
```

```
recipe_example <- recipe(WS_F ~ ., data = df_train) %>%
   step_BoxCox(all_outcomes())
```

Now, you may be frustrated by the fact that this doesn't actually transform any data, but you want to look at how values would change. There are two more steps involved to get there. This might seem a nuisance at first but their separation is actually quite beautiful and translates the conception of the pre-processing as a "blueprint" into the way we write the code. You'll understand why this is so useful by the end of Chapter 7.

To get the parameter of the Box-Cox-transformation (λ) based on the data set `df_train`, we have to "prepare" the recipe.

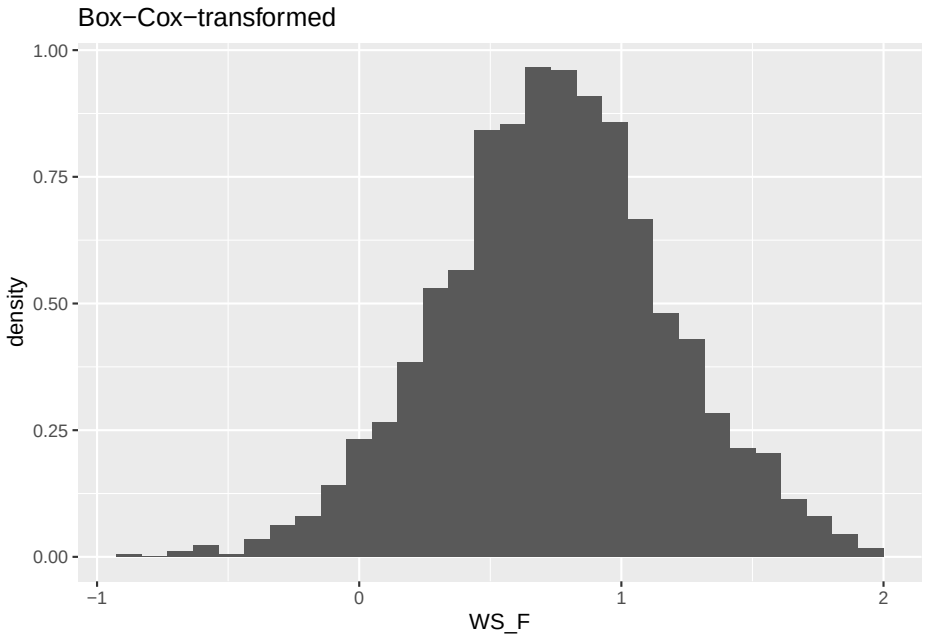
```
prep_example <- prep(recipe_example, training = df_train)
```

Finally we can actually transform the data. That is, "bake" the prepared recipe.

```
df_baked <- bake(prep_example, new_data = df_test)
```

The Box-Cox-transformed data now looks like this:

```
df_baked %>%
  ggplot(aes(x = WS_F, y = ..density..)) +
  geom_histogram() +
  labs(title = "Box-Cox-transformed")
```



Note that the Box-Cox-transformation can only be applied to values that are

strictly positive. In our example, wind speed (`WS_F`) is. If this is not satisfied, a Yeo-Johnson transformation can be applied.

```
recipe(WS_F ~ ., data = df) %>%
  step_YeoJohnson(all_outcomes())
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor         10
##
## Operations:
##
## Yeo-Johnson transformation on all_outcomes()
```

7.2.4.2 Standardization

Several algorithms explicitly require data to be standardized. That is, values of all predictors vary within a comparable range. The necessity of this step becomes obvious when considering KNN, where the magnitude of the distance is strongly influenced by the order of magnitude of the predictor values. To get a quick overview of the distribution of all variables (columns) in our data frame, we can use the skimr package.

```
library(skimr)
skim(df)
```

We see for example, that typical values of `PPFD_IN` are by a factor 100 larger than values of `VPD_F`. A distance calculated based on these raw values would therefore be strongly dominated by the difference in `PPFD_IN` values, and differences in `VPD_F` would hardly affect the distance. Therefore, the data must be standardized before using it with the KNN algorithm. Standardization is done, for example, by dividing each variable, that is all values in one column, by the standard deviation of that variable, and then subtracting its mean. This way, the resulting standardized values are centered around 0, and scaled such that a value of 1 means that the data point is one standard deviation above the mean of the respective variable (column). When applied to all predictors individually, the absolute values of their variations can be directly compared and only then it can be meaningfully used for determining the distance.

Also other algorithms require data to be normalised before training. A prominent example are Artificial Neural Networks.

Standardization can be done not only by centering and scaling (as described

above), but also by *scaling to within range*, where values are scaled such that the minimum value within each variable (column) is 0 and the maximum is 1.

Using the recipes package again, we can specify centering and scaling by two separate steps as:

```
recipe_example <- recipe(GPP_NT_VUT_REF ~ ., data = df_train) %>%
  step_center(all_numeric(), -all_outcomes()) %>%
  step_scale(all_numeric(), -all_outcomes())
```

Here, we used selectors to apply the recipe step to several variables at once. The first selector, `all_numeric()`, selects all variables that are either integers or real values. The second selector, `-all_outcomes()` removes any outcome (target) variables from this recipe step.

As seen above for the feature engineering example, this does not return a standardized version of the data frame `df_train`, but rather the information that allows us to apply the same standardization also to other data sets. In other words, we use the distribution of values in the data set to which we applied the function (here `df_train`) to determine the standardization (here: mean and standard deviation). As you will learn below, this is essential and critical to avoid *data leakage* (where information from the testing data set leaks into the training data set).

7.2.4.3 Zero-variance predictors

Sometimes, the data generation process yields variables that have the same value in each observation. And sometimes this is due to failure of the measurement device or some other bug in the data collection pipeline. Either way, this may cause some algorithms to crash or become unstable. Such "zero-variance" predictors are usually removed altogether. The same applies also to variables with "near-zero variance". That is, variables where only a few unique values occur in the entire data set with a high frequency. The danger is that, when data is split into training and testing sets, the variable may effectively become a "zero-variance" variable within the training subset.

We can test for zero-variance or near-zero variance predictors by quantifying the following metrics:

- Frequency ratio: Ratio of the frequency of the most common predictor over the second most common predictor. This should be near 1 for well-behaved predictors and get very large for problematic ones.
- Percent unique values: The number of unique values divided by the total number of rows in the data set (times 100). For problematic variables, this ratio gets small (approaches 1/100).

The function `nearZeroVar` of the caret package flags suspicious variables (`zeroVar = TRUE` or `nzv = TRUE`). In our data set, we don't find any:

```
nearZeroVar(df, saveMetrics= TRUE)
```

```
##                freqRatio percentUnique zeroVar   nzv
## TA_F            1.000000      94.02697   FALSE FALSE
## SW_IN_F         1.000000      99.22929   FALSE FALSE
## LW_IN_F         1.500000      98.54115   FALSE FALSE
## VPD_F           1.125000      75.63997   FALSE FALSE
## PA_F            1.142857      53.94990   FALSE FALSE
## P_F            69.600000      59.34489   FALSE FALSE
## WS_F            1.142857      62.95073   FALSE FALSE
## CO2_F_MDS       1.000000      98.07322   FALSE FALSE
## PPFD_IN         1.000000      99.58712   FALSE FALSE
## GPP_NT_VUT_REF  1.000000      99.77980   FALSE FALSE
## USTAR           1.000000     100.00000   FALSE FALSE
```

Using the recipes package, we can add a step that removes zero-variance predictors by:

```
recipe_example <- recipe_example %>%
  step_zv(all_predictors())
```

7.2.4.4 One-hot encoding

For ML algorithms that require that all predictors be numerical (e.g., neural networks, or KNN), categorical predictors have to be pre-processed and converted into new numerical predictors. The most common such transformation is *one-hot encoding*, where a categorical feature (predictor) that has N levels is replaced by N new features that contain either zeros or ones depending whether the value of the categorical feature corresponds to the respective column. This creates perfect collinearity between these new column, we can also drop one of them. This is referred to as *dummy encoding*. Check out the Figure from www.kaggle.com for a one-hot encoding visualization. Using the recipes package, one-hot encoding is implemented by:

```
recipe(GPP_NT_VUT_REF ~ ., data = df) %>%
  step_dummy(all_nominal(), one_hot = TRUE)
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor         10
##
## Operations:
```

```
##
## Dummy variables from all_nominal()
```

Checkpoint

Apply a one-hot encoding to the to the variable Species from the data set iris (from the package data sets). Load the data by `library(datasets)`. The dataset's name is `iris`. Visualize the result.

Solution

```
library(datasets)
rec <- recipe( ~ ., data = iris) %>%
  step_dummy(Species, one_hot = T) %>%
  prep(training = iris)

summary(rec)
```

```
## # A tibble: 7 x 4
##   variable           type    role      source
##   <chr>              <chr>   <chr>     <chr>
## 1 Sepal.Length       numeric predictor original
## 2 Sepal.Width        numeric predictor original
## 3 Petal.Length       numeric predictor original
## 4 Petal.Width        numeric predictor original
## 5 Species_setosa     numeric predictor derived
## 6 Species_versicolor numeric predictor derived
## 7 Species_virginica  numeric predictor derived
```

```
rec$template$Species_setosa
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$Species
## [1] "contr.treatment"
```

```
rec$template$Species_versicolor
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0
```

```
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$Species
## [1] "contr.treatment"
```

```
rec$template$Species_virginica
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$Species
## [1] "contr.treatment"
```

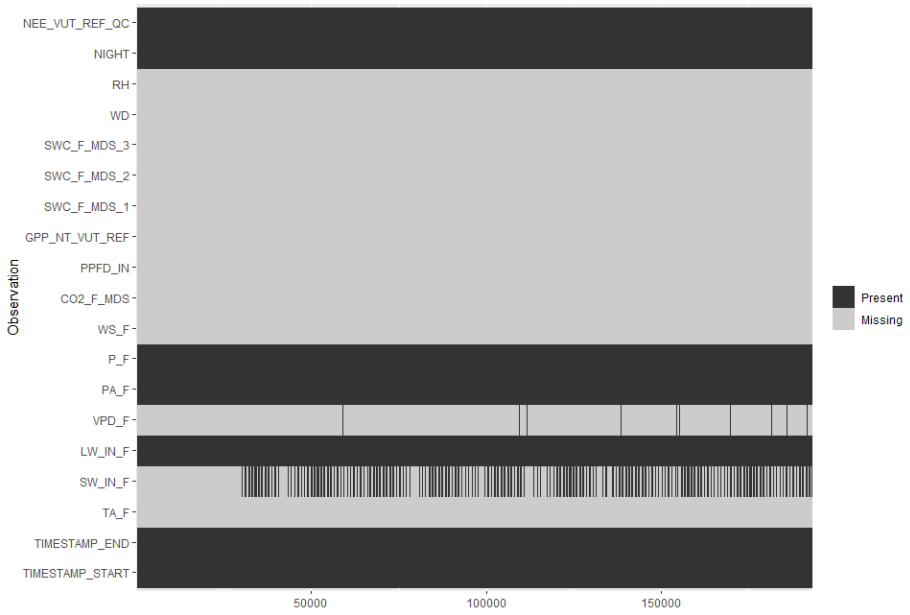
7.2.4.5 Dealing with missingness and bad data

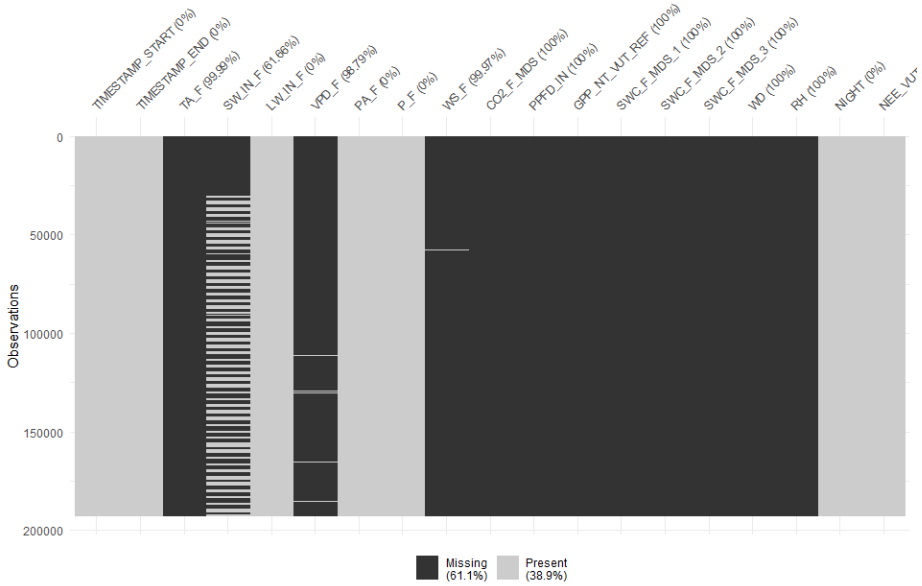
Several ML algorithms require missing values to be removed. That is, if any of the cells in one row has a missing value, the entire cell gets removed. We've seen how this is implemented in Chapter 2 for example with `na.omit()` or `tidyr::drop_na()`. Data may be missing for several reasons. Some yield random patterns of missing data, others not. In the latter case, we can speak of *informative missingness* (Kuhn & Johnson, 2003) and its information can be used for predictions. For categorical data, we may replace such data with `"none"` (instead of `NA`), while randomly missing data may be dropped altogether. Some ML algorithms (mainly tree-based methods, e.g., random forest) can handle missing values. However, when comparing the performance of alternative ML algorithms, they should be tested with the same data and removing missing data should be done beforehand.

Visualising missing data is essential for making decisions about dropping rows with missing data versus removing predictors from the model (which would imply too much data removal). The cells with missing data in a data frame can be visualised either using ggplot and `geom_raster()` or similarly (with additional information about the percentage of missing data for each variable) with `vis_miss()` from the `visdat` package. Note: the code below is set to `raw` since the notebook sturggles to handle the amount of values in *hhdf*. We have provided the results as images instead.

```
load("./data/FLX_CH-Lae_FLUXNET2015_FULLSET_HH_2004-2014_1-3_CLEAN.csv")

# DO NOT RUN THIS CODE IS NOT WORKING IN THE NOTEBOOK (or only for a max of 30000 valu
```

```
## Code from https://bradleyboehmke.github.io/HOML/engineering.html#dealing-with-missingness
hhdf %>%
  is.na() %>%
  reshape2::melt() %>%
  ggplot(aes(Var2, Var1, fill=value)) +
  geom_raster() +
  coord_flip() +
  scale_y_continuous(NULL, expand = c(0, 0)) +
  scale_fill_grey(name = "",
                  labels = c("Present",
                             "Missing")) +
  xlab("Observation") +
  theme(axis.text.y  = element_text(size = 100/ncol(df)))

library(visdat)
vis_miss(
  hhdf,
  cluster = FALSE,
  warn_large_data = FALSE
  )
```





The question about what is "bad data" and whether or when it should be removed is often critical. In Chapter 3, we've encountered quality control information in our data set (e.g., `NEE_VUT_REF_QC`), and removed data that didn't satisfy our own definition of the quality criterion. Such decisions are important to keep track of and should be reported as transparently as possible in publications. In reality, where the data generation process may start in the field with actual human beings writing notes in a lab book, and where the human collecting the data is often not the same as the human writing the paper that contains that data, it's often more difficult to keep track of such decisions. As a general principle, it is advisable to design data records such that decisions made during its process remain transparent throughout all stages of the workflow and that sufficient information be collected to enable later revisions of particularly critical decisions.

Often, and particularly when humans (and more so in the case of multiple humans) collected and recorded the data, steps are necessary to make it machine-readable. For example, dates need to be formatted consistently, units need to be specified consistently, etc. An overview of common problems with bad data and their solutions is given by the Quartz Guide to Bad Data.

Chapter 8

Supervised machine learning methods II

Chapter 9

Application 1: Variable selection

9.1 Introduction

In Chapter 7, we noted that the coefficient of determination R^2 may increase even when uninformative predictors are added to a model. This will ascribe some predictive power to an uninformative predictor that is in fact misguided by its (random) correlation with the target variable. Often, we start out formulating models, not knowing beforehand what predictors should be considered and we are tempted to use them all because the full model will always yield the best R^2. In such cases, we're prone to building overconfident models that perform well on the training set, but will not perform well when predicting to new data.

In this application session, we'll implement an algorithm that sequentially searches the best additional predictor to be included in our model, starting from a single one. This is called *stepwise-forward* regression (see definition below). There is also *stepwise-backward* regression where predictors are sequentially removed from a model that includes them all. The challenge is that we often lack the possibility to confidently assess generalisability. The effect of spuriously increasing R^2 by adding uninformative predictors can be mitigated, as we noted in Chapter 7, by considering alternative metrics that penalize the number of predictors in a model. They balance the tradeoff between model complexity (number of variables in the linear regression case) and goodness of fit. Such metrics include the *adjusted-*R^2,the Akaike Information Criterion (AIC), or the Bayesian Information Criterion (BIC). In cases, where sufficient data is available, also cross-validation can be used for assessing the generalisability of alternative models. Here, we'll assess how these different metrics behave for a sequence of linear regression models with an increasing

number of predictors. You'll learn to write code that implements an algorithm determining the order in which variables enter the model, starting from one and going up to fourteen predictors. You'll write your own stepwise-forward regression code.

Let's get started...

Forward stepwise regression

1. Let $\mathcal{M}_0$ denote the null model, which contains no predictors.
2. For $k = 0, .., p-1$:
   (a) Consider all $p-k$ models that augment $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the best model among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having the highest R^2 .
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, AIC, BIC, or adjusted R^2.

9.2 Warm-up 1: Nested for-loop

Given a matrix A and a vector B (see below), do the following tasks:

- Replace the missing values (`NA`) in the first row of A by the largest value of B. After using that element of B for imputing A, drop that element from the vector B and proceed with imputing the second row of A, using the (now) largest value of the updated vector B, and drop that element from B after using it for imputing A. Repeat the same procedure for all four rows in A.
- After imputing (replacing) in each step, calculate the mean of the remaining values in B and record it as a single-row data frame with two columns `row_number` and `avg`, where `row_number` is the row number of A where the value was imputed, and `avg` is the mean of remaining values in B. As the algorithm proceeds through rows in A, sequentially bind the single-row data frame together so that after completion of the algorithm, the data frame contains four rows (corresponding to the number of rows in A).

```
A <- matrix(c(6, 7, 3, NA, 15, 6, 7,
              8, 9, 12, 6, 11, NA, 3,
              9, 4, 7, 3, 21, NA, 6,
              7, 19, 6, NA, 15, 8, 10),
```

```
            nrow = 4, byrow = TRUE)
B <- c(8, 4, 12, 9, 15, 6)
```

Before implementing these tasks, try to write down a pseudo code. This is code-like text that may not be executable, but describes the structure of real code and details where and how major steps are implemented. Next, you'll need to write actual R code. For this, you will need to find answers to the following questions:

- How to go through each of the element in matrix?
- How to detect NA value?
- How to drop an element of a given value from a vector?
- How to add a row to an existing data frame?

Solution:

```
library(tidyverse)

summ <- data.frame()

for (i in 1:nrow(A)){

  for (j in 1:ncol(A)){

    if (is.na(A[i,j])){
      A[i,j] <- max(B)
    }
  }


  B <- B[-which(B == max(B))] # update the B vector removing the biggest values
}

  summ <- bind_rows(summ, data.frame(row_number = i, avg = mean(B)))

summ
```

```
##   row_number avg
## 1          4   5
```

9.3 Warm-up 2: Find the best single predictor

The first step of a stepwise forward regression is to find the single most powerful predictor in a univariate linear regression model for the target variable `GPP_NT_VUT_REF` among all fourteen available predictors in our data set (all except those of type `date` or `character`). Implement this first part of the search,

using the definition of the stepwise-forward algorithm above. Remove all rows with at least one missing value before starting the predictor search.

- Which predictor achieves the highest R^2?
- What value is the R^2?
- Visualise R^2 for all univariate models, ordered by their respective R^2 values.
- Do you note a particular pattern? Which variables yield similar R^2? How do you expect them to be included in multivariate models of the subsequent steps in the stepwise forward regression?

Hints:

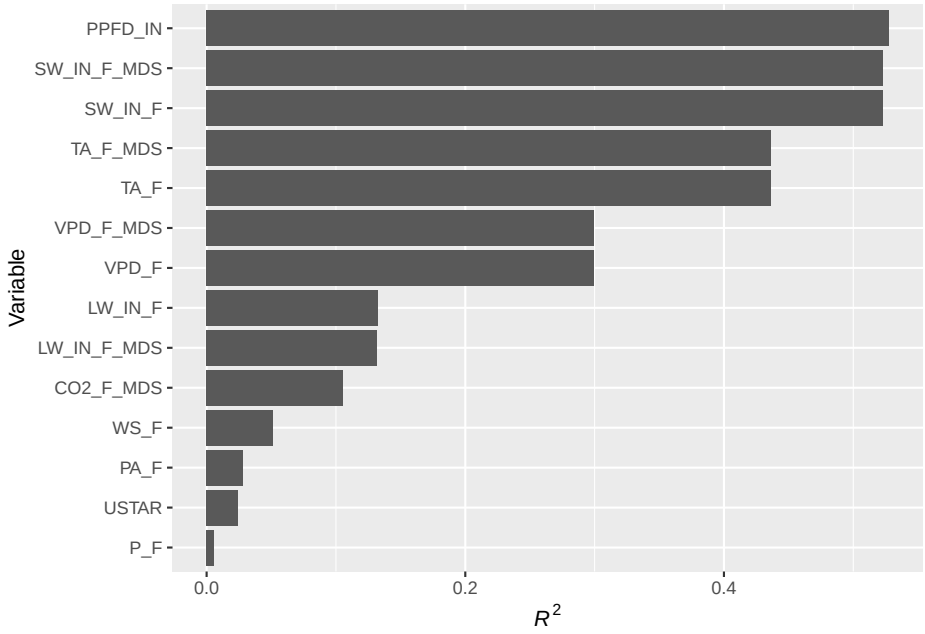
- Model structure:
  - The "counter" variables in the for loop can be provided as a vector, and the counter will sequentially take on the value of each element in that vector. For example: `for (var in all_predictors){ ... }`.
- Algorithm:
  - To record R^2 values for the different models, you may start by creating an empty vector (`vec <- c()`) before the loop and then sequentially add elements to that vector inside the loop (`vec <- c(vec, new_element)`). Alternatively, you can do something similar, but with a data frame (initialising with `df_rsq <- data.frame()` before the loop, and adding rows by `df_rsq <- bind_rows(df_rsq, data.frame(pred = predictor_name, rsq = rsq_result))` inside the loop).
  - A clever way how to construct formulas dynamically is described, for example in this stackoverflow post.
- Value retrieving:
  - Extract the R^2 from the linear model object: `summary(fit_lin)[["r.squared"]]`
- Visualising:
  - Search for solutions for how to change the order of levels to be plotted yourself.

```
library(tidyverse)
## read CSV and drop missing values in one step
df <- read_csv("./data/ddf_for_08_application.csv") %>%
  drop_na()

## specify target variable
target <- 'GPP_NT_VUT_REF'
```

```
## determine predictors as all except site ID, timestamp, and the target (should be 14)
preds <- df %>%
  dplyr::select(-siteid, -TIMESTAMP, -GPP_NT_VUT_REF) %>%
  names()

## initialise an empty data frame (necessary, because otherwise we cannot use bind_rows() below)
df_rsq <- data.frame()
# rsq_list <- c()  # alternative for vector

for (var in preds){

  ## create formula dynamically
  forml <- as.formula(paste(target, "~", var))

  ## fit linear model
  fit_lin <- lm(forml, data = df)

  ## extract R2 from linear model
  rsq <- summary(fit_lin)[["r.squared"]]

  ## add a row to the data frame that holds the results
  df_rsq <- bind_rows(df_rsq, data.frame(pred = var, rsq = rsq))

  # rsq_list <- c(rsq_list,rsq)  # alternative with vector
}

## print best single predictor and its R2
df_rsq %>%
  arrange(-rsq) %>%  # arrange by R2 in descending order (highest R2 on top)
  slice(1)           # print only the first row (with highest R2)
```

```
##      pred       rsq
## 1 PPFD_IN 0.5276123
```

```
library(ggplot2)

## alternative: determine the first variable to enter into our model
# preds[which.max(rsq_list)]

## use the data frame that holds the results for plotting
df_rsq %>%
  ggplot(aes(x = reorder(pred, rsq), y = rsq)) +
  geom_bar(stat = "identity") +
  labs(y = expression(italic(R)^2), x = "Variable") +
  coord_flip()
```



Solution:

`PPFD_IN` achieves the highest R^2 value, and the corresponding R^2 values is 0.5276123.

We could see from the above plot that `LW_IN_F_MDS` and `LW_IN_F`, `VPD_F` and `VPD_F_MDS`, `SW_IN_F` and `SW_IN_F_MDS` share basically the same R^2 values. Those variables are highly correlated and are explaining the same amount of variance in the target variable. In the subsequent steps of the stepwise forward regression, since each of the two variables are highly correlated with each other, it's likely that only one of them will be selected into our final model.

9.4 Full stepwise regression

Now, we take it to the next level and implement a full stepwise forward regression as described above. For each step (number of predictors k), record the following metrics: R^2, *adjusted-*R^2, the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and the 5-fold cross-validation R^2 and RMSE.

- Write pseudo-code for how you plan to implement the algorithm first.
- Implement the algorithm in R, run it and display the order in which predictors enter the model.
- Display a table with the metrics of all k steps, and the single variable, added at each step.

Hints:

- Model structure:
  - Recall what you learned in the breakout session, you may use the same idea on this task. Try to think of the blueprint (*pseudo-code*) first: How to go through different models in each forward step? How to store predictors added to the model and how to update candidate predictors?
- Algorithm:
  - A complication is that the set of predictors is sequentially complemented at each step of the search through k. You may again use `vec <- list()` to create an empty vector, and then add elements to that vector by `vec <- c(vec, new_element)`.
  - It may be helpful to explicitly define a set of "candidate predictors" that may potentially be added to the model as a vector (e.g., `preds_candidate`), and define predictors retained in the model from the previous step in a separate vector (e.g., `preds_retained`). In each step, search through `preds_candidate`, select the best predictor, add it to `preds_retained` and remove it from `preds_candidate`.
  - At each step, record the metrics and store them in a data frame for later plots. As in the first "warm-up" exercise, you may record metrics at each step as a single-row data frame and sequentially stack (bind) them together.
  - (As above) A clever way how to construct formulas dynamically is described, for example in this stackoverflow post.
  - The metrics for the k models are assessed *after* the order of added variables is determined. To be able to determine the metrics, the k models can be saved by constructing a list of models and sequentially add elements to that list (`mylist[[ name_new_element ]] <- new_element`). You can also fit the model again after determining which predictor worked best.
  - Your code will most certainly have bugs at first. To debug efficiently, write code first in a simple R script and use the debugging options in RStudio (see here).
- Value retriving
  - To get AIC and BIC values for a given model, use the base-R functions `AIC()` and `BIC()`.
  - To get the cross-validated R^2 and RMSE, use the caret function `train()` with RMSE as the loss function, and `method = "lm"` (to fit a linear regression model). Then extract the values by `trained_model$results$Rsquared` and `trained_model$results$RMSE`.

- Displaying:
  - To display a table nicely as part of the RMarkdown html output, use the function `knitr::kable()`
  - To avoid reordering of the list of variable names in plotting, change the type of variable names from "character" to "factor" by `pred <- factor(pred, levels = pred)`

```
library(caret)  # need train() for cross-validation

## specify target variable (as above)
target <- 'GPP_NT_VUT_REF'

## determine predictors as all except site ID, timestamp, and the target (should be 14
preds <- df %>%
  dplyr::select(-siteid, -TIMESTAMP, -GPP_NT_VUT_REF) %>%
  names()

# This is the vector of candidate predictors to be added in the model. To begin with,
preds_candidate <- preds

# predictors retained in the model from the previous step. To begin with, is empty.
preds_retained <- c()

## work with lists as much as possible (more flexible!)
df_metrics <- data.frame()

## outer loop for k predictors
for (k_index in 1:length(preds)){

  # rsq_candidates <- c()
  df_rsq_candidates <- data.frame()
  linmod_candidates <- list()

  ## inner loop for single additional predictor
  for (ipred in preds_candidate){

    # variable vector (new variable + retained variables) used in regression
    pred_add <- c(preds_retained, ipred)

    # define formulate with newly-added predictor
    forml  <- as.formula(paste( target, '~', paste(pred_add, collapse = '+')))

    # fit linear model
    fit_lin <- lm(forml, data = df)
```

```
    # add model object to list, and name the element according to the added variable
    linmod_candidates[[ ipred ]] <- fit_lin

    # record metrics for all candidates
    rsq <- summary(fit_lin)[["r.squared"]]
    df_rsq_candidates <- bind_rows(df_rsq_candidates, data.frame(pred = ipred, rsq = rsq))  # whe
    # rsq_candidates <- c(rsq_candidates,  rsq)  # when storing R2 as a vector

  }

  ## get name of candidate predictor that achieved the highest R2.
  pred_add <- df_rsq_candidates %>%  # when storing R2 in a data frame
    arrange(desc(rsq)) %>%
    slice(1) %>%
    pull(pred) %>%
    as.character()

  # pred_add <- preds_candidate[ which.max(rsq_candidates) ]   # when storing R2 as a vector

  ## add best predictors to retained predictors
  preds_retained <- c(preds_retained, pred_add)

  ## get the cross-validation R2
  forml  <- as.formula(paste( target, '~', paste(preds_retained, collapse = '+')))
  control <- trainControl(method = "cv", number = 5)
  set.seed(123)
  # cv_modl <- train(forml, data = df, method = "lm", , metric = "Rsquared", trControl = control)
  cv_modl <- train(forml, data = df, method = "lm", metric = "RMSE", trControl = control)

  # record AIC and BIC and adjusted-R2 of the respective model
  df_metrics <- df_metrics %>%
    bind_rows(
      data.frame( pred = pred_add,
                  rsq = summary(linmod_candidates[[ pred_add ]])[["r.squared"]],
                  adj_rsq = summary(linmod_candidates[[ pred_add ]])[["adj.r.squared"]],
                  cv_rsq = cv_modl$results$Rsquared,
                  cv_rmse = cv_modl$results$RMSE,
                  aic = AIC(linmod_candidates[[ pred_add ]]),
                  bic = BIC(linmod_candidates[[ pred_add ]])
      )
    )


  # remove the selected variable from the candidate variable list
  preds_candidate <- preds_candidate[-which(preds_candidate == pred_add)]
```

```
  # preds_candidate <- setdiff(preds_candidate,pred_add)  # alternative

}

data.frame(df_metrics$pred) # order in which variables enter the model
```

```
##    df_metrics.pred
## 1          PPFD_IN
## 2      LW_IN_F_MDS
## 3            VPD_F
## 4             WS_F
## 5      SW_IN_F_MDS
## 6        CO2_F_MDS
## 7             TA_F
## 8             PA_F
## 9         TA_F_MDS
## 10           USTAR
## 11             P_F
## 12       VPD_F_MDS
## 13         LW_IN_F
## 14         SW_IN_F
```

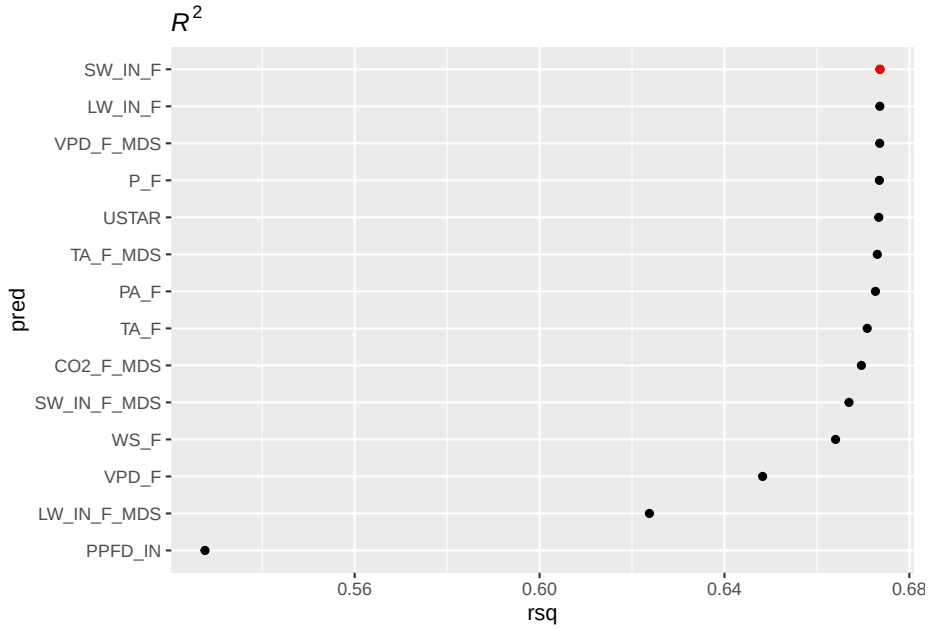
```
df_metrics %>% knitr::kable()
```

pred	rsq	adj_rsq	cv_rsq	cv_rmse	aic	bic
PPFD_IN	0.5276123	0.5274601	0.5286098	2.946732	15529.18	15547.30
LW_IN_F_MDS	0.6237535	0.6235110	0.6244311	2.631186	14824.62	14848.78
VPD_F	0.6482531	0.6479128	0.6488913	2.544541	14617.55	14647.76
WS_F	0.6640335	0.6636000	0.6644851	2.486970	14477.03	14513.28
SW_IN_F_MDS	0.6669387	0.6664013	0.6672468	2.476793	14452.07	14494.35
CO2_F_MDS	0.6696209	0.6689811	0.6695521	2.467823	14428.96	14477.28
TA_F	0.6708789	0.6701350	0.6703978	2.464609	14419.11	14473.48
PA_F	0.6726644	0.6718185	0.6718171	2.459555	14404.22	14464.63
TA_F_MDS	0.6730699	0.6721192	0.6642220	2.487336	14402.37	14468.82
USTAR	0.6733802	0.6723246	0.6639334	2.488412	14401.42	14473.91
P_F	0.6735098	0.6723487	0.6638548	2.488747	14402.19	14480.72
VPD_F_MDS	0.6735755	0.6723086	0.6624537	2.493879	14403.57	14488.14
LW_IN_F	0.6736125	0.6722398	0.6623996	2.494079	14405.22	14495.83
SW_IN_F	0.6736372	0.6721585	0.6620120	2.495602	14406.98	14503.63

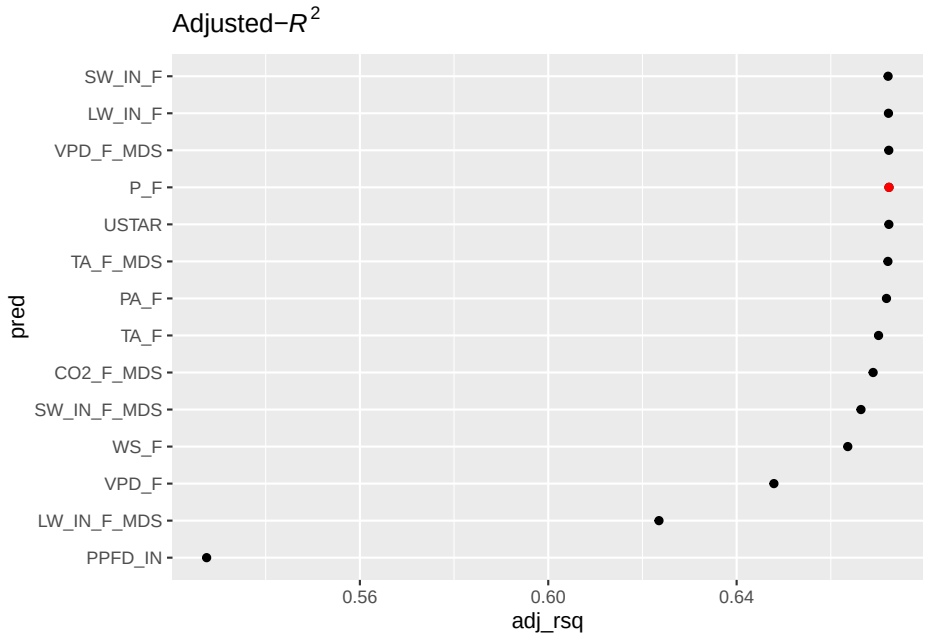
- Visualise all metrics as a function of the number of predictors (add labels for the variable names of the added predictor). Highlight the best-performing model based on the respective metric. How many predictors are in the best performing model, when assessed based on each metric?

```
library(ggplot2)
```

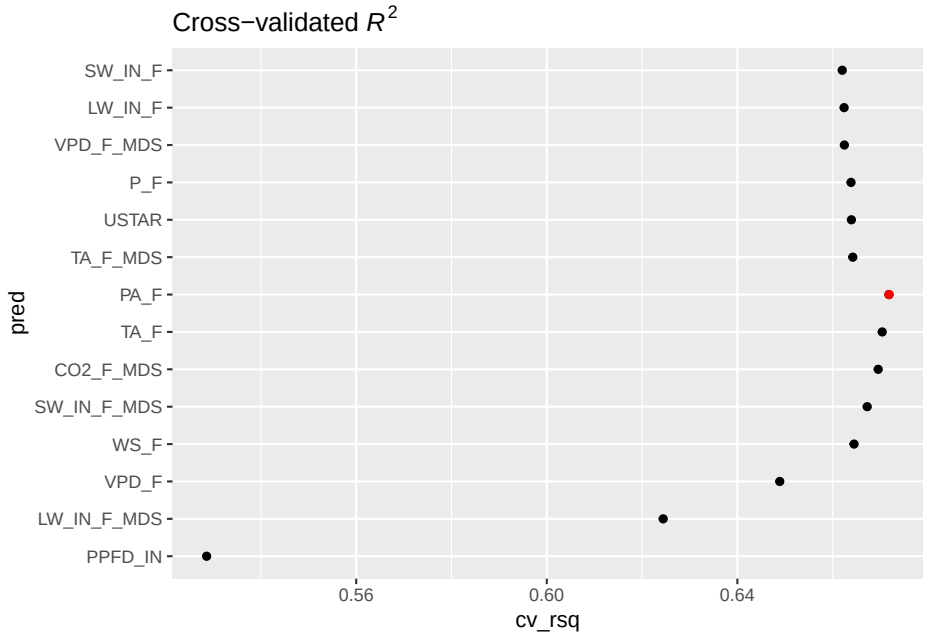
```
df_metrics$pred <-  factor(df_metrics$pred, levels = df_metrics$pred)

ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = rsq)) +
  geom_point(data = filter(df_metrics, rsq == max(rsq)), aes(x = pred, y = rsq), color = "red") +
  labs(title = expression(italic(R)^2)) +
  coord_flip()
```



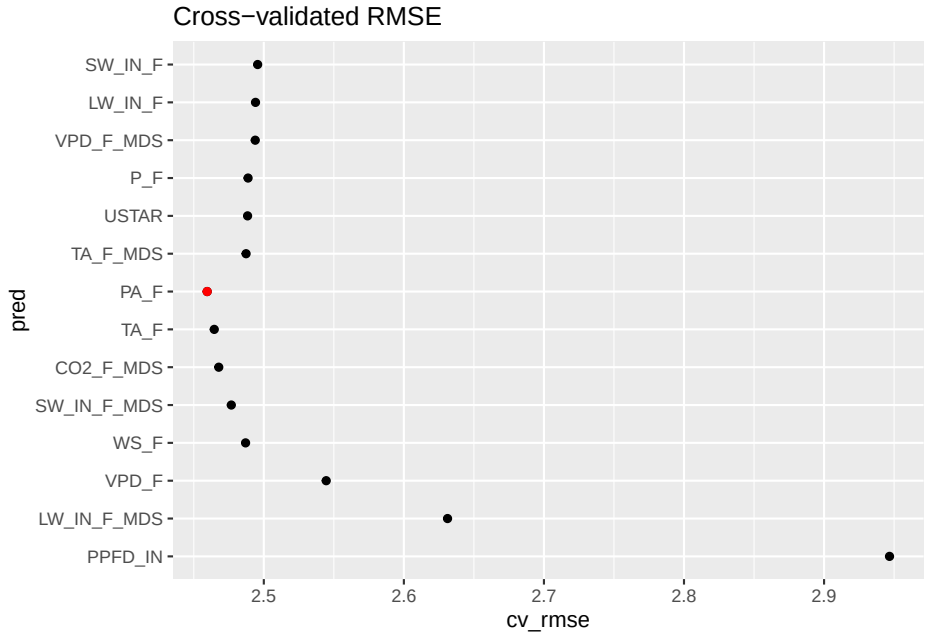

```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = adj_rsq)) +
  geom_point(data = filter(df_metrics, adj_rsq == max(adj_rsq)), aes(x = pred, y = adj_rsq), colo
  labs(title = expression(paste("Adjusted-", italic(R)^2))) +
  coord_flip()
```



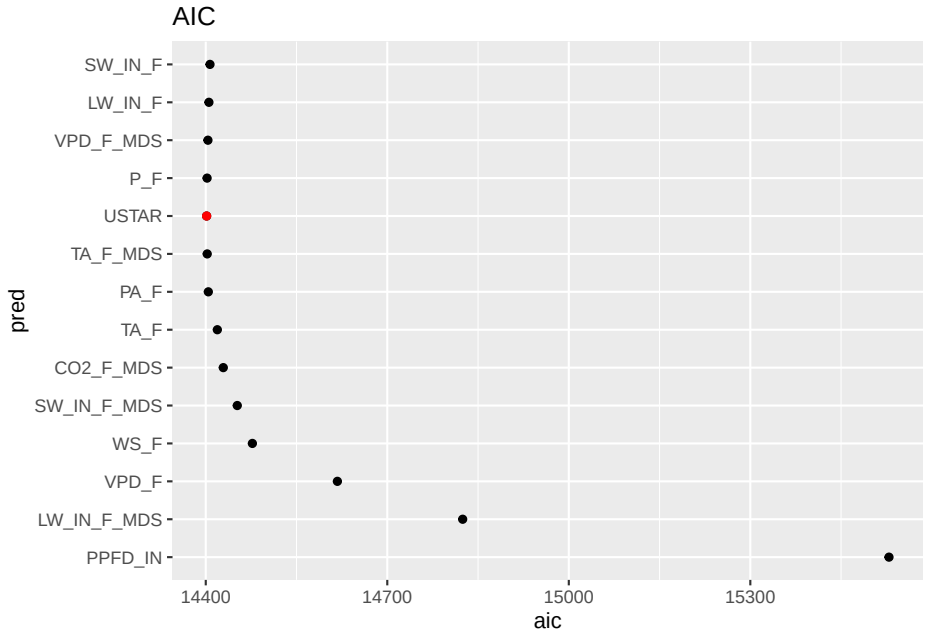

```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = cv_rsq)) +
  geom_point(data = filter(df_metrics, cv_rsq == max(cv_rsq)), aes(x = pred, y = cv_rsq
  labs(title = expression(paste("Cross-validated ", italic(R)^2))) +
  coord_flip()
```



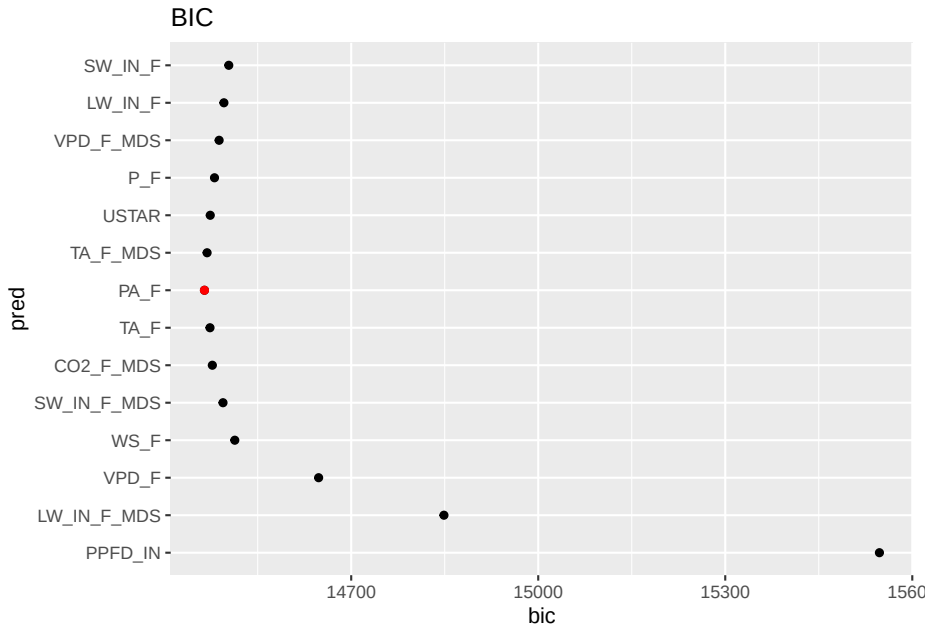
```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = cv_rmse)) +
  geom_point(data = filter(df_metrics, cv_rmse == min(cv_rmse)), aes(x = pred, y = cv_rmse), colo
  labs(title = expression(paste("Cross-validated ", RMSE))) +
  coord_flip()
```




```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = aic)) +
  geom_point(data = filter(df_metrics, aic == min(aic)), aes(x = pred, y = aic), color = "red") +
  labs(title = "AIC")+
  coord_flip()
```




```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = bic)) +
  geom_point(data = filter(df_metrics, bic == min(bic)), aes(x = pred, y = bic), color
  labs(title = "BIC")+
  coord_flip()
```



- Observe which predictors are selected into the final model and which not. Why?

Solution: Take the best model selected by BIC as an example, `VPD_F` is selected into the model and `VPD_F_MDS` is not. They won't enter the model both because they are highly correlated and when one is in, the other doesn't add information, which proves our expectation in the previous question.

- Whether it matters to the order of variables entering into the model which metric (i.e. AIC, BIC or R^2) is used? Why?

Solution: No, it doesn't matter. For each inner loop, when deciding which variable is sequentially chosen, we compare models of the same level (i.e. same number of variables). In this sense, model complexities for all the models are the same, and AIC/BIC only compares the goodness of fit part, which works the same as R^2.

- Discuss your results. Which metric do you trust most? Why? Try out your algorithm without evaluating the cross-validated R^2. Do you get a computational efficiency gain? Which metric comes closest to the cross-validated R^2 in terms of selecting the same number predictors?

Solution: Cross-validated R^2 is the "gold-standard" for assessing generalisability. However, it is computationally costly and may not provide robust results when the data set is small. Since AIC and BIC both penalise additional predictors, they both don't select the full model with 14 variables as the best model (However, it could happen that the best model selected by AIC/BIC is the full model). The BIC selects for the same number of predictors as the cross-validated R^2 and RMSE, but without the disadvantage of the computational costs. It provides a conservative and apparently robust estimate for the model generalisability. Note that the order of added predictors is determined by the simple R^2 and is therefore independent of the final metrics we use for choosing the best k.

9.5 Bonus: Stepwise regression out-of-the-box

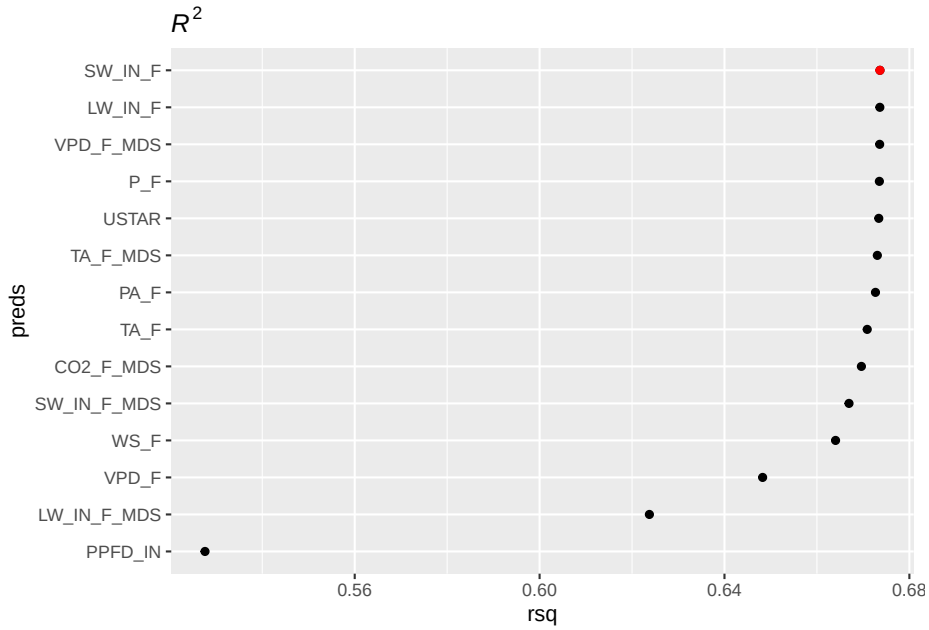
In R, you can also conduct the above variable selection procedures automatically. `regsubsets()` from leaps package provides a convenient way to do so. It does model selection by exhaustive search, including forward or backward stepwise regression.

- Do a stepwise forward regression using `regsubsets()`.
- Create a data frame with the same metrics as above (except cross-validated R^2).
- Visualise metrics as above.
- Are your result consistent?

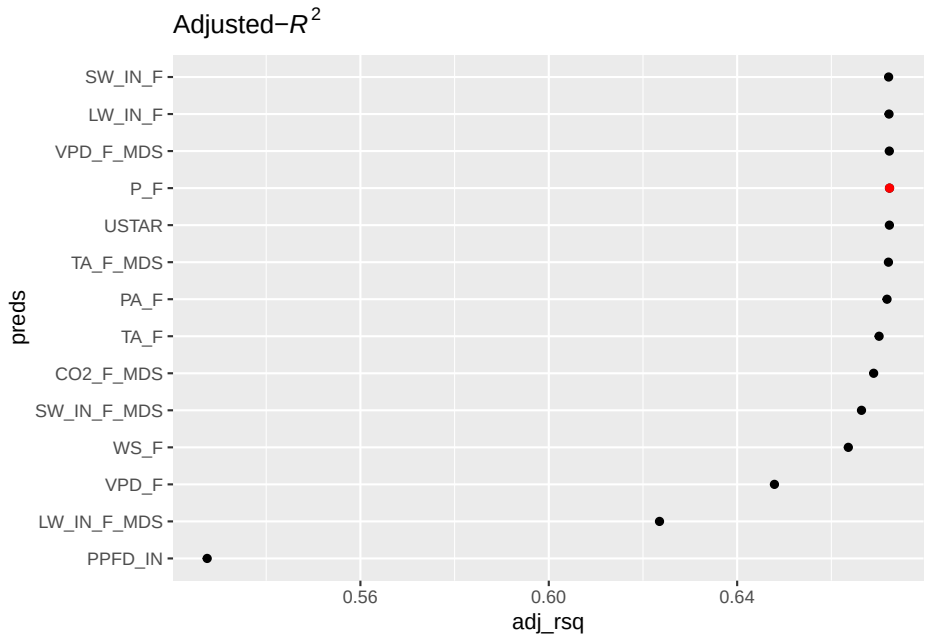
Hints:

- Specify stepwise *forward* by setting `method = "forward"`.
- Specify the number of predictors to examine, that is all fourteen, by setting `nvmax = 14`.
- AIC values of each step are stored in `summary(regfit.fwd)$cp` and BIC values of each step are in `summary(regfit.fwd)$bic`, etc.
- Get order in which predictors k are added (which corresponds to values returned by `summary(regfit.fwd)$bic`), by `all_predictors[regfit.fwd$vorder[2:15]-1]`. `vorder` is the order of variables entering into model. Note that `Intercept` is counted here as the first to enter into the model and should be removed.
- To avoid reordering of the list of variable names in plotting, change the type of variable names from "character" to "factor" by `preds_enter <- factor(preds_enter, levels = preds_enter)`

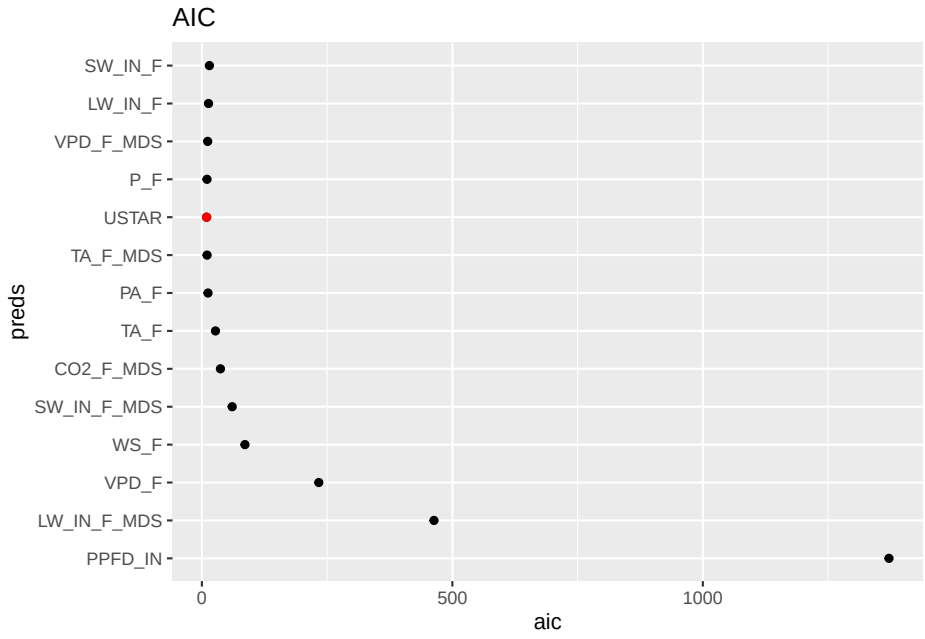
```
library(leaps)
regfit.fwd <- regsubsets(as.formula(paste(target, "~", paste(preds,collapse=" + "))), d
reg.summary <- summary(regfit.fwd)

# get variables in their order added to the model
preds_enter <- preds[regfit.fwd$vorder[2:15]-1]

## create metrics data frame
df_metrics_auto <- data.frame(
  preds = factor(preds_enter, levels = preds_enter),
  rsq = reg.summary$rsq,
  adj_rsq = reg.summary$adjr2,
  aic = reg.summary$cp,
  bic = reg.summary$bic
  )

ggplot() +
  geom_point(data = df_metrics_auto, aes(x = preds, y = rsq)) +
  geom_point(data = filter(df_metrics_auto, rsq == max(rsq)), aes(x = preds, y = rsq),
  labs(title = expression(italic(R)^2)) +
  coord_flip()
```



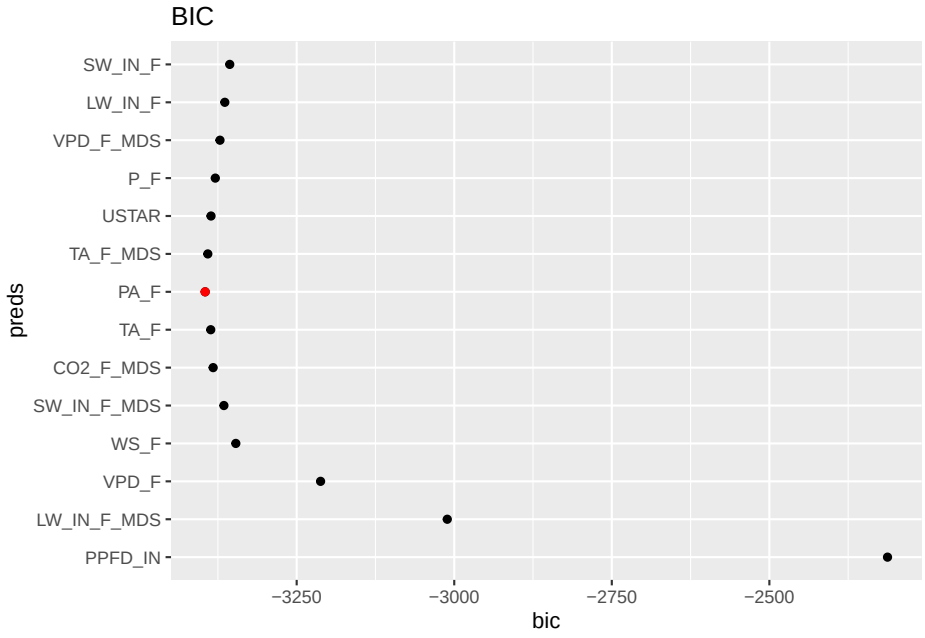

```
ggplot() +
  geom_point(data = df_metrics_auto, aes(x = preds, y = adj_rsq)) +
  geom_point(data = filter(df_metrics_auto, adj_rsq == max(adj_rsq)), aes(x = preds, y = adj_rsq)
  labs(title = expression(paste("Adjusted-", italic(R)^2))) +
  coord_flip()
```



```
ggplot() +
  geom_point(data = df_metrics_auto, aes(x = preds, y = aic)) +
  geom_point(data = filter(df_metrics_auto, aic == min(aic)), aes(x = preds, y = aic),
  labs(title = "AIC")+
  coord_flip()
```




```
ggplot() +
  geom_point(data = df_metrics_auto, aes(x = preds, y = bic)) +
  geom_point(data = filter(df_metrics_auto, bic == min(bic)), aes(x = preds, y = bic),
  labs(title = "BIC")+
  coord_flip()
```



9.6 Warm-up 2: Find the best single predictor

The first step of a stepwise forward regression is to find the single most powerful predictor in a univariate linear regression model for the target variable `GPP_NT_VUT_REF` among all fourteen available predictors in our data set (all except those of type `date` or `character`). Implement this first part of the search, using the definition of the stepwise-forward algorithm above. Remove all rows with at least one missing value before starting the predictor search.

- Which predictor achieves the highest R^2?
- What value is the R^2?
- Visualise R^2 by predictors, ordered by their respective R^2 values.
- Do you note a particular pattern? Which variables yield similar R^2? How do you expect them to be included in multivariate models of the subsequent steps in the stepwise forward regression?

Hints:

- Model structure:
  - The "counter" variables in the for loop can be provided as a vector, and the counter will sequentially take on the value of each element in that vector. For example: `for (var in all_predictors){ ... }`.
- Algorithm:

  - To record R^2 values for the different models, you may start by creating an empty vector (`vec <- c()`) before the loop and then sequentially add elements to that vector inside the loop (`vec <- c(vec, new_element)`). Alternatively, you can do something similar, but with a data frame (initialising with `df_rsq <- data.frame()` before the loop, and adding rows by `df_rsq <- bind_rows(df_rsq, data.frame(pred = predictor_name, rsq = rsq_result))` inside the loop).

  - A clever way how to construct formulas dynamically is described, for example in this stackoverflow post.

- Value retriving:

  - Extract the R^2 from the linear model object: `summary(fit_lin)[["r.squared"]]`

- Displaying:

  - Search for solutions for how to change the order of levels to be plotted yourself.

```
library(tidyverse)
## read CSV and drop missing values in one step
df <- read_csv("./data/ddf_for_08_application.csv") %>%
  drop_na()
```

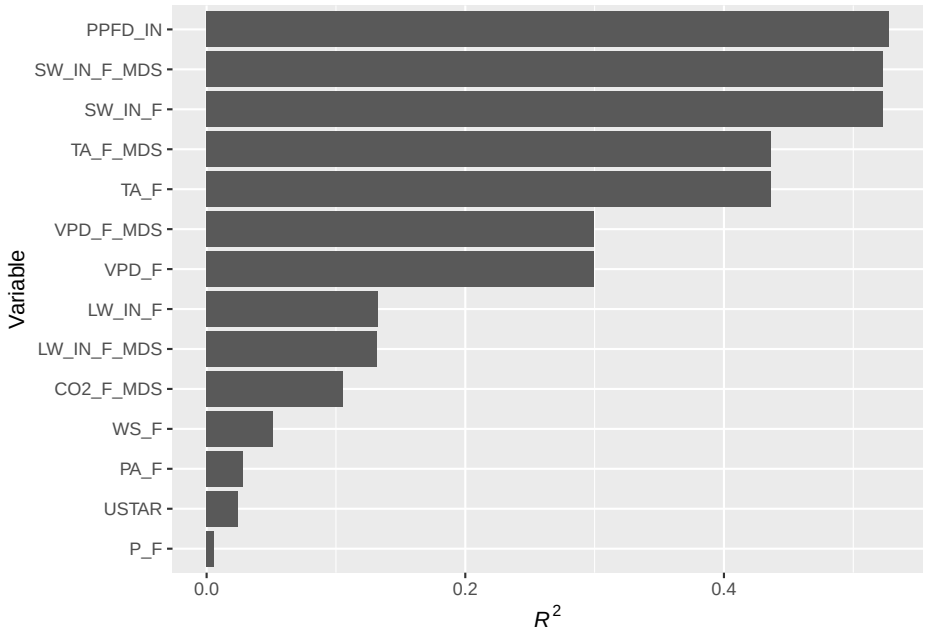
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   siteid = col_character(),
##   TIMESTAMP = col_date(format = ""),
##   TA_F = col_double(),
##   SW_IN_F = col_double(),
##   LW_IN_F = col_double(),
##   VPD_F = col_double(),
##   PA_F = col_double(),
##   P_F = col_double(),
##   WS_F = col_double(),
##   TA_F_MDS = col_double(),
##   SW_IN_F_MDS = col_double(),
##   LW_IN_F_MDS = col_double(),
##   VPD_F_MDS = col_double(),
##   CO2_F_MDS = col_double(),
##   PPFD_IN = col_double(),
##   GPP_NT_VUT_REF = col_double(),
##   USTAR = col_double()
## )
```

```
## specify target variable
target <- 'GPP_NT_VUT_REF'

## determine predictors as all except site ID, timestamp, and the target (should be 14)
preds <- df %>%
  dplyr::select(-siteid, -TIMESTAMP, -GPP_NT_VUT_REF) %>%
  names()

## initialise an empty data frame (necessary, because otherwise we cannot use bind_rows() below)
df_rsq <- data.frame()
# rsq_list <- c()  # alternative for vector

for (var in preds){

  ## create formula dynamically
  forml <- as.formula(paste(target, "~", var))

  ## fit linear model
  fit_lin <- lm(forml, data = df)

  ## extract R2 from linear model
  rsq <- summary(fit_lin)[["r.squared"]]

  ## add a row to the data frame that holds the results
  df_rsq <- bind_rows(df_rsq, data.frame(pred = var, rsq = rsq))

  # rsq_list <- c(rsq_list,rsq)  # alternative with vector
}

## print best single predictor and its R2
df_rsq %>%
  arrange(-rsq) %>%  # arrange by R2 in descending order (highest R2 on top)
  slice(1)           # print only the first row (with highest R2)
```

```
##      pred       rsq
## 1 PPFD_IN 0.5276123
```

```
library(ggplot2)

## alternative: determine the first variable to enter into our model
# preds[which.max(rsq_list)]

## use the data frame that holds the results for plotting
df_rsq %>%
  ggplot(aes(x = reorder(pred, rsq), y = rsq)) +
  geom_bar(stat = "identity") +
```

```
  labs(y = expression(italic(R)^2), x = "Variable") +
  coord_flip()
```



Solution: PPFD_IN achieves the highest R^2 value, and the corresponding R^2 values is 0.5276123.

We could see from the above plot that `LW_IN_F_MDS` and `LW_IN_F`, `VPD_F` and `VPD_F_MDS`, `SW_IN_F` and `SW_IN_F_MDS` share basically the same R^2 values. Those variables are highly corrected and are explaining the same amount of variance in the target variable.

In the subsequent steps of the stepwise forward regression, since each of the two variables are highly correlated with each other, it's likely that only one of them will be selected into our final model.

9.7 Full stepwise regression

Now, we take it to the next level and implement a full stepwise forward regression as described above. For each step (number of predictors k), record the following metrics: R^2, *adjusted-*R^2, the Akaike Information Criterion (AIC), (Bayesian Information Criterion) BIC, and the 5 time repeated 5-fold cross-validation R^2 and RMSE.

- Display the order in which predictors enter the model.

- Display a table with the metrics of all k steps, and the single variable, added at each step.

Hints:

- Model structure:
  - Recall what you learned in the breakout session, you may use the same idea on this task. Try to think of the blueprint (**pseudo-code**) first: how to go through different models in each forward step? how to store predictors added to the model and how to update candidate predictors?
- Algorithm:
  - A complication is that the set of predictors is sequentially complemented with each step of the outer loop. You may again use `vec <- list()` to create an empty vector, and then add elements to that vector by `vec <- c(vec, new_element)`.
  - It may be helpful to explicitly define a set of "candidate predictors" that may potentially be added to the model as a vector (e.g., `preds_candidate`), and define predictors retained in the model from the previous step in a separate vector (e.g., `preds_retained`). In each step, search through `preds_candidate`, select the best predictor, add it to `preds_retained` and remove it from `preds_candidate`.
  - At each step, record the metrics and store them in a data frame for later plots.
  - (As above) A clever way how to construct formulas dynamically is described, for example in this stackoverflow post.
  - The metrics for the k models are assessed *after* the order of added variables is determined. To be able to determine the metrics, the k models can be saved by constructing a list of models and sequentially add elements to that list (`mylist[[ name_new_element ]] <- new_element`). You can also fit the model again after determining which predictor worked best.
  - Your code will most certainly have bugs at first. To debug efficiently, write code first in a simple R script and use the debugging options in RStudio (see here).
- Value retriving
  - To get AIC and BIC value, use the base-R functions `AIC()` and `BIC()`.
  - To get the cross-validated R^2, use the caret function `train()` with "Rsquared" as the loss function, and `method = "lm"` (to fit a linear regression model). Then extract the value by

`trained_model$results$Rsquared`. The same applies for cross-validated RMSE.

- Displaying:
  - To display a table nicely as part of the RMarkdown html output, use the function `knitr::kable()`
  - To avoid reordering of the list of variable names in plotting, change the type of variable names from "character" to "factor" by `pred <- factor(pred, levels = pred)`

```
library(caret)  # need train() for cross-validation

## specify target variable (as above)
target <- 'GPP_NT_VUT_REF'

## determine predictors as all except site ID, timestamp, and the target (should be 14
preds <- df %>%
  dplyr::select(-siteid, -TIMESTAMP, -GPP_NT_VUT_REF) %>%
  names()

# This is the vector of candidate predictors to be added in the model. To begin with,
preds_candidate <- preds

# predictors retained in the model from the previous step. To begin with, is empty.
preds_retained <- c()

## work with lists as much as possible (more flexible!)
df_metrics <- data.frame()

## outer loop for k predictors
for (k_index in 1:length(preds)){

  # rsq_candidates <- c()
  df_rsq_candidates <- data.frame()
  linmod_candidates <- list()

  ## inner loop for single additional predictor
  for (ipred in preds_candidate){

    # variable vector (new variable + retained variables) used in regression
    pred_add <- c(preds_retained, ipred)

    # define formulate with newly-added predictor
    forml  <- as.formula(paste( target, '~', paste(pred_add, collapse = '+')))
```

```
    # fit linear model
    fit_lin <- lm(forml, data = df)

    # add model object to list, and name the element according to the added variable
    linmod_candidates[[ ipred ]] <- fit_lin

    # record metrics for all candidates
    rsq <- summary(fit_lin)[["r.squared"]]
    df_rsq_candidates <- bind_rows(df_rsq_candidates, data.frame(pred = ipred, rsq = rsq))  # whe
    # rsq_candidates <- c(rsq_candidates,  rsq)  # when storing R2 as a vector

  }

  ## get name of candidate predictor that achieved the highest R2.
  pred_add <- df_rsq_candidates %>%  # when storing R2 in a data frame
    arrange(desc(rsq)) %>%
    slice(1) %>%
    pull(pred) %>%
    as.character()

  # pred_add <- preds_candidate[ which.max(rsq_candidates) ]   # when storing R2 as a vector

  ## add best predictors to retained predictors
  preds_retained <- c(preds_retained, pred_add)

  ## get the cross-validation R2
  forml  <- as.formula(paste( target, '~', paste(preds_retained, collapse = '+')))
  control <- trainControl(method = "cv", number = 5)
  set.seed(123)
  cv_modl <- train(forml, data = df, method = "lm", metric = "Rsquared", trControl = control)
  # record AIC and BIC and adjusted-R2 of the respective model
  df_metrics <- df_metrics %>%
    bind_rows(
      data.frame( pred = pred_add,
                  rsq = summary(linmod_candidates[[ pred_add ]])[["r.squared"]],
                  adj_rsq = summary(linmod_candidates[[ pred_add ]])[["adj.r.squared"]],
                  cv_rsq = cv_modl$results$Rsquared,
                  cv_rmse = cv_modl$results$RMSE,
                  aic = AIC(linmod_candidates[[ pred_add ]]),
                  bic = BIC(linmod_candidates[[ pred_add ]])
      )
    )


  # remove the selected variable from the candidate variable list
```

```
  preds_candidate <- preds_candidate[-which(preds_candidate == pred_add)]
  # preds_candidate <- setdiff(preds_candidate,pred_add)  # alternative

}

data.frame(df_metrics$pred) # order in which variables enter the model
```

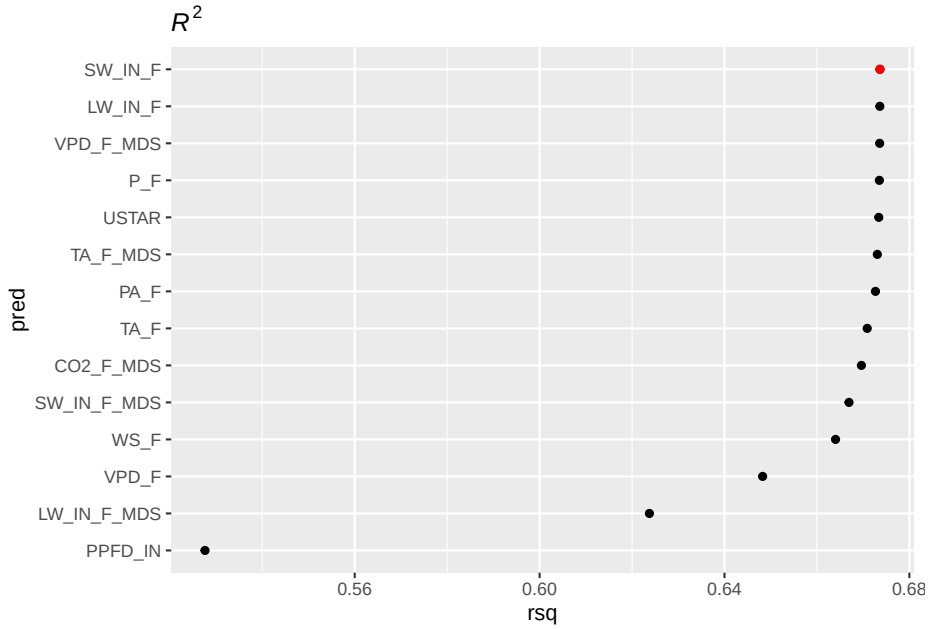
```
##    df_metrics.pred
## 1          PPFD_IN
## 2      LW_IN_F_MDS
## 3            VPD_F
## 4             WS_F
## 5      SW_IN_F_MDS
## 6        CO2_F_MDS
## 7             TA_F
## 8             PA_F
## 9         TA_F_MDS
## 10           USTAR
## 11             P_F
## 12       VPD_F_MDS
## 13         LW_IN_F
## 14         SW_IN_F
```

```
df_metrics %>% knitr::kable()
```

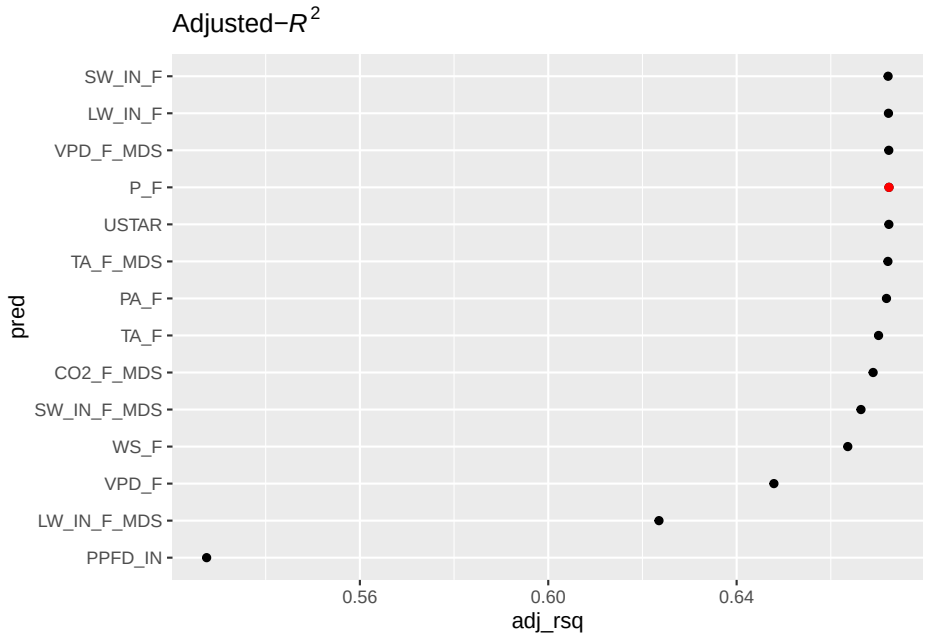
pred	rsq	adj_rsq	cv_rsq	cv_rmse	aic	bic
PPFD_IN	0.5276123	0.5274601	0.5286098	2.946732	15529.18	15547.30
LW_IN_F_MDS	0.6237535	0.6235110	0.6244311	2.631186	14824.62	14848.78
VPD_F	0.6482531	0.6479128	0.6488913	2.544541	14617.55	14647.76
WS_F	0.6640335	0.6636000	0.6644851	2.486970	14477.03	14513.28
SW_IN_F_MDS	0.6669387	0.6664013	0.6672468	2.476793	14452.07	14494.35
CO2_F_MDS	0.6696209	0.6689811	0.6695521	2.467823	14428.96	14477.28
TA_F	0.6708789	0.6701350	0.6703978	2.464609	14419.11	14473.48
PA_F	0.6726644	0.6718185	0.6718171	2.459555	14404.22	14464.63
TA_F_MDS	0.6730699	0.6721192	0.6642220	2.487336	14402.37	14468.82
USTAR	0.6733802	0.6723246	0.6639334	2.488412	14401.42	14473.91
P_F	0.6735098	0.6723487	0.6638548	2.488747	14402.19	14480.72
VPD_F_MDS	0.6735755	0.6723086	0.6624537	2.493879	14403.57	14488.14
LW_IN_F	0.6736125	0.6722398	0.6623996	2.494079	14405.22	14495.83
SW_IN_F	0.6736372	0.6721585	0.6620120	2.495602	14406.98	14503.63

- Visualise all metrics as a function of the number of predictors (add labels for the variable names of the added predictor). Highlight the best-performing model based on the respective metric. How many predictors are in the best performing model, when assessed based on each metric?

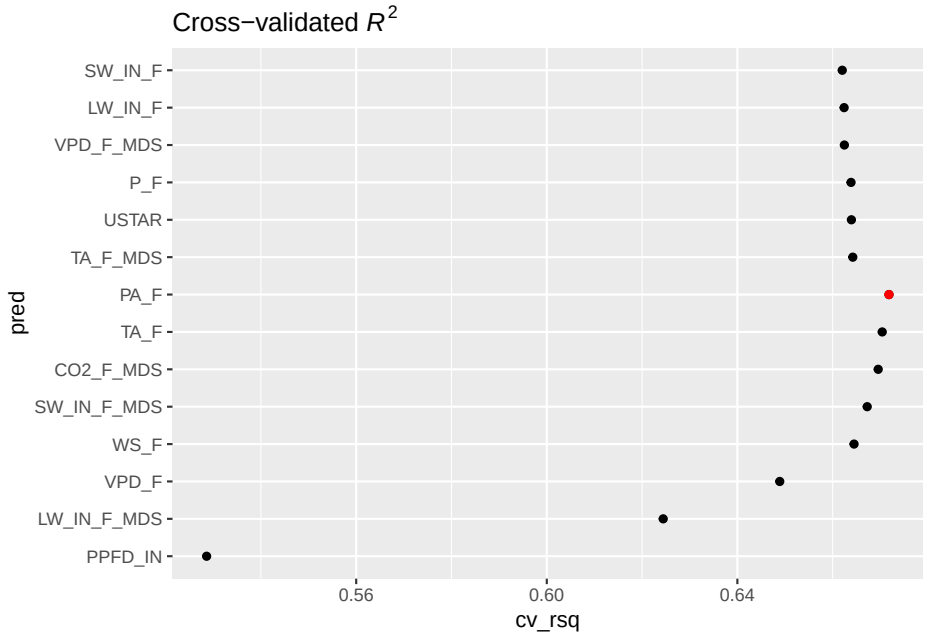
```
library(ggplot2)

df_metrics$pred <-  factor(df_metrics$pred, levels = df_metrics$pred)

ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = rsq)) +
  geom_point(data = filter(df_metrics, rsq == max(rsq)), aes(x = pred, y = rsq), color = "red") +
  labs(title = expression(italic(R)^2)) +
  coord_flip()
```



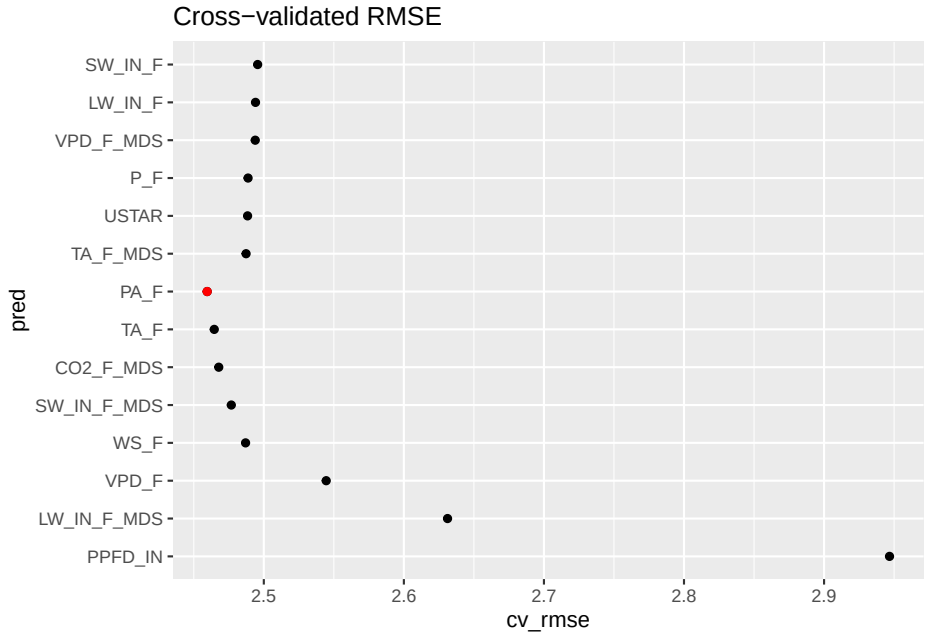

```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = adj_rsq)) +
  geom_point(data = filter(df_metrics, adj_rsq == max(adj_rsq)), aes(x = pred, y = adj_rsq), colo
  labs(title = expression(paste("Adjusted-", italic(R)^2))) +
  coord_flip()
```



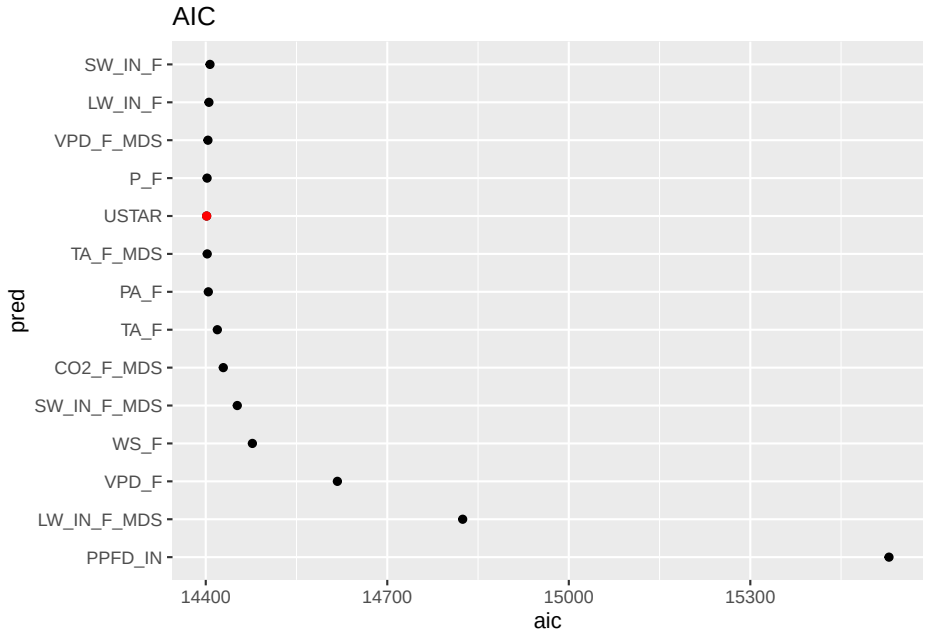

```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = cv_rsq)) +
  geom_point(data = filter(df_metrics, cv_rsq == max(cv_rsq)), aes(x = pred, y = cv_rsq
  labs(title = expression(paste("Cross-validated ", italic(R)^2))) +
  coord_flip()
```



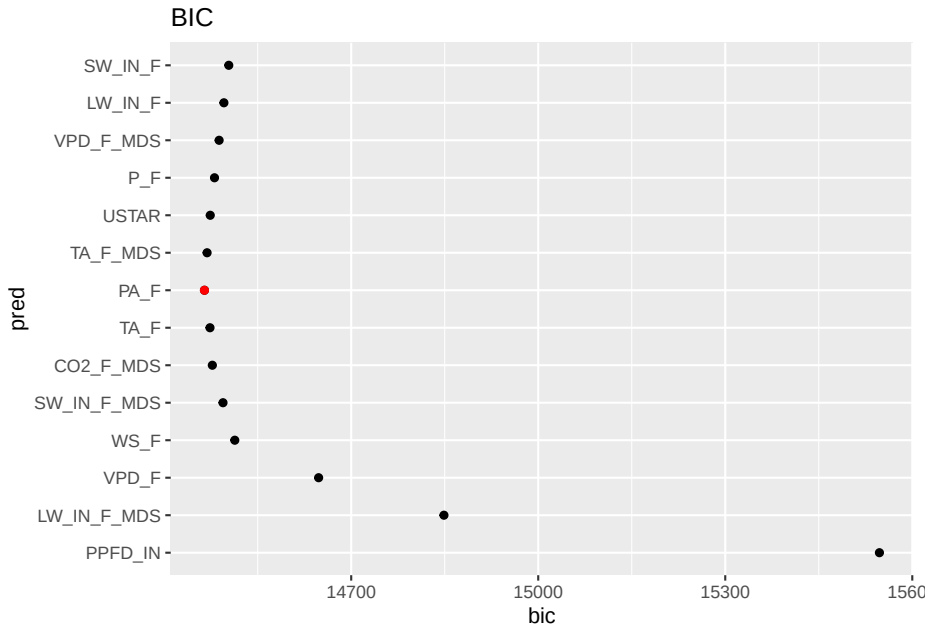
```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = cv_rmse)) +
  geom_point(data = filter(df_metrics, cv_rmse == min(cv_rmse)), aes(x = pred, y = cv_rmse), colo
  labs(title = expression(paste("Cross-validated ", RMSE))) +
  coord_flip()
```




```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = aic)) +
  geom_point(data = filter(df_metrics, aic == min(aic)), aes(x = pred, y = aic), color = "red") +
  labs(title = "AIC")+
  coord_flip()
```




```
ggplot() +
  geom_point(data = df_metrics, aes(x = pred, y = bic)) +
  geom_point(data = filter(df_metrics, bic == min(bic)), aes(x = pred, y = bic), color
  labs(title = "BIC")+
  coord_flip()
```



- Observe which predictors are selected into the final model and which not. Why?

Solution: Take the best model selected by BIC as an example, `VPD_F` is selected into the model and `VPD_F_MDS` is not. They won't enter the model both because they are highly correlated and when one is in, the other doesn't add information, which proves our expectation in the previous question.

- Whether it matters to the order of variables entering into the model which metric (i.e. AIC, BIC or R^2) is used? Why?

Solution: No, it doesn't matter. For each inner loop, when deciding which variable is sequentially chosen, we compare models of the same level (i.e. same number of variables). In this sense, model complexities for all the models are the same, and AIC/BIC only compares the goodness of fit part, which works the same as R^2.

- Discuss your results. Which metric do you trust most? Why? Try out your algorithm without evaluating the cross-validated R^2. Do you get a computational efficiency gain? Which metric comes closest to the cross-validated R^2 in terms of selecting the same number predictors?

Solution: Cross-validated R^2 is the "gold-standard" for assessing generalisability. However, it is computationally costly and may not provide robust results when the data set is small. Since AIC and BIC both penalise additional predictors, they both don't select the full model with 14 variables as the best model (However, it could happen that the best model selected by AIC/BIC is the full model). The BIC selects for the same number of predictors as the cross-validated R^2 and RMSE, but without the disadvantage of the computational costs. It provides a conservative and apparently robust estimate for the model generalisability. Note that the order of added predictors is determined by the simple R^2 and is therefore independent of the final metrics we use for choosing the best k.

9.8 BONUS Stepwise regression out-of-the-box

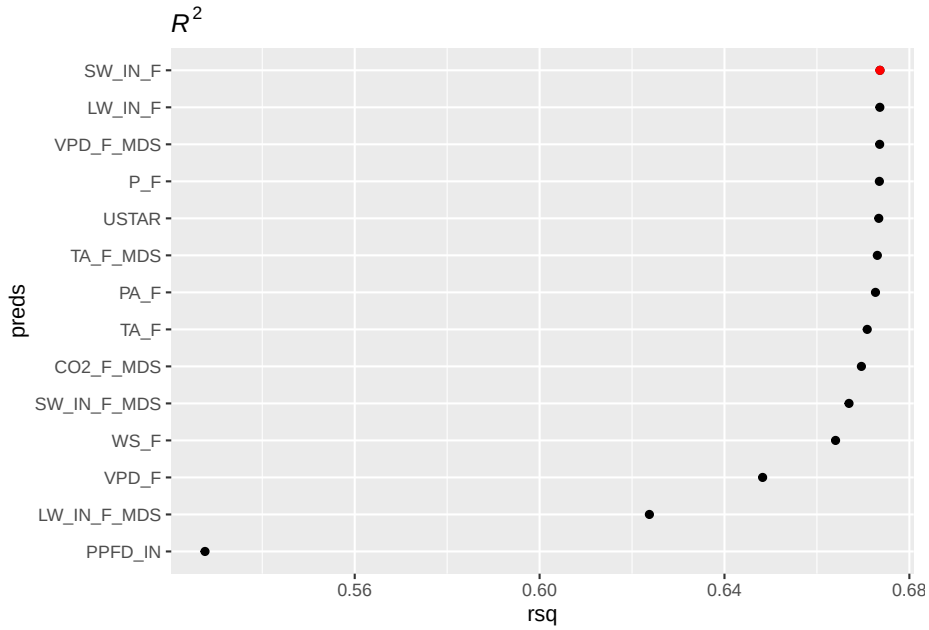
In R, you can also conduct the above variable selection procedures automatically. `regsubsets()` from leaps package provides a convenient way to do so. It does model selection by exhaustive search, including forward or backward stepwise regression.

- Do a stepwise forward regression using `regsubsets()`.
- Create a data frame with the same metrics as above (except cross-validated R^2).
- Visualise metrics as above.
- Are your result consistent?

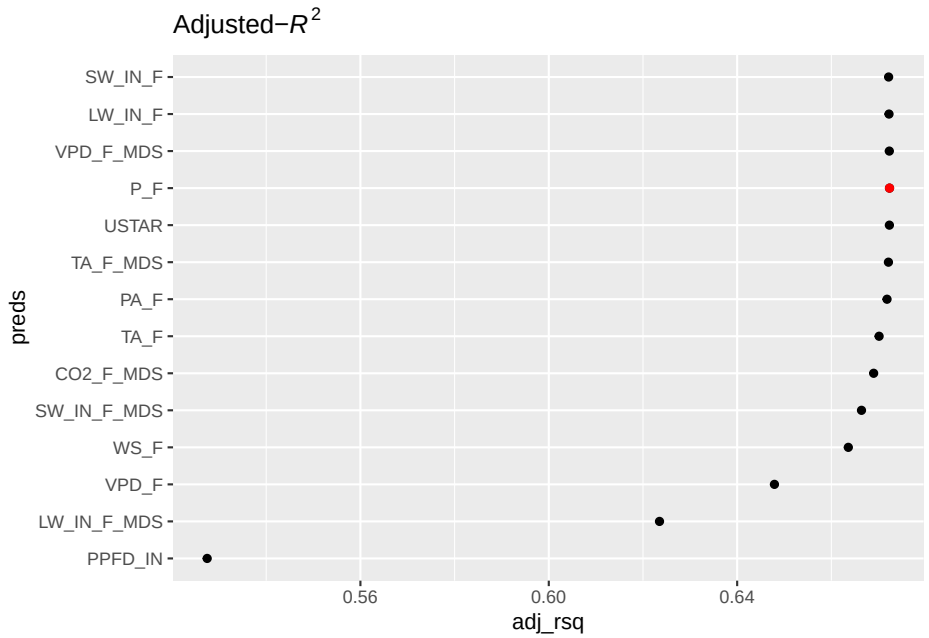
Hints:

- Specify stepwise *forward* by setting `method = "forward"`.
- Specify the number of predictors to examine, that is all fourteen, by setting `nvmax = 14`.
- AIC values of each step are stored in `summary(regfit.fwd)$cp` and BIC values of each step are in `summary(regfit.fwd)$bic`, etc.
- Get order in which predictors k are added (which corresponds to values returned by `summary(regfit.fwd)$bic`), by `all_predictors[regfit.fwd$vorder[2:15]-1]`. `vorder` is the order of variables entering into model. Note that `Intercept` is counted here as the first to enter into the model and should be removed.
- To avoid reordering of the list of variable names in plotting, change the type of variable names from "character" to "factor" by `preds_enter <- factor(preds_enter, levels = preds_enter)`

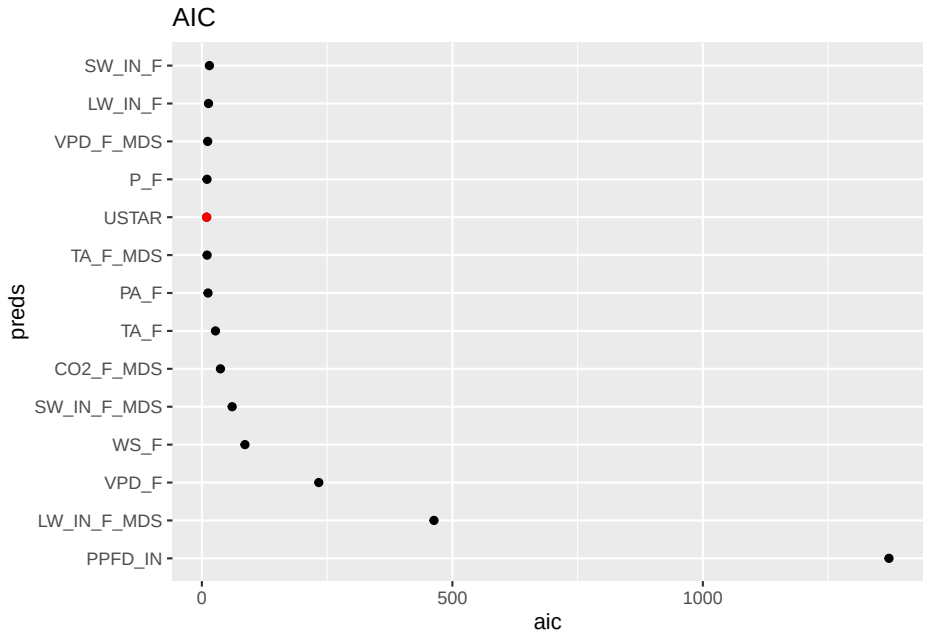
```
library(leaps)
regfit.fwd <- regsubsets(as.formula(paste(target, "~", paste(preds,collapse=" + "))), d
reg.summary <- summary(regfit.fwd)

# get variables in their order added to the model
preds_enter <- preds[regfit.fwd$vorder[2:15]-1]

## create metrics data frame
df_metrics_auto <- data.frame(
  preds = factor(preds_enter, levels = preds_enter),
  rsq = reg.summary$rsq,
  adj_rsq = reg.summary$adjr2,
  aic = reg.summary$cp,
  bic = reg.summary$bic
  )

ggplot() +
  geom_point(data = df_metrics_auto, aes(x = preds, y = rsq)) +
  geom_point(data = filter(df_metrics_auto, rsq == max(rsq)), aes(x = preds, y = rsq),
  labs(title = expression(italic(R)^2)) +
  coord_flip()
```



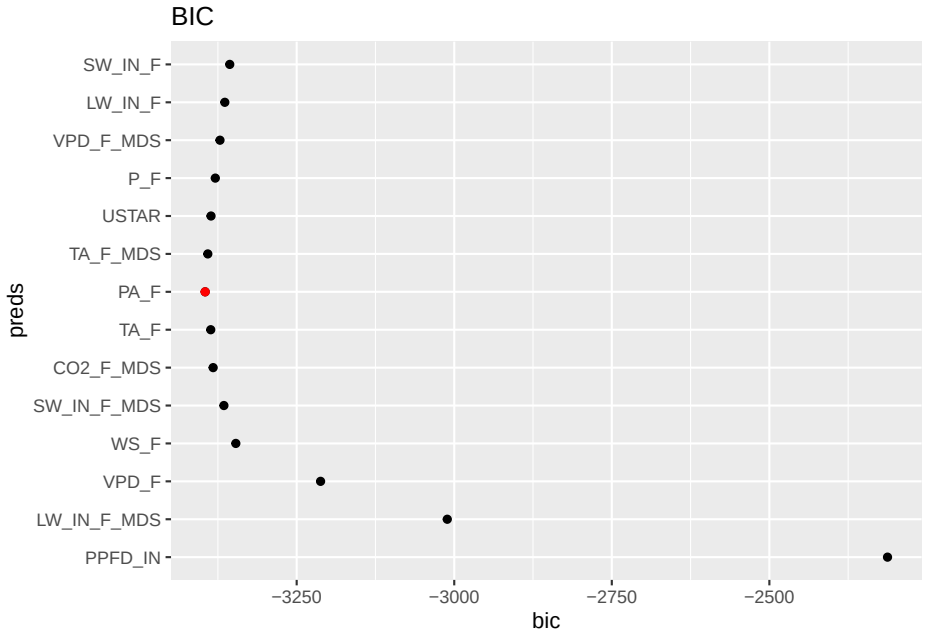

```
ggplot() +
  geom_point(data = df_metrics_auto, aes(x = preds, y = adj_rsq)) +
  geom_point(data = filter(df_metrics_auto, adj_rsq == max(adj_rsq)), aes(x = preds, y = adj_rsq)
  labs(title = expression(paste("Adjusted-", italic(R)^2))) +
  coord_flip()
```



```
ggplot() +
  geom_point(data = df_metrics_auto, aes(x = preds, y = aic)) +
  geom_point(data = filter(df_metrics_auto, aic == min(aic)), aes(x = preds, y = aic),
  labs(title = "AIC")+
  coord_flip()
```




```
ggplot() +
  geom_point(data = df_metrics_auto, aes(x = preds, y = bic)) +
  geom_point(data = filter(df_metrics_auto, bic == min(bic)), aes(x = preds, y = bic),
  labs(title = "BIC")+
  coord_flip()
```


BIC
SW_IN_F
LW_IN_F
VPD_F_MDS
P_F
USTAR
TA_F_MDS
PA_F
TA_F
CO2_F_MDS
SW_IN_F_MDS
WS_F
VPD_F
LW_IN_F_MDS
PPFD_IN
preds
−3250
−3000
−2750
−2500
bic

Chapter 10

XXX

Chapter 11

XXX

Chapter 12

XXX

Chapter 13

XXX

Chapter 14

XXX

Chapter 15

XXX

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.21.