Species Distribution Modelling using MAXENT

Modelling your species habitat suitability under present and future climate conditions

Practical Manual, 2020

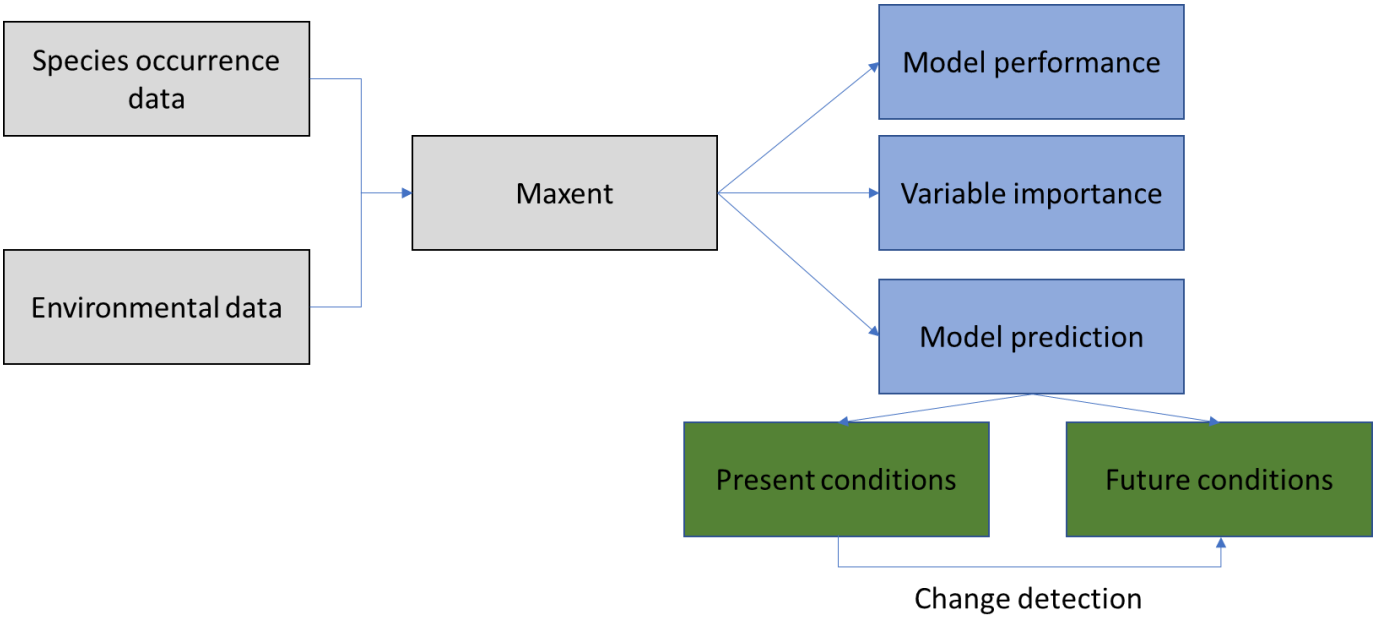
Nuno César de Sá (n.g.cesar.sa@cml.leidenuniv.nl)
Rosaleen March, PhD (r.g.march@cml.leidenuniv.nl)

Found a "bug"? Let us know!

Contents:

Introductory notes:

This manual represents the example that the tutors will show during the class. You should **adapt it to your chosen species** and folder structure. This manual should be enough to give you an example of all the options and alternatives available to you for setting up your own experiment and then proceed for your final report. The next figure is a general overview of the main steps but do notice that each step has multiple steps within:



During this example, we will go through each of the above steps. *The* **species selected is the *Rhinolophys Euryale***, the Mediterranean horseshoe bat. As the name implies, it is a type of bat characteristic of the Mediterranean region, thus is inhabits in a region with dry summers and wet winters. Still these bats live in caves and near wooded areas, so that might play a role in their distribution. In this tutorial we will explore how the species distribution is going to change in the coming 50 years if we maintain a "business as usual" attitude regarding climate change mitigation.

In previous lectures you have selected (or has been selected for you) a species. During the course Systematics in Biodiversity you have been working with different aspects of that same species. **In this case it is not different: this exercise should be made with that species** – so you need to adapt from the examples given in this tutorial.

Setting up your working environment:

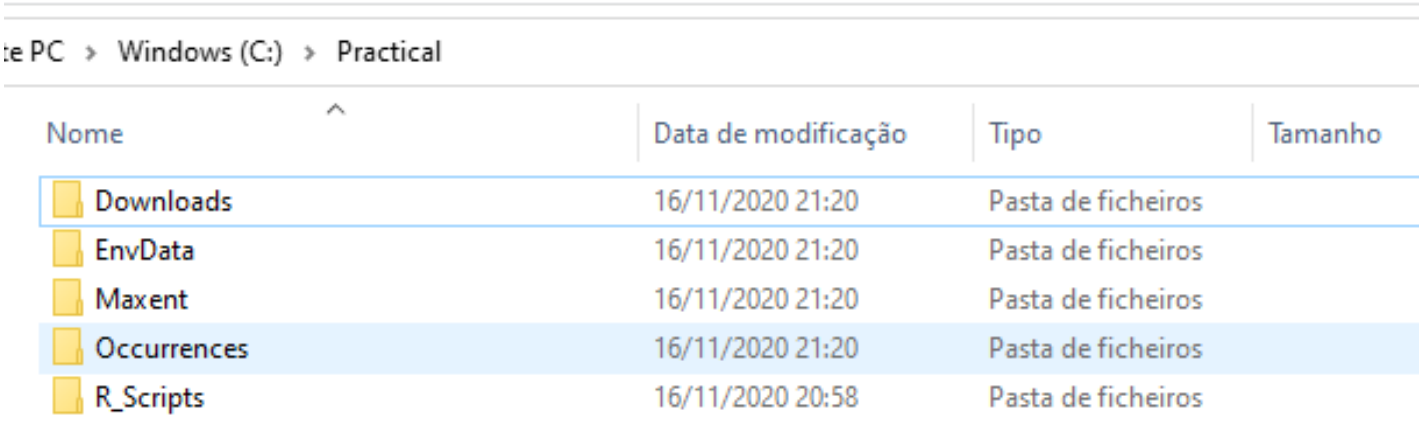
The script for this section is named **00_SettingUpWorkspace**. Open and investigate it.

```
dir.create("C:/Practical/")
setwd("C:/Practical")
#from now on the base folder is c:/Practical
```

```
#first tier of folders
dir.create("./Downloads")
dir.create("./Occurrences")
dir.create("./EnvData")
dir.create("./Maxent")


#Second tier
dir.create("./Occurrences/Shapefiles")
dir.create("./Occurrences/Tables")

dir.create("./EnvData/AOI")
dir.create("./EnvData/WLD")

dir.create("./Maxent/EXP01")

#Third tier
dir.create("./EnvData/AOI/PresentAOI/")
dir.create("./EnvData/AOI/FutureAOI/") #NOTICE - you might have multiple future scenarios

dir.create("./EnvData/WLD/PresentWLD/")
dir.create("./EnvData/WLD/FutureWLD/") #NOTICE - you might have multiple future scenarios
```

NOTICE: The last folder is named "future" only, but you can adapt these later to be the different environmental scenarios you would like to test. You can have as many "scenarios" to test as you want so it is ok to have different folders for each. Remember though that MAXENT will use the last subfolder name to identify the scenario and this means if you have scenarios with the same subfolder name they will be overwritten in the final output. **The recommendation is to add some extra indicator as I did above: "WLD" for raw global datasets and AOI for cropped areas.**

If you followed my suggestions, then the final folder structure should look something like this:



NOTICE: There are some specific **differences between running R in windows and R in MAC OS**. You can find their explanation and examples here and a description of shortcuts here. Let us know if you are having issues in adapting the code to MAC OS and we will help you

A common mistake in this section:

In R (and almost every other programming language) the symbols "\" and "/" have different meanings.

"\" is a special command that tells R to "exit" the regular execution during the compiling procedure. This exit command can be used to tell the computer that special characters are appearing, e.g. for &, % or $ to be correctly compiled they often must be written as \& \% or \%. **If you use the string "\Practical" in a function you will have the error: \P is an unrecognized escape…"** – R is telling that it does not know what to do with P as an escape character. **The correct way to do this in R is to either use two "\\" or "/". The following example shows this common error and how to avoid it:**

```
> list.files("c\Practical")
Error: '\P' is an unrecognized escape in character string starting ""c\P"
> list.files("c\\Practical")
character(0)
> list.files("c/Practical")
character(0)
>
```

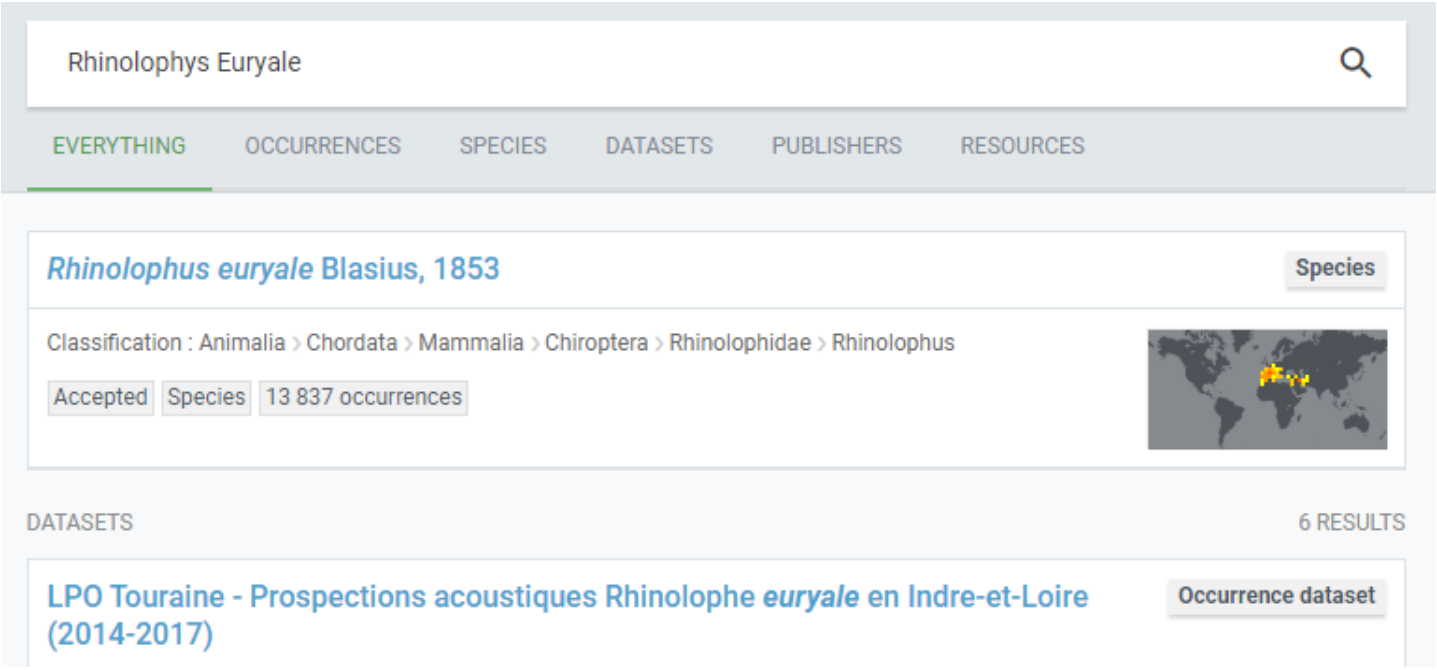
Occurrence data

Downloading occurrence data from GBIF

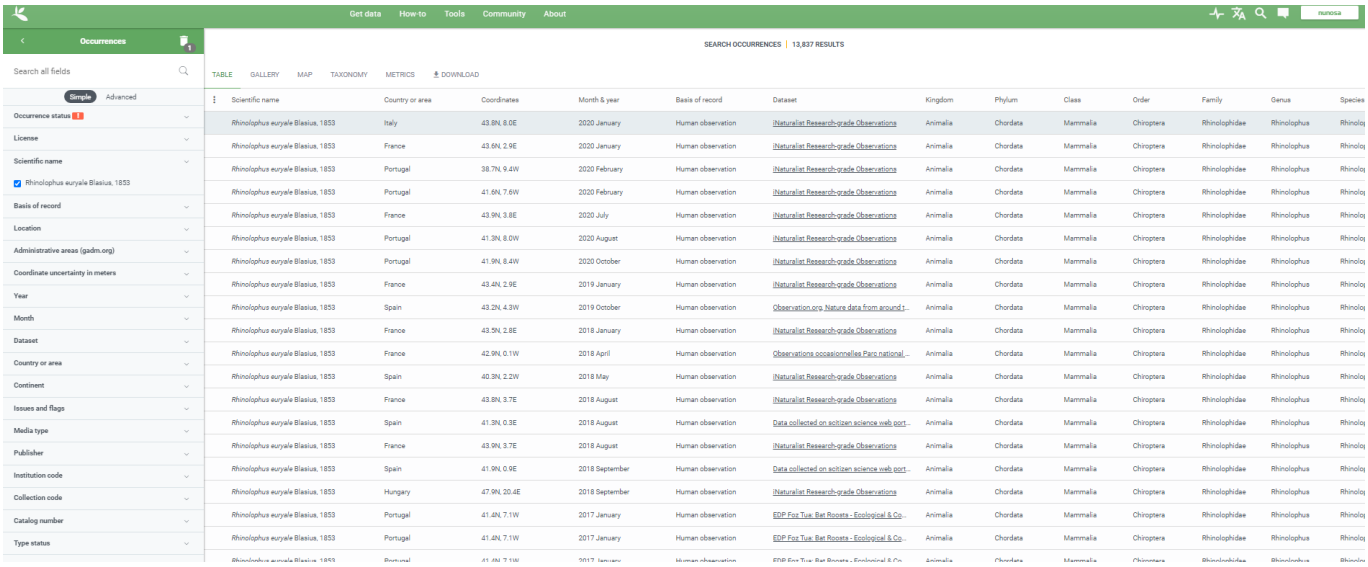
Data download and preparation was part of a previous GIS practical exercise, so we expect that you have already done this before. This section is here in any case, but it does not include an exploitation of the occurrence data in a GIS. **If you have prepared your data before, as we expect you to have done, skip this section but do confirm that your species occurrence ready to be used in MAXENT (section: *Preparing data for Maxent*)**

Step by step:

- Go to GBIF www.gbif.org
    - Create an account and login so you can download data
- Enter **your species name** on the search field.
    - For the purpose of example, we will model the distribution of the bat *Rhinolophys Euryale*



- Press "<number> occurrences" green button on the top right to proceed to the data download section.
- The next section will show you more information about the actual data that GBIF has collected. And more filtering options are given on the left panel.

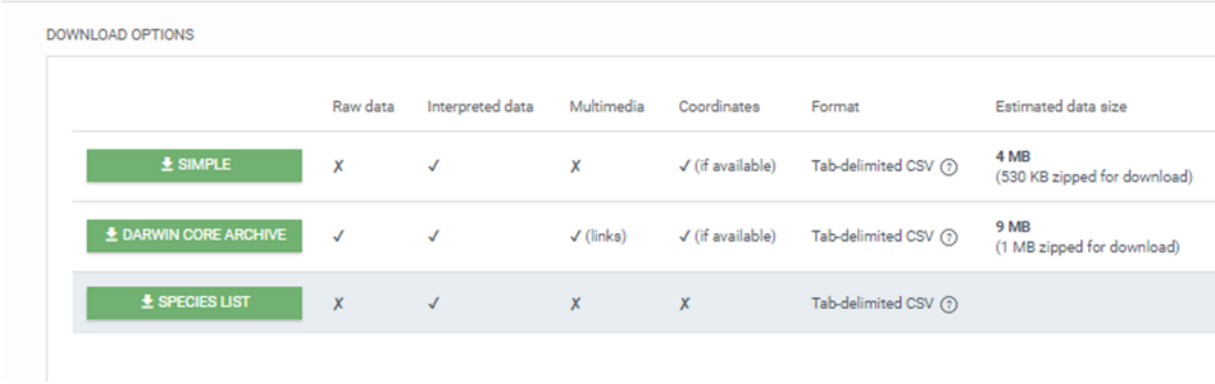


- **Explore these filtering options and how they work.** What they will do is help you do a quick cleaning of the data available in the repository.
    - Keep track of the filtering options you have selected for your species and use that information on your report.h
- For our case, we choose the following:
    - In **"occurrence status"** select "Present"
    - In **"Scientific name"** select the names that you are interested on.
    - In **"Location"** select "Including coordinates"

    - (Optional) In **"Basis of record"** you can see that some records will be "preserved specimens". This might indicate that they are being stored in a museum but often the specimen is stored there but the coordinates refer to the location where it was collected. (Often, only through GIS this is possible to address).

    - (Optional) In "Issues and flags" pay attention to any special issue (e.g. "Zero coordinate") and do not tick those options. Beware that in this case you need to activate the filter to select.

Once you are satisfied with the filters, press on the download button (top of the data table) and download the "simple" version. Once the data is processed, you will be notified by email.



Once you have received the email and completed the download, extract the data to the table folder (if you used the folder structure suggested before) and you can begin to explore what you have.

Ficheiro Editar Formatar Ver Ajuda

gbifID	datasetKey	occurrenceID	kingdom	phylum	class	order	family	genus	species	infraspecificEpithet	taxonRank	scientificName	verb
2979566300	040c5662-da76-4782-a48e-cdea1892d14c	http://bins.boldsystems.org/index.php/Public_RecordView?processid=IBICH052-19	Animalia		Chor								
2963920427	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/58371454	Animalia		Chordata		Mammalia						
2963804375	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/48095863	Animalia		Chordata		Mammalia						
2963788420	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/56401439	Animalia		Chordata		Mammalia						
2898523192	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/62592750	Animalia		Chordata		Mammalia						
2855022321	f946666e-67dc-4848-9fa8-2162f3559e33	f83452f5-d9c7-4098-bf1d-5e094b4303aa	Animalia		Chordata		Mammalia						

Preparing occurrence data for MAXENT

The data is delivered in "Comma separated values" (.csv) using "tabular" separators and points as decimals. Some excel version (and operating systems) will expect a different format and you will not be able to open the file just by clicking on it.

In general:

- "USA" format: decimals as points and commas to separate values
- "EU" format: decimals as commas and semicolons to separate values
- "Tabular" format: decimals as commas/points and spaces (tabular spaces) to separate values

Rhinolophus_euryale_csv0 - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

"species"	"longitude"	"latitude"
"Rhinolophus euryale"	10.261719	51.193676
"Rhinolophus euryale"	5.566667	50.633333
"Rhinolophus euryale"	5.566667	50.633333
"Rhinolophus euryale"	20.167	48.617
"Rhinolophus euryale"	20.75	48.516666
"Rhinolophus euryale"	20.887191	48.495193
"Rhinolophus euryale"	20.506927	48.467712
"Rhinolophus euryale"	20.542033	48.460844
"Rhinolophus euryale"	20.542033	48.460844
"Rhinolophus euryale"	20.542033	48.460844
"Rhinolophus euryale"	20.542033	48.460844
"Rhinolophus euryale"	20.542033	48.460844
"Rhinolophus euryale"	20.542033	48.460844
"Rhinolophus euryale"	-0.29907	48.20928
"Rhinolophus euryale"	-0.29907	48.20928
"Rhinolophus euryale"	-0.29907	48.20928

Rhinolophus_euryale_csv1 - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

```
"species","longitude","latitude"
"Rhinolophus euryale",10.261719,51.193676
"Rhinolophus euryale",5.566667,50.633333
"Rhinolophus euryale",5.566667,50.633333
"Rhinolophus euryale",20.167,48.617
"Rhinolophus euryale",20.75,48.516666
"Rhinolophus euryale",20.887191,48.495193
"Rhinolophus euryale",20.506927,48.467712
"Rhinolophus euryale",20.542033,48.460844
"Rhinolophus euryale",20.542033,48.460844
"Rhinolophus euryale",20.542033,48.460844
"Rhinolophus euryale",20.542033,48.460844
"Rhinolophus euryale",20.542033,48.460844
"Rhinolophus euryale",20.542033,48.460844
"Rhinolophus euryale",-0.29907,48.20928
"Rhinolophus euryale",-0.29907,48.20928
"Rhinolophus euryale",-0.29907,48.20928
```

Rhinolophus_euryale_csv2 - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

```
species;longitude;latitude
Rhinolophus euryale;10,261719;51,193676
Rhinolophus euryale;5,566667;50,633333
Rhinolophus euryale;5,566667;50,633333
Rhinolophus euryale;20,167;48,617
Rhinolophus euryale;20,75;48,516666
Rhinolophus euryale;20,887191;48,495193
Rhinolophus euryale;20,506927;48,467712
Rhinolophus euryale;20,542033;48,460844
Rhinolophus euryale;20,542033;48,460844
Rhinolophus euryale;20,542033;48,460844
Rhinolophus euryale;20,542033;48,460844
Rhinolophus euryale;20,542033;48,460844
Rhinolophus euryale;20,542033;48,460844
Rhinolophus euryale;-0,29907;48,20928
Rhinolophus euryale;-0,29907;48,20928
Rhinolophus euryale;-0,29907;48,20928
```

Converting between different .csv data formats:

- Importing in CSV Tab delimited format:
  - Microsoft Excel
    - Open a blank workbook
    - Go to data tab
    - Click button "from text/CSV" in the General External or "get and transform" data section
    - Select your CSV file
    - Follow the Text import wizard and adapt to your case.
    - Save your file with a different name
      - Recommended: add an indicator in the end of the file e.g. species_csv1.csv for **USA type data (as is the case of this tutorial)**

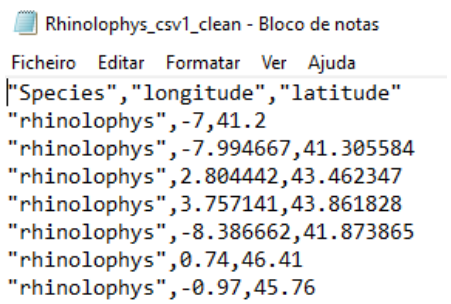
        - This will help you maintain your folder and data organized
    - Notepad:
        - Open in notepad and use the substitutions functions to correct the decimals and separators
    - Using R:
        - Adapt from example in file: Saving_CSV_files.R
        - Open GBIF file: Read the file using read.csv or read.table adapted to NA style
        - Save corrected file: Write the file opened before to your disk using: write.csv(US Style) or write.csv2(EU style) or write.table(Customizable) depending on what file type you want to save it to (remember, MAXENT will require US Style).

```
#set work directory
setwd("C:/Practical")

#read.csv <- NA style <- commas as separators, points as decimals
#read.csv <- EU style <- semicolon as separators, commas as decimals
#read.table <- allows you to change more parameters (read.csv and read.csv2 are special functions of the read.table function)

#first you load the .csv file that you want to convert
sp <- read.csv2("./Occurrences/Tables/Rhinolophys_csv2_clean.csv",header=T)

#and then you write it (write.csv or write.csv2)
write.csv(sp,"./Occurrences/Tables/Rhinolophys_csv1_clean.csv",row.names = F)
```

Prepare your data for MAXENT:

Once the previous details on the data format are sorted and excel is able to open it, you need to:

- Only have 3 columns: **species, longitude, latitude (follow this order precisely)**
    - Species: species name, no spaces (underscores not recommended) and make sure its only one species
        - E.g. *Rhinolophys Euryale* becomes Rhinolophys
    - Longitude and latitude are obvious
- **Save your file as .csv** but before saving make sure that:
    - Is there only 1 name for the species? If not, correct for this, its recommended to use a simple version of the name without any underscore: e.g. Rhinolophys
    - Are there missing numbers on the latitude/longitude rows?
    - Is there text on the latitude and longitude rows?
    - Some of these errors occurred because of problems or mis-formatting on the previous step, investigate if it is possible to correct them or not, otherwise remove the problematic rows.
- In the end, the .csv should look like this (when opened with the notepad):



When you are finished, save your file as .csv and give it an appropriate name with perhaps **a tag indicating the data format type or if there was any important detail.** Do not use "strange" characters when naming the file, such "&" or "#" or "_" or "/" because these might create conflicts later.

- If you excel saves the .csv file into a format different than the USA style, then you need to use one of the previous steps to convert it to USA style
- **If you cleaned your data using GIS (as you should have done in a previous practical)** and exported as table .txt or .csv, confirm it is USA style before proceeding

Here you could (should) load your data in a GIS (e.g. ArcGIS or QGIS) and explore your data further to find occurrence data that is out of place. **This data quality verification is actually part of the practical GIS exercise.**

Some of the more common problems with occurrence data from GBIF are:

- Data located in museums, botanical gardens, a zoo, or any other type of collections.
- Data in absurd locations: Sea (land animals), 0º latitude and 0º longitude, in unexpected continents.

These are the easy problems, but more complex problems can exist. For example, occurrences in locations where the species has been identified as exotic or invasive. What to do in these cases? Should "exotic" occurrences be excluded? (Dinis, 2020) Another example is having both citizen observation and Atlas data at the same time (César de Sá, 2019) . Often these problems the actual objective of the research.

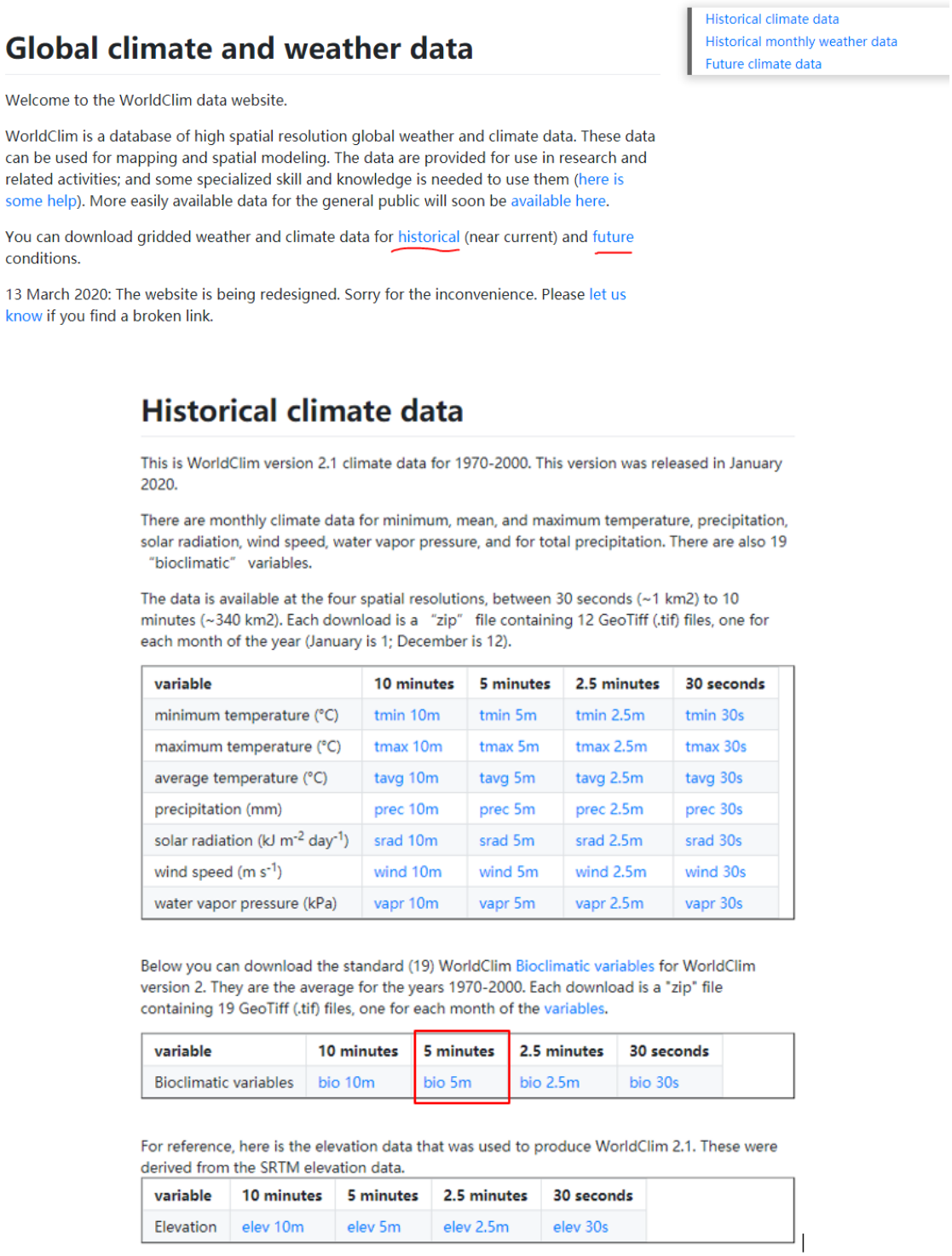
When you are finished with this analysis, just save those occurrence points into a **.csv in the NA format following the previous steps and proceed with your analysis**.

Environmental data

Downloading Environmental data

Step by Step:

- Go to https://worldclim.org/ and press on the "historical data"



Historical climate data
Historical monthly weather data
Future climate data

Global climate and weather data

Welcome to the WorldClim data website.

WorldClim is a database of high spatial resolution global weather and climate data. These data can be used for mapping and spatial modeling. The data are provided for use in research and related activities; and some specialized skill and knowledge is needed to use them (here is some help). More easily available data for the general public will soon be available here.

You can download gridded weather and climate data for historical (near current) and future conditions.

13 March 2020: The website is being redesigned. Sorry for the inconvenience. Please let us know if you find a broken link.

Historical climate data

This is WorldClim version 2.1 climate data for 1970-2000. This version was released in January 2020.

There are monthly climate data for minimum, mean, and maximum temperature, precipitation, solar radiation, wind speed, water vapor pressure, and for total precipitation. There are also 19 "bioclimatic" variables.

The data is available at the four spatial resolutions, between 30 seconds (~1 km2) to 10 minutes (~340 km2). Each download is a "zip" file containing 12 GeoTiff (.tif) files, one for each month of the year (January is 1; December is 12).

variable	10 minutes	5 minutes	2.5 minutes	30 seconds
minimum temperature (°C)	tmin 10m	tmin 5m	tmin 2.5m	tmin 30s
maximum temperature (°C)	tmax 10m	tmax 5m	tmax 2.5m	tmax 30s
average temperature (°C)	tavg 10m	tavg 5m	tavg 2.5m	tavg 30s
precipitation (mm)	prec 10m	prec 5m	prec 2.5m	prec 30s
solar radiation (kJ m^{-2} day^{-1})	srad 10m	srad 5m	srad 2.5m	srad 30s
wind speed (m s^{-1})	wind 10m	wind 5m	wind 2.5m	wind 30s
water vapor pressure (kPa)	vapr 10m	vapr 5m	vapr 2.5m	vapr 30s

Below you can download the standard (19) WorldClim Bioclimatic variables for WorldClim version 2. They are the average for the years 1970-2000. Each download is a "zip" file containing 19 GeoTiff (.tif) files, one for each month of the variables.

variable	10 minutes	5 minutes	2.5 minutes	30 seconds
Bioclimatic variables	bio 10m	bio 5m	bio 2.5m	bio 30s

For reference, here is the elevation data that was used to produce WorldClim 2.1. These were derived from the SRTM elevation data.

variable	10 minutes	5 minutes	2.5 minutes	30 seconds
Elevation	elev 10m	elev 5m	elev 2.5m	elev 30s

- **Download the 5 minutes spatial resolution Bioclimatic variables (equivalent to ~10km resolution at the equator).**

Bioclimatic variables

Bioclimatic variables are derived from the monthly temperature and rainfall values in order to generate more biologically meaningful variables. These are often used in species distribution modeling and related ecological modeling techniques. The bioclimatic variables represent annual trends (e.g., mean annual temperature, annual precipitation) seasonality (e.g., annual range in temperature and precipitation) and extreme or limiting environmental factors (e.g., temperature of the coldest and warmest month, and precipitation of the wet and dry quarters). A quarter is a period of three months (1/4 of the year).

They are coded as follows:

BIO1 = Annual Mean Temperature
BIO2 = Mean Diurnal Range (Mean of monthly (max temp - min temp))
BIO3 = Isothermality (BIO2/BIO7) (* 100)
BIO4 = Temperature Seasonality (standard deviation *100)
BIO5 = Max Temperature of Warmest Month
BIO6 = Min Temperature of Coldest Month
BIO7 = Temperature Annual Range (BIO5-BIO6)
BIO8 = Mean Temperature of Wettest Quarter
BIO9 = Mean Temperature of Driest Quarter
BIO10 = Mean Temperature of Warmest Quarter
BIO11 = Mean Temperature of Coldest Quarter
BIO12 = Annual Precipitation
BIO13 = Precipitation of Wettest Month
BIO14 = Precipitation of Driest Month
BIO15 = Precipitation Seasonality (Coefficient of Variation)
BIO16 = Precipitation of Wettest Quarter
BIO17 = Precipitation of Driest Quarter
BIO18 = Precipitation of Warmest Quarter
BIO19 = Precipitation of Coldest Quarter

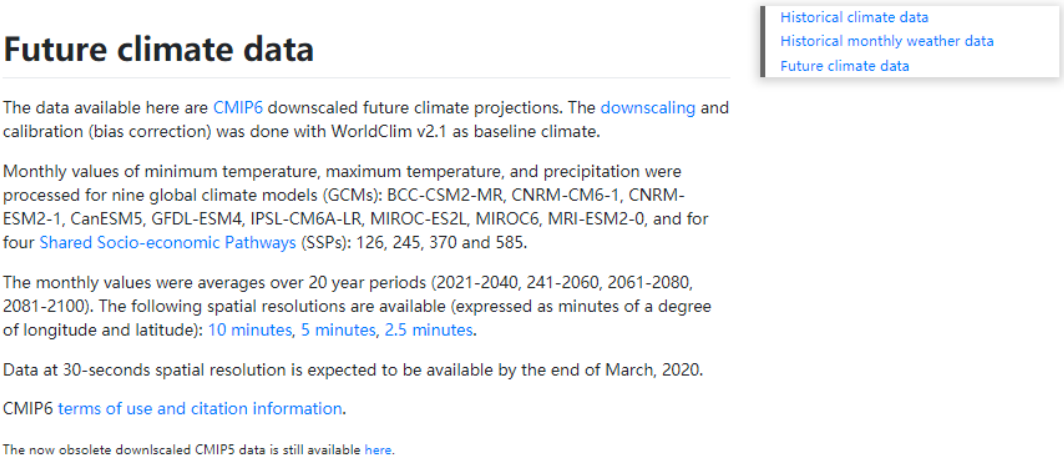
This scheme follows that of ANUCLIM, except that for temperature seasonality the standard deviation was used because a coefficient of variation does not make sense with temperatures between -1 and 1).

To create these values yourself, you can use the 'biovars' function in the R package dismo

Take note of the codes and meanings of each variable. Machine learning models always find a way to fit whatever data you give to them which means that in theory you simply add more environmental data to your model, and it will apparently improve. But of course, too much data implies that the model will fit to spurious relationships between datasets: remember correlation is not causality. **The variables you will be using on your model should provide a reasonable explanation of the species ecology.**

Let us download the future scenarios:

- Go to "Future Climate data" in the https://worldclim.org/ website:



Historical climate data
Historical monthly weather data
Future climate data

Future climate data

The data available here are CMIP6 downscaled future climate projections. The downscaling and calibration (bias correction) was done with WorldClim v2.1 as baseline climate.

Monthly values of minimum temperature, maximum temperature, and precipitation were processed for nine global climate models (GCMs): BCC-CSM2-MR, CNRM-CM6-1, CNRM-ESM2-1, CanESM5, GFDL-ESM4, IPSL-CM6A-LR, MIROC-ES2L, MIROC6, MRI-ESM2-0, and for four Shared Socio-economic Pathways (SSPs): 126, 245, 370 and 585.

The monthly values were averages over 20 year periods (2021-2040, 241-2060, 2061-2080, 2081-2100). The following spatial resolutions are available (expressed as minutes of a degree of longitude and latitude): 10 minutes, 5 minutes, 2.5 minutes.

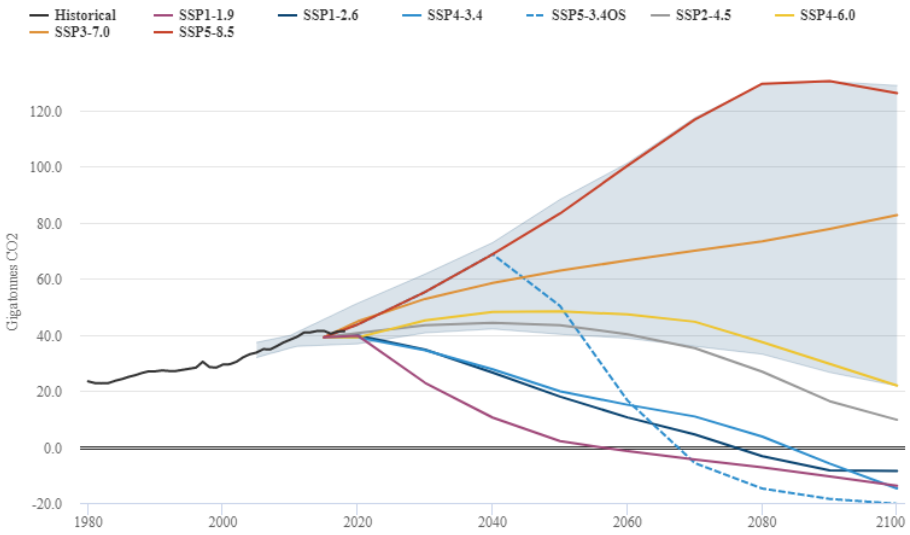
Data at 30-seconds spatial resolution is expected to be available by the end of March, 2020.

CMIP6 terms of use and citation information.

The now obsolete downlscaled CMIP5 data is still available here.

General Circulation Models (GCM) are advanced climate models that simulate the planet's atmosphere. **We highly insist that everyone uses** IPSL-CM6A-LR GCM **for this report** but to remember to provide some details about the GCM model based on the scientific publication.

Shared Socio-economic Pathways (SSP) represent different paths our planet can take regarding Carbon emissions: CarbonBrief CMIP6. ***Everyone should at least use scenario SSP3-7.0 ("a rocky road").*** But you are highly encouraged to use more scenarios!



- **Select the 5 minutes spatial resolution (same spatial resolution as the historical data!):**
  - On the next menu you will see multiple "time intervals". Here you are free to select whichever interval you are interested on but think on what would be a nice research question. **We opted for 2061-2080 interval period (so, ~50 years in the future).**

2061-2080

GCM	ssp126	ssp245	ssp370	ssp585
BCC-CSM2-MR	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc
CNRM-CM6-1	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc
CNRM-ESM2-1	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc
CanESM5	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc
GFDL-ESM4	tn, tx, pr, bc	--, --, --, --	tn, tx, pr, bc	-, -, pr, -
IPSL-CM6A-LR	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc
MIROC-ES2L	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc
MIROC6	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc
MRI-ESM2-0	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc	tn, tx, pr, bc

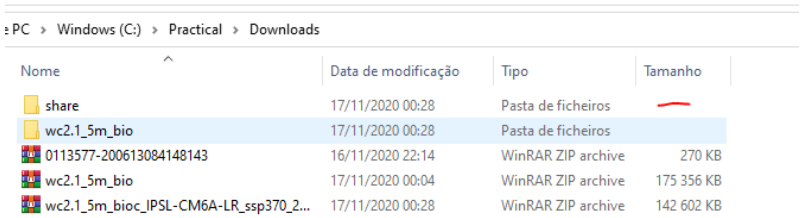
Mixing up different scenarios with different GCM and different time periods would make it hard to define what is being tested. As in any experimental setting, you must maintain some aspects constant.

Still, for your report explain SSP370 and (optionally but very encouraged) **explain how the species distribution is affected by this and other scenarios – it is always a good topic for the report.**

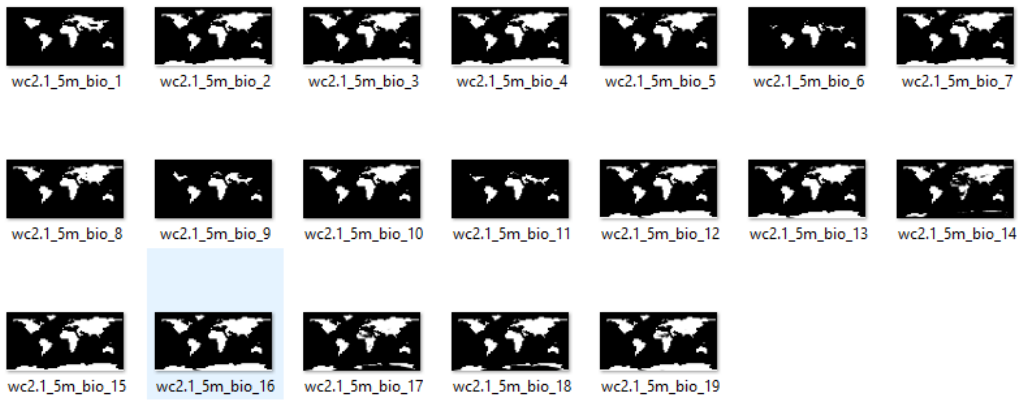
- You must use the same variable names in the future scenarios as in the historical scenarios, so:
  - **Download the Bioclimatic variables (bc) on the scenario(s) and time-period of choice**
  - **Unzip all data on the DOWNLOADS folder**
    - Next, we will use R to load the data and prepare it for MAXENT. This will include renaming, cropping and saving the files to the different folders we created earlier.
    - Remember, that you can also now create the specific folders on the EnvData subfolder related to each scenario you choose so you can use it later on MAXENT.

NOTICE: Worldclim has recently updated their side, so there might be some variation on how and where the data is delivered. Be aware that some steps might vary slightly for different scenarios and resolutions. But understand that the overall idea is the same.

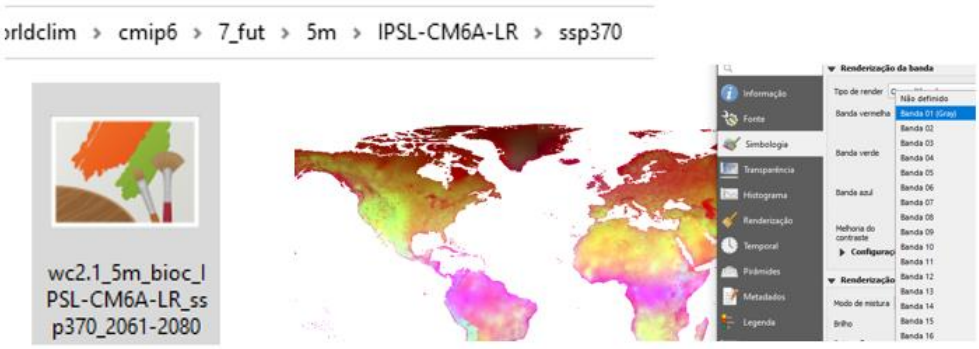
After the download is complete, and you've unzipped the files, your DOWNLOADS folder should look like this:



- Wc2.1_5m_bio is the
    - Worldclim historical data version 2.1
    - 5m resolution
    - Bioclimatic variables
    - **Historical data is provided** as multiple rasters where each is one of the bioclimatic variables



- Share is the SSP scenario folder (it is very deep and full of subfolders)
    - **The future scenarios are provided** as a single raster which has multiple layers inside



Since the historical and future data are provided in different forms (one as multiple rasters with a single layer and the other with a single raster with multiple layers) we must deal with both datasets using a different approach. The next steps will deal with that problem. – also remember that MAXENT expects environmental data as multiple single layer rasters in .asc (ascii) format.

Loading environmental data in R

The models should be trained in geographical areas that are like the occurrence data. This is to avoid overfitting behaviour of the model – of course – some species have a global distribution and therefore clipping or cropping that data becomes somewhat moot. Nevertheless, it is beneficial to learn how to do it. The R script for this section is: **01_CroppingEnvVariables.R**

Notice: The paths to the **folders used on this script refer to the folder structure shown before**. It is therefore important that you do any adaptation for it to work on the folder structure you created. **It is highly recommended that you use the latest version of R and R-Studio**. You will likely be prompted or requested to install RTools at some point. To do that, follow these instructions: https://cran.r-project.org/bin/windows/Rtools/

Step by Step:

- Open R-studio and open the script: **01_CroppingEnvVariables.R**
  - **Install the packages requested** on the top and any other dependencies that it requests or gives warning for.
  - Load the packages and confirm no error or warning is given.
- Run the next code snippet to load the variables:

```
#wordclim version: 2.0
#future variables scenario:IPSL-CM6A-LR - ssp370 - 61-80

#set work directory
setwd("C:/Practical")

#lists all historical Bioclimatic variables into two objects: one for the names and one for the path to the files
list.files("./Downloads/wc2.1_5m_bio",pattern=".tif")
#unfortunately the names are not in numerical order, we can fix this when we list the files
list.files("./Downloads/wc2.1_5m_bio",pattern=".tif")[c(1,12:19,2:11)]

#fetching historical data
rst.nms <- list.files("./Downloads/wc2.1_5m_bio",pattern=".tif")[c(1,12:19,2:11)]
rst.fld <- list.files("./Downloads/wc2.1_5m_bio",pattern=".tif",full.names = T)[c(1,12:19,2:11)]

#loading all rasters into a single multi-band raster:
rst.stk <- stack(rst.fld)
```

  - Line 13: sets up the R work environment in C:/Practical
  - 16 to 18 create two lists of files, one is the file names and the other is the path to the files.
    - Beware that the bio files are loaded in the wrong order so those numbers: c(1,12:19,2:11) correct for that – the example below exemplifies.

```
> #lists all historical Bioclimatic variables into two objects: one for the names and one for the path to the files
> list.files("./Downloads/wc2.1_5m_bio",pattern=".tif")
 [1] "wc2.1_5m_bio_1.tif"  "wc2.1_5m_bio_10.tif" "wc2.1_5m_bio_11.tif" "wc2.1_5m_bio_12.tif"
 [5] "wc2.1_5m_bio_13.tif" "wc2.1_5m_bio_14.tif" "wc2.1_5m_bio_15.tif" "wc2.1_5m_bio_16.tif"
 [9] "wc2.1_5m_bio_17.tif" "wc2.1_5m_bio_18.tif" "wc2.1_5m_bio_19.tif" "wc2.1_5m_bio_2.tif"
[13] "wc2.1_5m_bio_3.tif"  "wc2.1_5m_bio_4.tif"  "wc2.1_5m_bio_5.tif"  "wc2.1_5m_bio_6.tif"
[17] "wc2.1_5m_bio_7.tif"  "wc2.1_5m_bio_8.tif"  "wc2.1_5m_bio_9.tif"
> #unfortunately the names are not in numerical order, we can fix this when we list the files
> list.files("./Downloads/wc2.1_5m_bio",pattern=".tif")[c(1,12:19,2:11)]
 [1] "wc2.1_5m_bio_1.tif"  "wc2.1_5m_bio_2.tif"  "wc2.1_5m_bio_3.tif"  "wc2.1_5m_bio_4.tif"
 [5] "wc2.1_5m_bio_5.tif"  "wc2.1_5m_bio_6.tif"  "wc2.1_5m_bio_7.tif"  "wc2.1_5m_bio_8.tif"
 [9] "wc2.1_5m_bio_9.tif"  "wc2.1_5m_bio_10.tif" "wc2.1_5m_bio_11.tif" "wc2.1_5m_bio_12.tif"
[13] "wc2.1_5m_bio_13.tif" "wc2.1_5m_bio_14.tif" "wc2.1_5m_bio_15.tif" "wc2.1_5m_bio_16.tif"
[17] "wc2.1_5m_bio_17.tif" "wc2.1_5m_bio_18.tif" "wc2.1_5m_bio_19.tif"
```

- The first step is to rename all the variables so that they have more meaningful names. **It is vital to ensure that ALL variables in ALL scenarios have precisely the SAME names otherwise MAXENT will not be able to recognize them**
  - The next code snippet renames the historical data raster's to list of names shown, which are the same as bioclimatic codes.

```
#Renaming bioclimatic layers:
names(rst.stk)
names(rst.stk) <- c("Bio01","Bio02","Bio03","Bio04",
                    "Bio05","Bio06","Bio07","Bio08",
                    "Bio09","Bio10","Bio11","Bio12",
                    "Bio13","Bio14","Bio15","Bio16",
                    "Bio17","Bio18","Bio19")

```

- The same must be done for the future scenario. If you have multiple scenarios, you need to adapt the code to those scenarios independently.
    - First, we load the future scenario multilayer raster

```
#fetching future scenario
rst.ssp370 <- stack("./Downloads/share/spatial03/worldclim/cmip6/7_fut/5m/IPSL-CM6A-LR/ssp370/wc2.1_5m_bioc_IPSL-CM6A-LR_ssp370_2061-2080.tif")
```

    - Then we check if the layers are in the correct order and, if so, we just rename them

```
> names(rst.ssp370)
 [1] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.1"  "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.2"
 [3] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.3"  "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.4"
 [5] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.5"  "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.6"
 [7] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.7"  "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.8"
 [9] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.9"  "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.10"
[11] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.11" "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.12"
[13] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.13" "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.14"
[15] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.15" "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.16"
[17] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.17" "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.18"
[19] "wc2.1_5m_bioc_IPSL.CM6A.LR_ssp370_2061.2080.19"
> names(rst.ssp370) <- names(rst.stk)
> names(rst.ssp370)
 [1] "Bio01" "Bio02" "Bio03" "Bio04" "Bio05" "Bio06" "Bio07" "Bio08" "Bio09" "Bio10" "Bio11" "Bio12" "Bio13" "Bio14"
[15] "Bio15" "Bio16" "Bio17" "Bio18" "Bio19"
```

Cropping/clipping Environmental data in R

This section will use the species occurrence data to **delineate a box around it and use that box to crop (aka clip) the historical and future scenario data**. The script used here is the same as in the previous section.

Step by Step:

- First, we load the .csv (Remember if you need to change the path to the files)
    - Read.csv <- opens .csv files in NA style by default
    - Read.csv2 <- opens .csv files in EU style by default
        - Both can be changed and adapted to custom decimals and delimiterss
- Then we create the R object equivalent to a point shapefile: SpatialPointsDataFrame(SPDF)
- And finally, define the Coordinate system as being WGS84 using a PROJ4 definition
    - PROJ4 is a cross-platform library specifically created for coordinate projection operations: https://proj.org/

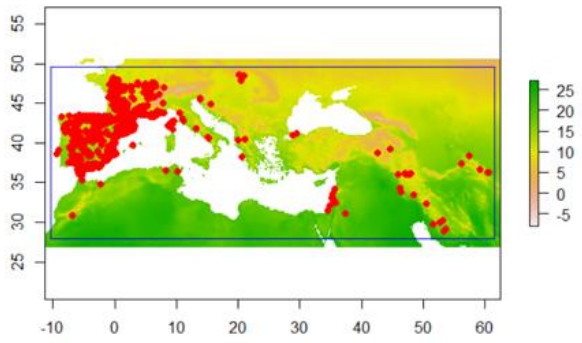
```
#To crop the area for the model training, we will use the occurrence data.
#In truth, the AOI used for the model training can have deep implications on the modelling fitness
#e.g. https://www.sciencedirect.com/science/article/pii/S1574954120301291?dgcid=rss_sd_all discusses this

# Load species occurrence file
# notice im using read.csv2, which expects a EU type of table. If you want to use the NA style then its read.csv
# if your tables are stored in some other format, then use read.table and check the package details for custom delimiters and decimals
sp <- read.csv2("./Occurrences/Tables/Rhinolophys_csv2_clean.csv",header=T) #load csv of occurrence -> already 3 column
head(sp) #check table looks correct
sp_shp <- sp #rename table
coordinates(sp_shp) <- ~longitude+latitude #convert table to points shapefile
proj4string(sp_shp) <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
```

    - There are thousands of coordinate systems, but you can look up for their details and how to use them in various programming languages in webpages such as the spatialreference.org

> The operation executed on the previous code snippet is **NOT A REPROJECTION**. It simply defines the projection of the object sp_shp in R. If you define the wrong projection, the object will not be plotted in the correct location.

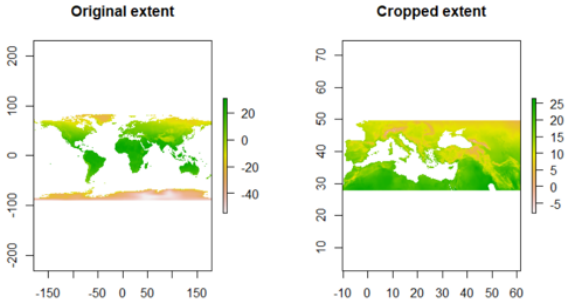
- We can now explore how our data looks like using R and check if everything is in place:

```
#Create bounding box around points
bbox <- extent(sp_shp) #create bounding box of points
bbox <- bbox+2 #increase border so we do not truncate data
plot(rst.stk$Bio01,ext=bbox+2)
plot(bbox, col='blue',add=T) #check if box surrounds points
plot(sp_shp,add=T,pch=19,col='red') #add points
```



- The above code snippet only plots the data, it did not alter or change anything about it. To crop/clip the image we need to use the next code snippet:

```r
#cropping the present data
stk.present.AOI.crop <- crop(rst.stk,bbox) #clip to training area
#plotting the example
par(mfrow=c(1,2)) #sets the plotting area to a 1 line 2 columns set up
plot(rst.stk$Bio01,main="Original extent")
plot(stk.present.AOI.crop $Bio01,main="Cropped extent")
par(mfrow=c(1,1)) #sets it back to 1 image per plot
```



- Adapting the above function to the future scenarios is then obvious:

```r
#cropping the future data
stk.ssp370.AOI.crop <- crop(rst.ssp370,bbox)
```

The last step is to save everything to the folders we have previously created. If you have multiple scenarios, you must adapt the next code snippet.

MAXENT uses the LAST subfolder to name its models outputs. Make sure to give a different subfolder name for each different scenario. If you do not do this, MAXENT will overwrite the output files.

```r
#first the uncropped data:
#now we can save them to another folder in a format
#that maxent can read
#saving the cropped historical data data in .asc format
writeRaster(rst.stk,                          #multilayer raster
            "./EnvData/WLD/PresentWLD/.asc", #output folder plus extension .asc
            overwrite=T,                      #overwrites any files with the same name in the folder
            bylayer=T,                        #saves each variable with the layer name that we set before
            suffix="names")                   #uses the band names instead of the band number

#Same for the future data
writeRaster(rst.ssp370,                       #multilayer raster
            "./EnvData/WLD/FutureWLD/.asc",      #output folder plus extension .asc
            overwrite=T,                      #overwrites any files with the same name in the folder
            bylayer=T,                        #saves each variable with the layer name that we set before
            suffix="names")                   #uses the band names instead of the band number

#now we do the same, for the AOI
writeRaster(stk.present.AOI.crop ,            #multilayer raster
            "./EnvData/AOI/PresentAOI/.asc",    #output folder plus extension .asc
            overwrite=T,                      #overwrites any files with the same name in the folder
            bylayer=T,                        #saves each variable with the layer name that we set before
            suffix="names")                   #uses the band names instead of the band number

#Same for the future data
writeRaster(stk.ssp370.AOI.crop,              #multilayer raster
            "./EnvData/AOI/FutureAOI/.asc",     #output folder plus extension .asc
            overwrite=T,                      #overwrites any files with the same name in the folder
            bylayer=T,                        #saves each variable with the layer name that we set before
            suffix="names")                   #uses the band names instead of the band number
```

- The writeRaster function will take some time, especially if you have multiple scenarios and you are using global datasets. Wait for it to finish before proceeding. Perhaps you can read about Spearman's rank correlation and Variance of inflation factor meanwhile.

Before running the script above, confirm all folders exist and are in the proper place. And confirm the paths you give in the writeRaster command is correct.

You can also do these steps in a GIS but, it is more work than a simple R script and more error prone. Also notice that a simple script makes it easier to also detect where errors might have occurred.

Selecting environmental variables

- **Spearman's rank correlation coefficient: r**
  - Measures the statistical association or dependence between two variables. It is a nonparametric since we have no reason to expect any normality on the environmental data.

$$r_s = \rho_{\mathrm{rg}_X, \mathrm{rg}_Y} = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X} \sigma_{\mathrm{rg}_Y}},$$

  - The test will be executed by successively comparing each environmental variable against one of all the other variables – we will call it Pairwise for simplicity sake although it is not a true pairwise test.
    - If r > 0.5, then you have high degree of correlation.
      - **As a rule of thumb, we consider that a variable is problematic if r is bigger than 0.7**

  - Another interpretation of this correlation coefficient is to consider how much association exists between variable X and Y. If this association is too high, then, in the context of SDM, it is also hard to explain which of the variables is contributing for the distribution of the species since both have the same explanatory relationship. Besides, potentially negatively affecting MAXENT predictions.

- **Variance Inflation Factor (VIF)**
  - Measures the statistical dependence of a variable X to a combination of N other variables – so – how the model correlation affects X variable in the given model with N variables. It also quantifies the severity of multicollinearity for a ordinary least squares regression

$$VIF_i = \frac{1}{1 - R_i^2}$$

  - If you have variables X1, X2, X3,…. Xn, that you will use in the model, then the VIF is calculated for each variable:
    - VIF (X1): X1 ~ X2 +X3 + … + Xn
    - VIF (X2): X2 ~ X1 +X3 + … + Xn
    - …..
  - The R2 in each case is the R^2 of the multilinear regression using all variables except for the target variable that is being tested
  - **The rule of thumb in the VIF value is: VIF > 10 implies a model structure that is highly correlated and that it likely will affect the model performance.**

- If a high number of variables selected have high pairwise correlation, then, it is likely that the VIF will be high as well!

- **Variables and area of interest:**
  - Historical data is provided as multiple rasters where each is one of the bioclimatic variables
  - The Spearman's rank correlation (r) and VIF tests should both be done using the area of interest and not the entire global data. (unless your species has a global distribution).
- **Statistical criteria:**
  - Test all the pairwise correlations between the variables of interest
    - If there is no r-pairwise correlation > 0.7
      - Do not remove any variable & confirm VIF < 10
    - If one or more r-pairwise correlations > 0.7
      - If VIF > 10
        - Remove one of the correlated variables
          - Consider the ecological significance when choosing which variable to remove -> **remember that statistically, highly correlated variables explain the same thing, but they can have different ecological meanings.**
      - If VIF < 10
        - **You can use this set of variables and proceed with the analysis**
          - PS: if VIF is very close to 10, you can still consider removing variables, it is up to you.

Very important:

Autocorrelation testing (r and VIF) are important only for the areas where the model is trained (meaning, the cropped area of interest data). This is because global correlations between climate variables might NOT exist at a local scale and, otherwise, local correlations between climate variables might exist NOT exist at a global scale. Your model will only use the data made available in the AOI during training, so what happens in the AOI will affect the model.

Furthermore, autocorrelation effects on the predicted dataset (e.g. scenarios) do NOT affect the model training either. This means, that **correlation analysis does NOT have to be performed on the target dataset**.

And lastly, these statistical tests test two different aspects of variable and model correlation. A simple understanding of correlation is that if Variable X is highly associated with Variable Y, then you can predict the values of Y using variable X. This means that when you use them together **1) it is hard to interpret which variable is responsible for a specific output and 2) models tend to overfit the more easily.**

In your case:

- Which variables are ecologically significant for the species?
  - Base your decision on what you already know about the species
  - Make a list of potential variables that have more significance
    - Then use the statistical tests to verify that you can use them on your model
  - **Explain the reasoning for selecting the variables and describe which of these variables were excluded (or not) on the report**
- **Use this set of N hypothetical bioclimatic environmental variables which are related to the species physiology for the next step by step test of spearman's r correlation and VIF to confirm (or not) that you can use this pre-selection.**
  - The result of these step is also a more "parsimonious" model which uses the most data possible without negatively affecting the model.

Testing Spearman's rank correlation:

- In my case, the first selection was:
  - Bio 01 – Mean annual temperature
  - Bio 04 – Temperature seasonality
  - Bio 07 – Temperature Annual range
  - Bio 12 – Annual precipitation
  - Bio 15 – Precipitation seasonality
  - Bio 19 – Precipitation of the coldest quarter

Then, I performed a **Pairwise spearman's R correlation** test: (remember: **02_VariableSelection.R)**

```
9  ###first load all the env data of your AOI in the present
10 setwd("c:/Practical")
11 path2presentData <- list.files("./EnvData/AOI/PresentAOI/",pattern=".asc",full.names = T)
12 stk.present.AOI.crop <- stack(path2presentData)
13 names(stk.present.AOI.crop) <- c("Bio01","Bio02","Bio03","Bio04",
14                                  "Bio05","Bio06","Bio07","Bio08",
15                                  "Bio09","Bio10","Bio11","Bio12",
16                                  "Bio13","Bio14","Bio15","Bio16",
17                                  "Bio17","Bio18","Bio19")
18
19 ### the autocorrelation testing is important ONLY for the areas where the
20 ### model is trained, so, for this section, we use only the cropped enviromental data
21
22 ### pairwise testing
23 #first we convert the cropped raster to a data.frame
24 stk.present.AOI.crop <- na.omit(as.data.frame(stk.present.AOI.crop)) #we also remove NA's
25 #now this stores the pearson correlation in a matrix
26
27 #here we can already select only the variables that we are interested on - or - you can just calculate
28 #using all the variables. It is the same, the pairwise correlation does not change since each one is a different comparision
29 stk.present.AOI.crop.sel <- stk.present.AOI.crop[,c("Bio01","Bio04","Bio07","Bio12","Bio15","Bio19")]
30
31 #cor function calculates the pairwise r correlation
32 cor.tab.allvariables <-cor(stk.present.AOI.crop)
33 cor.tab.selvariables <-cor(stk.present.AOI.crop.sel)
34
35 cor(stk.present.AOI.crop.sel,method="spearman")
36 cor(stk.present.AOI.crop.sel,method="spearman")
37
38 #remember to change to write.csv if needed
39 write.csv2(cor.tab.allvariables,"CorrelationTable_AOI_AllVariables.csv")
40 write.csv2(cor.tab.selvariables,"CorrelationTable_AOI_SelVariables.csv")
```

- This short script starts by loading the rasters from the AOI folder (lines 11 and 12)
- Renames them (lines 13)
- Converts them to a R dataframe object (line 24).

        - It also removes all "Not Available" (na) values. These are places where there is no values in the raster, e.g. the water areas.
    - Uses the cor function of R (line 32, 33) to calculate the pairwise correlation (in-built function).
    - Saves the results, to two different tables: one considering only the selected variables (above) and one using all the 19 bioclimatic variables. Either is fine since the pairwise correlation **does NOT change when you remove variables**. But its easier to explore a smaller table.
    - The pairwise correlations can then be more easily investigated using Excel

- Open the file: "CorrelationTable_AOI_SeVariables.csv" in Excel

	A	B	C	D	E	F	G
1		Bio01	Bio04	Bio07	Bio12	Bio15	Bio19
2	Bio01	1	-0,18395	0,08421	-0,5858	0,656659	-0,30598
3	Bio04	-0,18395	1	0,91608	-0,40059	-0,12086	-0,4368
4	Bio07	0,08421	0,91608	1	-0,57283	0,104595	-0,5289
5	Bio12	-0,5858	-0,40059	-0,57283	1	-0,40992	0,851161
6	Bio15	0,656659	-0,12086	0,104595	-0,40992	1	-0,06551
7	Bio19	-0,30598	-0,4368	-0,5289	0,851161	-0,06551	1
8							

    - Highlight the content of the table - > Styles section -> Conditional Formatting - > highlight cells rules -> Larger than 0.7
        - Also highlight negative correlations: less than -0.7 in different colour if you like.

	A	B	C	D	E	F	G
1		Bio01	Bio04	Bio07	Bio12	Bio15	Bio19
2	Bio01	1	-0,18395	0,08421	-0,5858	0,656659	-0,30598
3	Bio04	-0,18395	1	0,91608	-0,40059	-0,12086	-0,4368
4	Bio07	0,08421	0,91608	1	-0,57283	0,104595	-0,5289
5	Bio12	-0,5858	-0,40059	-0,57283	1	-0,40992	0,851161
6	Bio15	0,656659	-0,12086	0,104595	-0,40992	1	-0,06551
7	Bio19	-0,30598	-0,4368	-0,5289	0,851161	-0,06551	1
8							

    - In my case, there are two pairwise correlations > 0.7
        - Bio07~Bio04
        - Bio19~Bio12
    - We can use test the VIF now to see if my model is ok even considering these two correlations

- **Testing multicollinearity using VIF:**

- Its trivial to do by hand, but, extremely repetitive and prone to error. Luckily we can use the vif function in the USDM package

```
#### VIF Testing #####

#multicollinearity testing
library(usdm)

#e.g. i select Bio01; Bio04; Bio07; Bio 12; Bio 15 and bio 19
stk.present.AOI.crop.sel <- stk.present.AOI.crop[,c("Bio01","Bio04","Bio07","Bio12","Bio15","Bio19")]
#and the VIF test
vif(stk.present.AOI.crop.sel, maxobservations=nrow(stk.present.AOI.crop.sel))
```

- Inputs of vif function:
  - Target dataframe
  - Number of observations to be used (if less than the total number, them it will use a sample of the data frame)

```
> vif(stk.present.AOI.crop.sel, maxobservations=nrow(stk.present.AOI.crop.sel))
  Variables      VIF
1     Bio01  3.302447
2     Bio04 11.499434
3     Bio07 11.401577
4     Bio12  9.375736
5     Bio15  2.338738
6     Bio19  5.847520
>
```

- This implies that either Bio07 or Bio 04 must be removed.
  - I opted to remove Bio07 – The mean temperature range since I already have enough variables related with temperature.

The pairwise correlation does not have to be tested again since it does not change when variables are removed. **But the VIF must be recalculated every time a variable is removed since we removed variables from the model**. In your report we expect that you state all intermediate VIF results and not just the final model, so we can understand your reasoning when removing variables.

```
df.stk.AOI <- stk.present.AOI.crop[,c("Bio01","Bio04","Bio12","Bio15","Bio19")] #minus the temperature range
vif(df.stk.AOI, maxobservations=nrow(df.stk.AOI))
```

```
> vif(df.stk.AOI, maxobservations=nrow(df.stk.AOI))
  Variables      VIF
1     Bio01 3.020686
2     Bio04 1.766862
3     Bio12 9.354233
4     Bio15 2.240141
5     Bio19 5.834206
>
```

We finally have everything set up (almost) to maxent!

Checklist before MAXENT:

Regarding the occurrence data:

- Is the Occurrence data in a .csv or .txt file format in NA style?
- Are the fields on the table in the correct order: species, longitude, latitude?
- Do I have only one species name in the column species?
- Do I have only numbers in the column's longitude and latitude?
- Do I have missing data on the table?

Regarding the Environmental data:

- Do I have a list of the selected variables?
- Are the variables saved in .asc format?
- Are they saved to different folders where each last subfolder has a unique name?
    - Does the data inside of each subfolder have the same names? (They should!)

Have you downloaded MAXENT?

Latest version: https://biodiversityinformatics.amnh.org/open_source/maxent/

Version 3.4.1: downloadlink

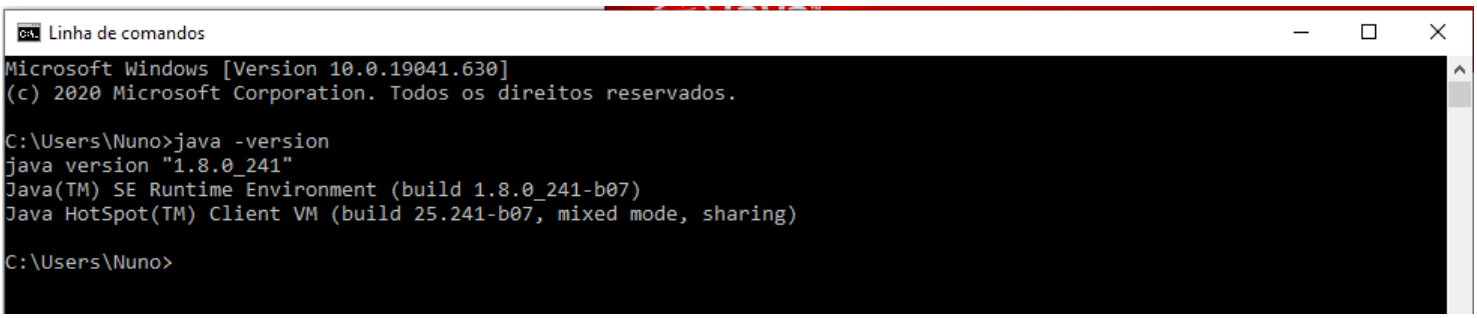
- I had problems running the latest version of MAXENT. The problem I had, is detailed later. Download this earlier version just in case you have the same problem later.

MAXENT requires JAVA to be installed in the computer to run. Do I have it?

https://www.java.com/en/download/help/download_options.xml

Confirming if JAVA is installed (Windows): (you can find how to do this in other operating systems on the internet).

- Open a command line window (use the search option on the bottom left)
- Type: "java -version" and you should see the following response:



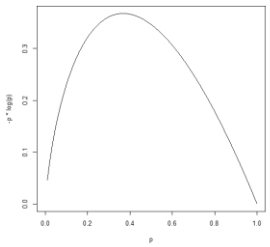
MAXENT – Maximum Entropy Modelling:

MAXENT Intro:

This small intro aims to give a quick intro to MAXENT, its main ideas and some of its recent history. In terms of SDM what we are wanting to know is the **probability (π)** of **Presence (P)** of a species given the observation of some **Environmental factors (E)**, aka Bayes theorem:

$$\boldsymbol{\pi(P|E) = \frac{\pi(E|P) \cdot \pi(P)}{\pi(E)}}$$

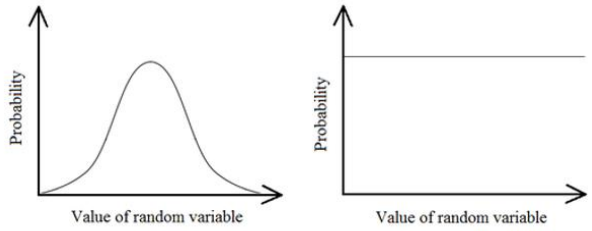
We can achieve this by different method, but the one MAXENT uses is based on the maximum entropy principle. This principle is based on the premise that when estimating any probability distribution, you should select the one that is most uncertain - initially proposed by E. T. Jaynes in 1957 for thermodynamics. **Entropy is a measure for disorde**r: When you flip a coin for which the π(head) = π(tail)=0.5, then the outcome of each toss is hard to predict and therefore has a high entropy. But if it is a biased coin where the probability for head is almost 1, then the unpredictability of the outcome shifts towards a clear expectation for head, and so the entropy decreases. If we plot a coin toss case, the highest uncertainty relative to the results would occur when the coin is unbiased:



Multiple metrics exist for measuring it, but MAXENT uses the commonly used Shannon Information Entropy which measures the contribution of each possible outcome of a variable to the overall entropy in relation to the probability of these outcomes.

$$H(\hat{\pi}) = -\sum_{x \in X} \hat{\pi}(x) \ln \hat{\pi}(x)$$

Cool, now in terms of Machine learning from all the possible distribution that fit our constrains, **we need to find the one that maximizes the entropy.** This is achieved by minimizing the relative entropy (RE) between two distributions: one that is defined in function of our observations and one uniform distribution (everything has the same probability);



In Phillips, (2004, 2006) this is described as minimizing the **relative entropy (RE)** between an observed **probability π-tilde** and the a **Gibbs distribution (q_λ)** that fits the observations (plus some weighter regularization parameters to avoid overfitting):

$$\mathrm{RE}(\tilde{\pi} \parallel q_{\boldsymbol{\lambda}}) + \sum_j \beta_j |\lambda_j|$$

$$q_\lambda(x) = \frac{\exp\left(\sum_{j=1}^{n} \lambda_j f_j(x)\right)}{Z_\lambda}$$

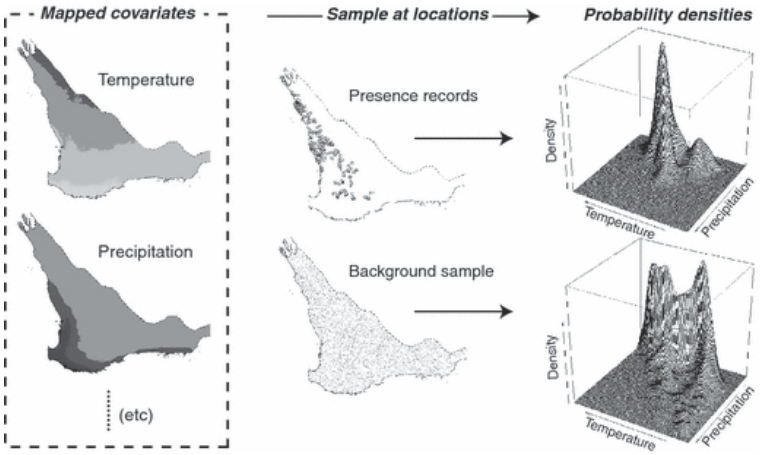
Feature weights*f(x)

Normalization constant

And these are the parameters (feature weights) that the machine learning procedure will try to change each parameter individually until it finds the best solution (using a sequential-algorithm for searching the solution space).

But what does this all mean?!?!

What this means for the species distribution models is that probability of occurrence is given by the conditional probability of a given set of presence observations against the distribution of the "random background" or, basically, if the distribution of the species was random: (figure adapted from Elith, 2010). **Notice that the *Environmental probability density* is obtained by the "Background points" and the final presence density function is the one that "maximizes" their similarity (MAXENT) by minimizing their difference (aka relative entropy) through tweaking the parameters that define the Gibbs distribution.**



But it also means that even if MAXENT aims to find the most uncertain model that fits the constraints (occurrence data), still, any model that you train with MAXENT will be constrained by the real observations of the species. Therefore, it is important to consider if the occurrence data indeed reflects the ecological niche of the species and how reasonable is it to extrapolate (in the environmental space) to ranges where you have no information available at all. This is one of the core problems in all cases of machine learning: "unseen" examples.

The big growth of MAXENT being used in Species Distribution Models started when Phillips, 2006 made his seminal publication on the topic and when the Java utility that is used in this tutorial was made available. Only recently, this software has been made open source and recently there has been some debate about the nature of the algorithm. Renner, 2013 showed that MAXENT as applied in SDM was a particular case of Poisson Regression Models which lead to a number of changes on how the model is used and provided a number of options to address some of the more challenges in MAXENT SDM modelling (see Table 1 in Renner).

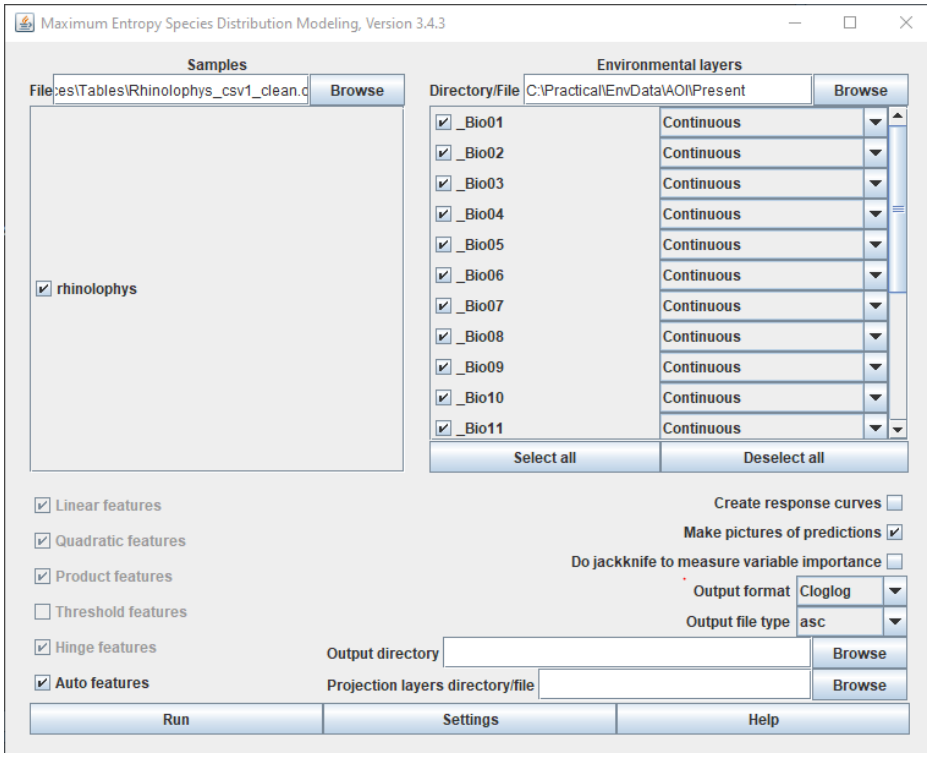
So now, not only the Java code was made fully open source but also MAXENT was coded in R on maxnet package albeit with some limitations in terms of the visualizations and outputs produced. Most importantly, the possibility of using Poisson regressions to implement the model, implies that everything can be done using standard R packages (which we will not do today!). Opportunities and its implications on this topic are further discussed in Phillips, 2017, by the original authors of MAXENT.

Deeper explanations on MAXENT and how it works and how it is adapted for SDM are beyond the scope of this class. But feel free to investigate more about it:

- Simon DeDeo (Complexity Explorer channel) (not applied to ecology)
- John Harte (Stanford) - Talk on Maximum Entropy in Ecology complexity
- Phillips - 2004; 2006; 2017 -> original MAXENT papers in SDM
- A brief tutorial on MAXENT
- Various: (1) ; (2); (3) -> **Highly recommended for ecologists and potentially very useful for your reports.**

Setting up MAXENT:

- Open the MAXENT software
    - Use the executable jar file (.bat)
- Load your occurrence data
- Load the training environmental layers (the ones cropped to the AOI).
- **You should now see this:**

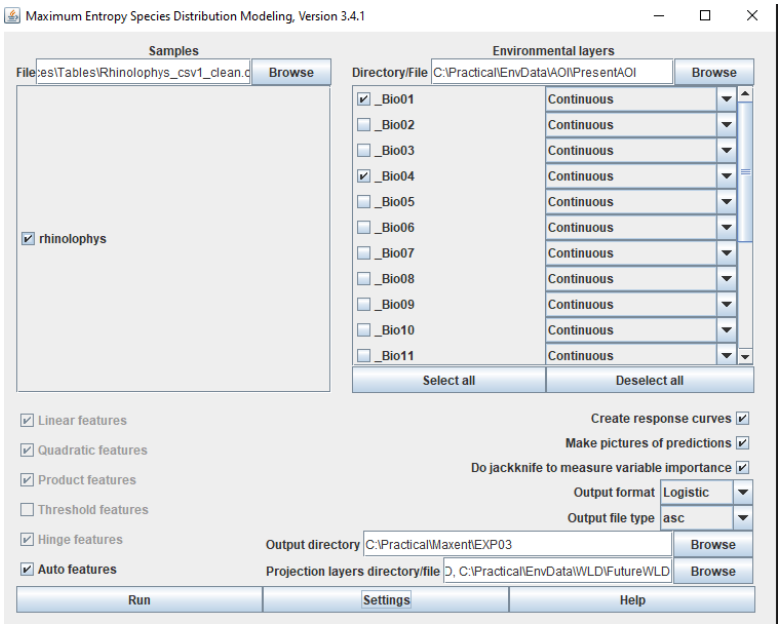


- **On the menu on the right**, select only the variables that were identified before:
    - in the case of this tutorial: Bio01, Bio04, Bio12, Bio15, Bio19
- **Options on the main menu**
    - Auto-features: On
    - Create response curves: On
    - Make pictures of predictions: On
    - Do jacknife to measure variable importance: On
    - Output format: Logistic
    - Output file type: asc
    - **Output directory:**
        - C:\Practical\Maxent\EXP01 **(ADAPT TO YOUR CASE)**
        - Maxent produces a lot of outputs so it advised to have different folders for different runs as you try out the software.
    - **Projection layers directory/file**
        - This is where you add the paths to your scenarios or world data

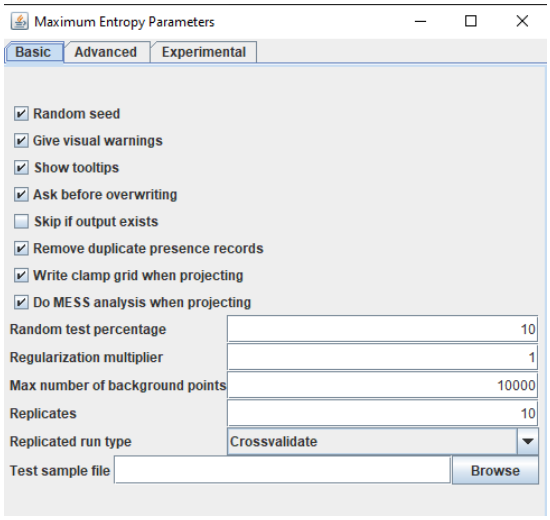
- Maxent allows you to add all the paths at once, separated by a comma (which is handy because otherwise you would have to do one by one)
- **In my case, given where my files are:**

  C:\Practical\EnvData\WLD\PresentWLD,
  C:\Practical\EnvData\AOI\FutureAOI,
  C:\Practical\EnvData\WLD\FutureWLD

- **The main menu of MAXENT should now look like:**



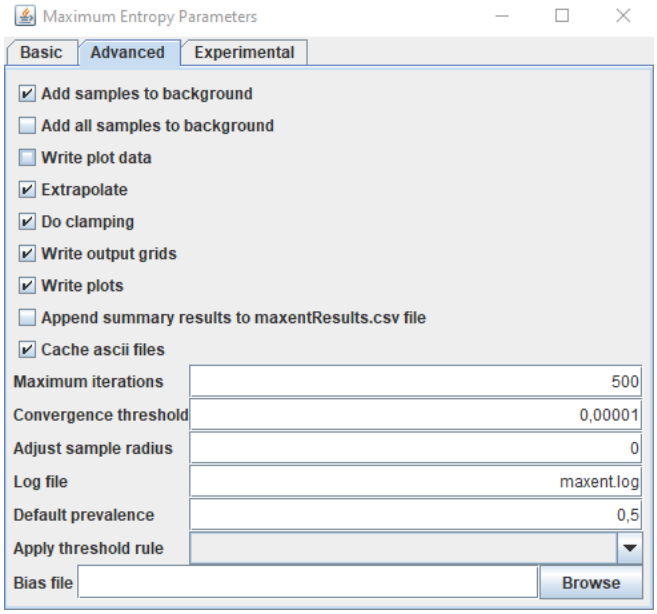
- Select "Settings" and then Basic.
  - Set the Basic menu the same as this:



- **Basic Menu:**
  - Notice the number of replicates:
    - The number of replicates is the number of times maxent repeats the entire process. The more "the better". **Your final result on the report should consider at least 10 replicates.**

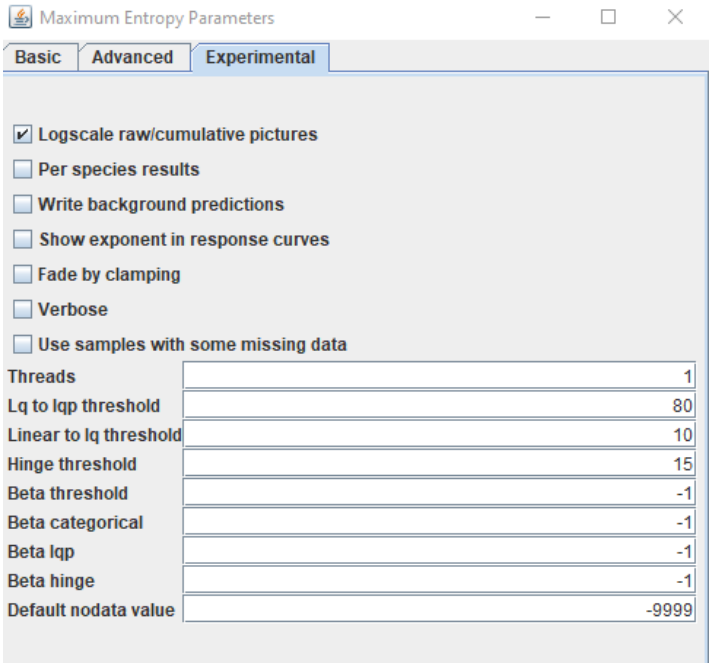
- It's recommended though, that first time you run the model, you are more concerned to check if everything is running properly, so , feel free to run a smaller number of replicates, e.g. 3 or 4.

- Set the Advanced section like this:



- For you to investigate:
  - What is the impact of the "extrapolate" and "do clamping"?
    - Hint 1 & hint 2
    - What is the best option for your case?
    - You can discuss this on your report or with us during the exercise.

- **The "Experimental" section it should look like:**



Most of these options are general options (per species results, threads, etc) but some are related to the inner workings of the model. **The Threads option refers to the number of cores you want to use for the calculation -> the more the faster,** but it might slow down your computer. Choose whichever number of cores you are comfortable with.

Notice the "Fade by clamping" option. This relates with the clamping options on the advanced section. Investigating what it does might be interesting and something you can consider looking at for your report.

When all these options are done, go back to the main section and press "Run". And wait.

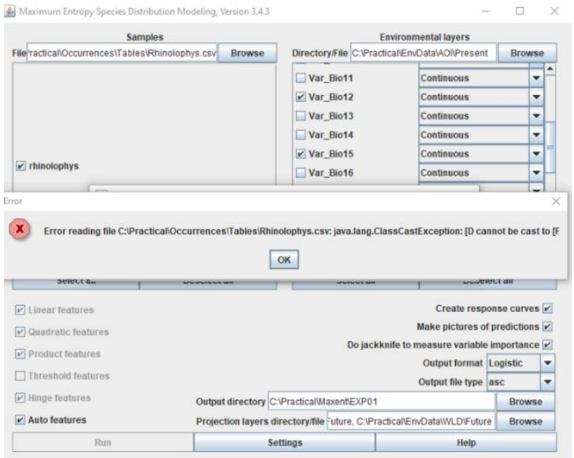
MAXENT – Common warnings & errors messages:

Common warnings: Missing data or repeated/overlapping occurrences

This message is telling that some of the environmental data is missing. This can be for many reasons, perhaps the points fall outside the map area or perhaps the point falls into a pixel/cell that is empty (eg. Water). Ignore these messages and if the process continues running it means you are ok to go.

Another notification you might get is that some occurrence is overlapped with another. This happens because maxent only considers one occurrence per pixel/cell and you can have multiple occurrences of a species within a 10 by 10 km cell. Ignore this warning, and proceed, it should be fine.

Do notice that if at any point you georeferenced the data incorrectly, both these problems will become serious. **In my case, I also had a serious problem:**



This seems to be a problem with the latest version of maxent and my Java installation. It is common for these things to happen with older software or when new version of Java is launched. My solution was to use an older version that I had which worked fine. Find his older version here.

Validating Species Distribution Models:

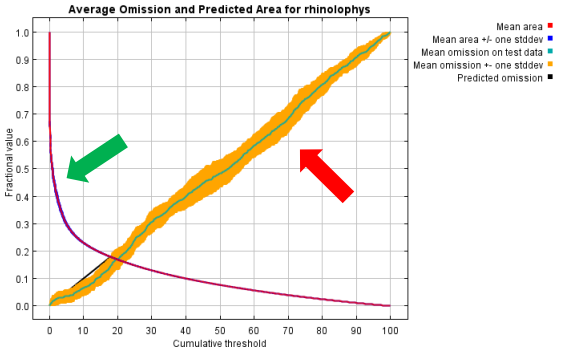
Interpreting MAXENT report & outputs:

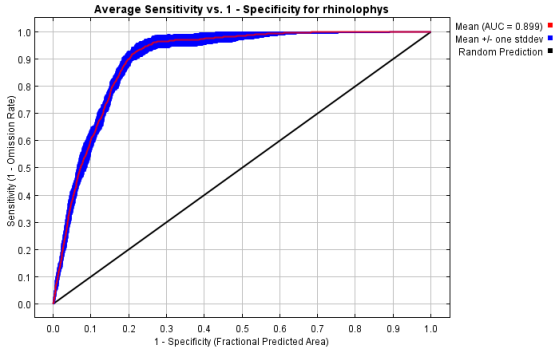
- Open your maxent/Results folder and investigate

You will find an .html file/s that summarize your results and includes the AUC evaluation. If you have repetitions, there is one html file per repetition, and one that summarizes the results of each replicate, reporting means and standard deviations.

The folder also contains ascii files which you can open in ArcGIS, png files, and other separate files. It is important to note though, that these ascii files do not carry projection information with them and therefore, ArcGIS or QGIS will warn you that there is no projection system. You know though, that the data is in WGS84

On the report file, notice your first and second graphic: 1) Average omission and predicted area for the modelled species and 2) AUC value (second graphic – AUC-ROC).





The graphic on the left shows how the predicted area of presence changes in function of the threshold (green arrow). When the th >= 0, then the entire map predicted presence, therefore, fractional area = 1. This helps to understand if your model is on the risk of overestimating presences. The second arrow (red) points at the mean omission error overlapping the predicted omission –> meaning, the omission predicted on the cross validation of the training data and omission predicted on the test data.

On the right graphic, it is the AUC-ROC curve. Notice, it is based on the variations in areas and not real presence/absence validation data. According to Mandrekar, 2010:

- AUC ~ 0.5 -> No "better" than random
- 0.7 > AUC ≤ 0.8 -> Acceptable
- 0.8 > AUC ≤ 0.9 -> Excellent
- AUC > 0.9 -> Outstanding

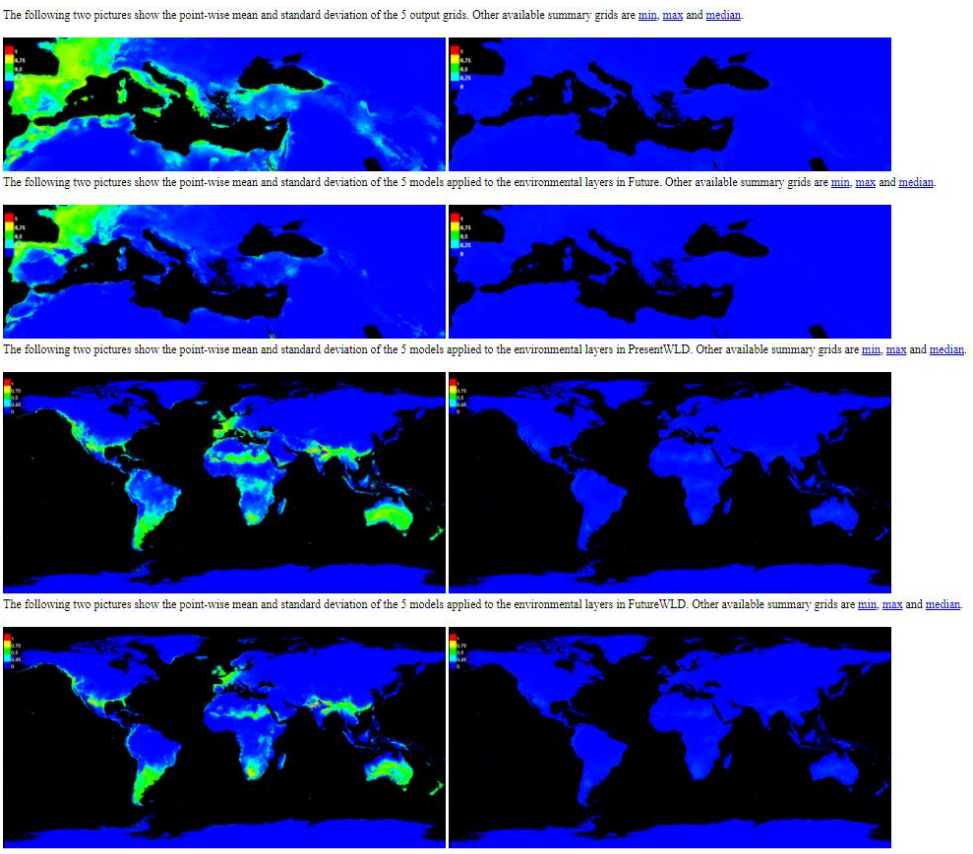
Remember though, that, prevalence would affect this standard interval and in our case, we do not have a precise distribution of the species. If we had, we would not need to model it. So, you have to interpret this in the context of your species distribution.

But this result must often be interpreted in function of how "generalist" or "specialist" your species is. Generalist species occupy broader areas so they are more prevalent and as explained in the pink box before and Phillips et al. 2006, the maximum AUC is 1-a/2 where a is the total fraction of the area occupied by the species.

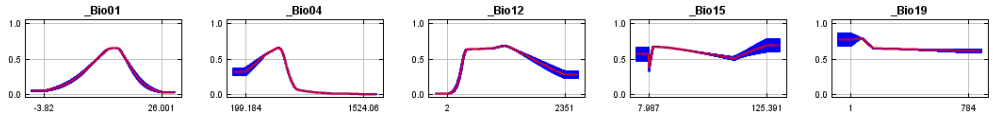
What this implies is that for generalist species you do not necessarily expect high AUC for accurate models whilst for specialist species you should expect very high AUC for accurate models. **Please consider this when discussing if your AUC was "good" or "bad".**

In the "Pictures of the model" section - you see the probability of occurrence within each pixel and the output of the different scenarios in the order you submitted them.

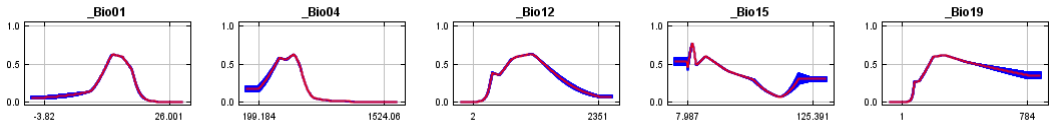
- Maxent reports the mean probability (left) and its standard deviation (right) of all the replicates you ran
- The layers are in your main results folder also:
    - E.g. Species_avg.asc ; species_scenarion_avg.asc
    - And they can also be explored in a GIS or opened in R. These are normal .asc format rasters that you can use for further analysis
    - Furthermore, you also have .asc files of each individual run.



- **In the Response curves section:**
  - You have the "response" curves of the species probability in function of the values in each variable.
    - Keeping all variables at their mean value except for the target value:



    - While predicting using only target value

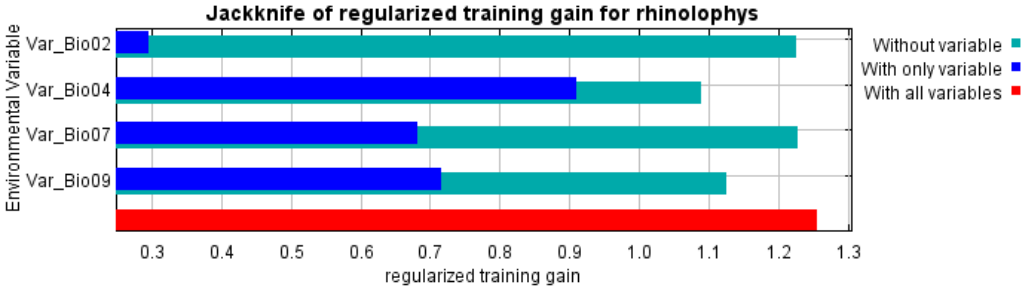


  - This is very helpful to identify many things:
    - **Identify ranges of the values that correlate highly with the presence** of the species (e.g. Bio01, approximately around 15 mean annual temperature) (eyeballing)
    - **Correlations between variables** (if they have similar responses)
      - Bio01, Bio04 seem to have a small degree of correlation
    - **Extrapolations**
      - If one variable has high probability near the maximum value, that means that if that maximum value changes in a future scenario, then the model is extrapolating beyond that range.
        - E.g. bio 19 for both cases
    - **Non-linear relationships**
      - In some case, you have a "normal" response, as in it follows a "kind of" gaussian distribution while in other it behaves very weird
- **In the Analysis of Variable contributions section:**
  - Variable importance:

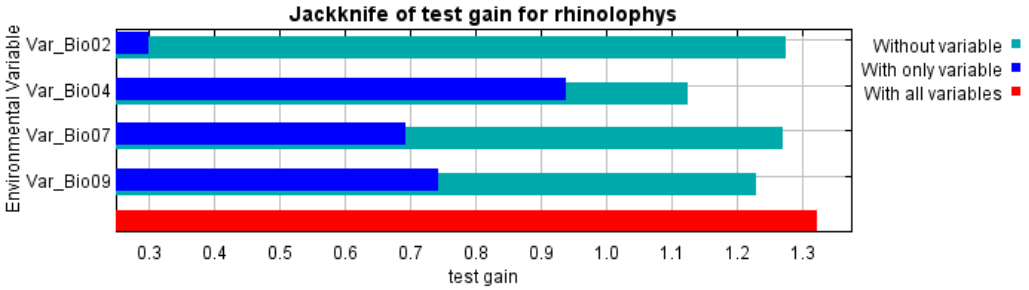
Variable	Percent contribution	Permutation importance
_Bio04	52.9	50.8
_Bio12	36.3	31.9
_Bio01	6.6	12.9
_Bio15	3.2	2.6
_Bio19	1.1	1.9

    - Percent contribution:
      - Changes in the regularization parameters, normalized for [0 – 100]%
    - Permutation importance:
      - Changes in training AUC by excluding/including given variable, normalized [0 – 100]%
    - Most important variable: Bio04 – Temperature seasonality
    - Correlation between variables will seriously affect these results.

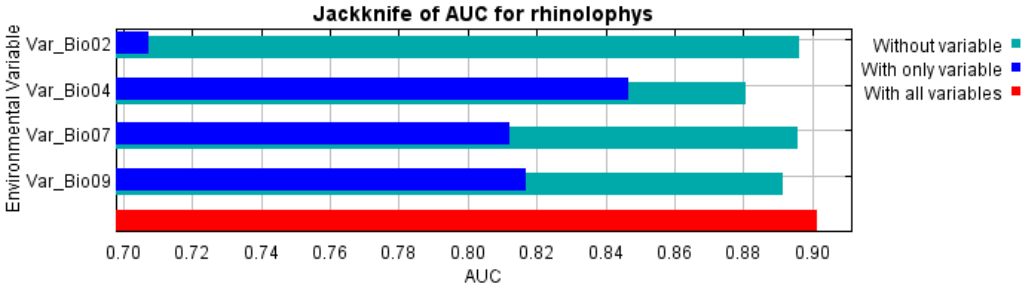
- **The Jackknife test consist** of multiple tests which include or exclude a specific variable from the model. Each report how much "gain" did X variable contribute or not.
  - Training gain refers to testing during the model training phase
  - Test gain refers to tests during using the data left out for validation
  - AUC gain refers to variations on the AUC accuracy using the TEST data.



The next picture shows the same jackknife test, using test gain instead of training gain. Note that conclusions about w



Lastly, we have the same jackknife test, using AUC on test data.



These two previous reports are usually very important for ecologists because they provide a direct hint into understanding if the model did in capture known ecological ideas about the species. In my case, Temperature Seasonality (BIO04) played the most important role in the distribution of the species. And locations with lower seasonality (look at response curves), and it makes sense, since these bats inhabit caves which help maintain a stable temperature. Locations with high levels of seasonality would imply that also the temperature in these caves would change **(I am guessing all this! I am not an ecologist).**

These are main model fitness and performance statistics that MAXENT software provides on the main page. There is other information's worth exploring though.

Go to one of the models, scroll to the top of the page press one of the hyperlinks, e.g [0] and explore what is there.

While much of the information is similar, they are the results related with that specific model run, there a section in the end that is interesting:

Besides seeing the specific omission rate and AUC-ROC curve you now have a table that shows a list of thresholds. Among these, most widely used ones are: '10 percentile training presence', 'Equal training sensitivity and specificity', and 'Maximum training sensitivity plus specificity'

Cumulative threshold	Logistic threshold	Description	Fractional predicted area	Training omission rate
1.000	0.045	Fixed cumulative value 1	0.549	0.003
5.000	0.162	Fixed cumulative value 5	0.404	0.032
10.000	0.259	Fixed cumulative value 10	0.331	0.081
0.797	0.038	Minimum training presence	0.568	0.000
11.844	0.282	10 percentile training presence	0.311	0.099
22.934	0.392	Equal training sensitivity and specificity	0.227	0.226
13.130	0.296	Maximum training sensitivity plus specificity	0.299	0.108
21.368	0.382	Equal test sensitivity and specificity	0.237	0.209
9.563	0.253	Maximum test sensitivity plus specificity	0.336	0.077
1.868	0.078	Balance training omission, predicted area and threshold value	0.497	0.006
5.732	0.180	Equate entropy of thresholded and original distributions	0.390	0.035

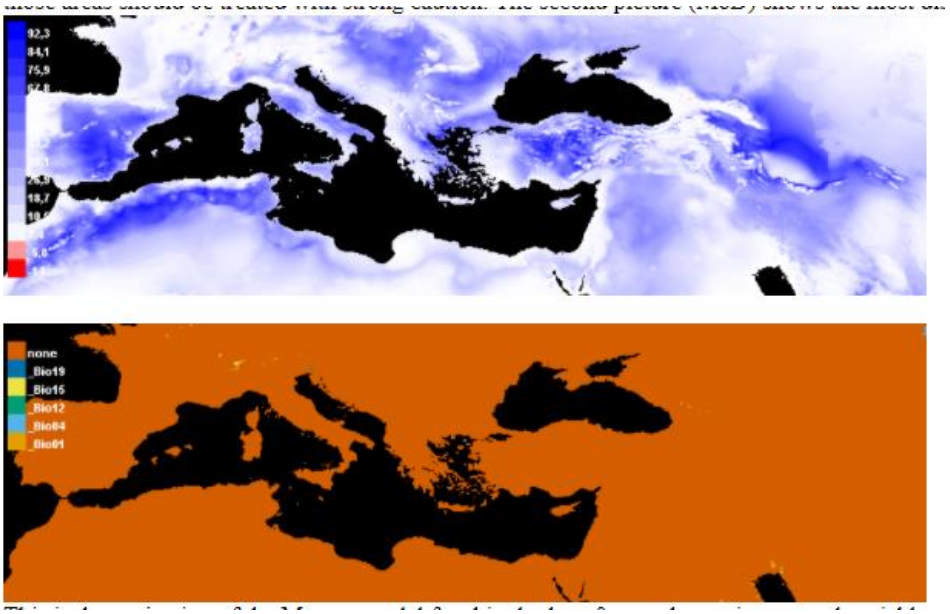
The question here is, from what probability between 0 and 1, should we consider that the species is present? In a classic approach, you would think that any probability above 50% is a guess better than "random". But in truth, this probability depends on the prevalence of the species and on other factors. Therefore, selecting this threshold becomes a specific problem in SDM. **So, again, if a species is prevalent you might want to minimize your Type II error (false negative) because predicting where it is not, is harder than predicting where it is. The opposite applies for a specialist species, perhaps you want to minimize the Type I error.** If you want to "balance" the Tipe I and Tipe II errors, you can choose the "Equal test sensitivity and specificity" which generally applies for most cases.

These thresholds are related to the concept of confusion matrix. On the table, you have multiple options for the threshold value which maximize or minimize specific aspects of the data. Basically, what the software is doing is varying the threshold in an interval (in 0 to 1) and reporting which thresholds maximize what is described: **e.g. 0.382 ensures that your test sensitivity is equal to the specificity – which means that your model is equally likely to predict true positives (sensitivity) and true negatives (Specificity)**.

For one of the next steps (a change detection map) you will need to set a threshold. To select the threshold value for your model:

- Go to each different run [0][1]….[N] and take note of the "Logistic threshold" for the "Equal sensitivity and specificity" criteria.
- **Your final threshold value will be the average of all the thresholds.**
    - **You can add this mean + standard deviation of the threshold value to your report**

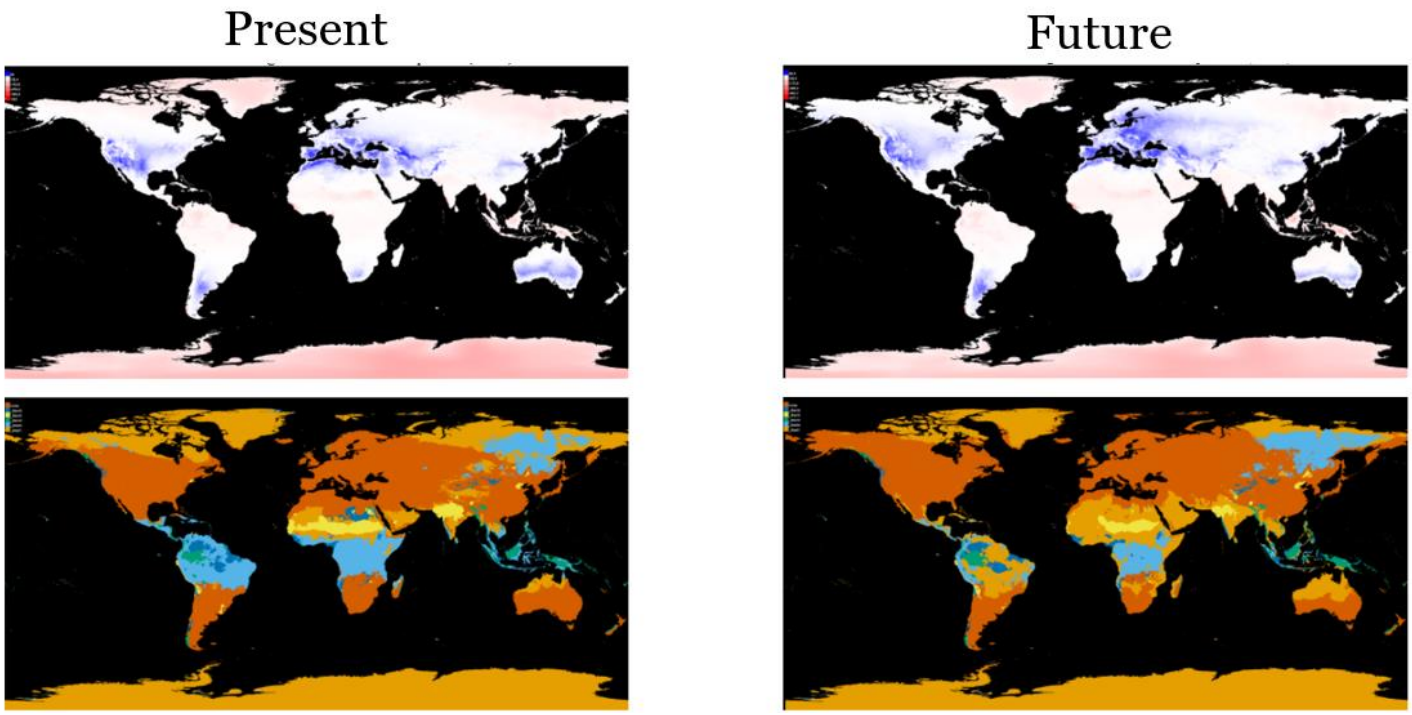
- Find the picture where Multivariate Environmental Similarity Surfaces (MESS) and "Most dissimilar variable" (MoD) are shown. For the formulas, see the Appendix of Elith, 2010 page 3



- These two results show you two specific things for the run [0] (or the one you selected)
    - Where in space there was the most difference between the data used for modelling and the data available -> **meaning the areas where you are likely extrapolating**.
    - And on the bottom, **it tells which variable is most dissimilar** – e.g. no variable is really different in the above case
- **This is VERY important when** analysing future scenarios because it tells the researcher something about the uncertainty of the predictions. The more extrapolation that exists, the more unlikely is your prediction.
    - Notice that, options like fade by clamping, extrapolate etc on the model setting will affect the above estimates.
- **MESS and MOD are also provided as .asc files and are on the main output folder of MAXENT.** How to combine them and then use them, is up to you if you want to go for the extra credit. **(PSS: averages & modes might be a simple and fair enough approach!)**

A final interesting map available on the individual report of each which provides the "absolute difference between using clamping or not using clamping" is visible in this figure:

If I look the prediction to the present time and future at a global scale:



It seems there are not that many differences for the training ranges (Mediterranean) but some visible changes in the tropical areas and around the equator and in the colder regions of the north. Overall this means that in my case, for most of the world and especially my study area around the Mediterranean, Clamping did not have that much of a significant impact. Which is great, I can at least generally my results to the study area.

Next, we calculate the species range, so remember to calculate the mean value of the threshold you selected because you will need it next.

Calculating range-shift changes and the change map:

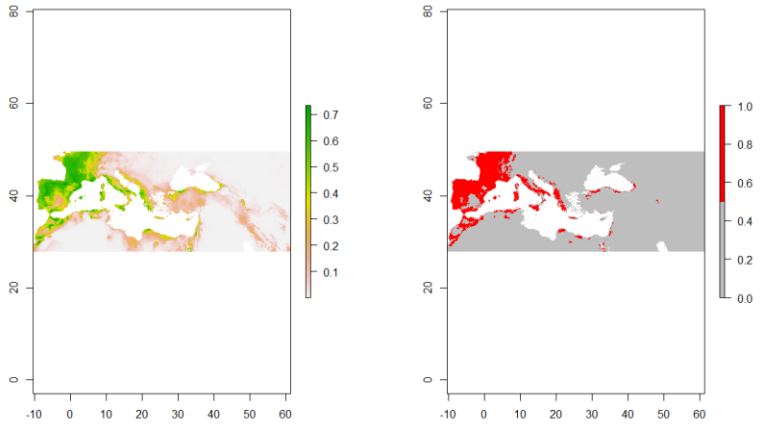
This is a very straightforward section. The objective here is to calculate where there was loss and gain of species habitat, generate a nice-looking raster that you can then use in ArcGIS to make a nice-looking map. You can also use this information to compare how much are was gained or lost in total for each of the scenarios. For this section, we will use **03_Making_ChangeMaps.R**

Step by step:

- Load the needed packages, the rasters and apply the chosen threshold:

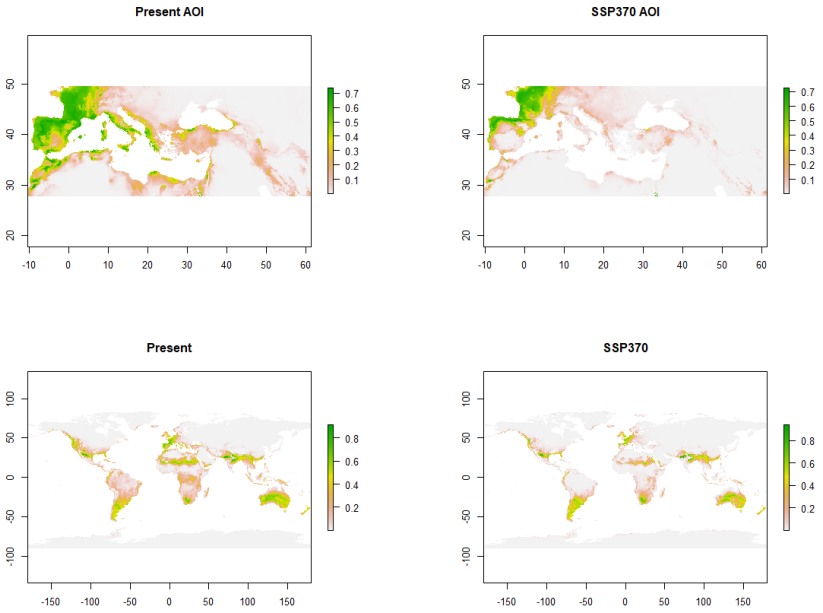
```
gc()
#Packages
library(sp)
library(rgdal)
library(raster)
library(biomod2)

#species used: Rhinolophus_euryale_csv2
#wordclim version: 2.0
#future variables scenario:HadGEM2ES_RCP85

#checks where your R is currently working
getwd()
#sets a new working directory
setwd("C:/Practical/")


#Create Binary map based on Threshold
prob.rst.aoi <- raster("./Maxent/EXP02/rhinolophys_avg.asc") # load present AOI distribution

#set the threshold based on the results
th <- 0.362 #define threshold
pres.rst.aoi <- prob.rst.aoi >= th

par(mfrow=c(1,2))
plot(prob.rst.aoi)
plot(pres.rst.aoi,col=c("gray","red"))
```

- The command par(mfrow=c(1,2)) tells r that the next plot should have 1 row and 2 columns:



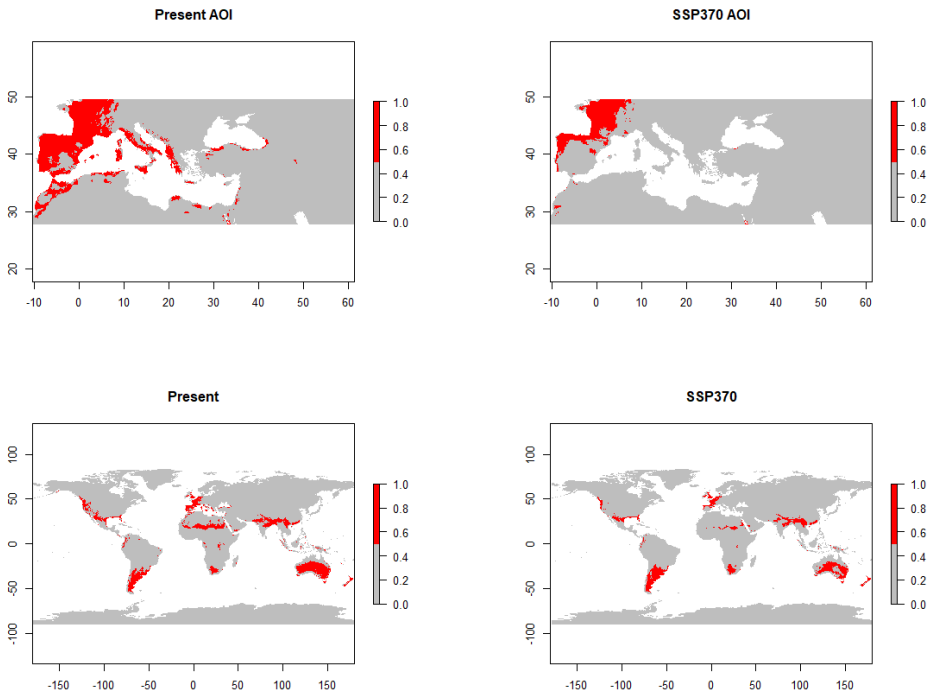
- Let us adapt this code for the remaining scenarios and plot them:

```
#first load the rest of the scenarios:

#Projection for the future scenarion ssp370, on the AOI
prob.rst.aoi.ssp370 <- raster("./Maxent/EXP02/rhinolophys_FutureAOI_avg.asc")

#Projection for current historical data for the entire world
prob.rst.wld <- raster("./Maxent/EXP02/rhinolophys_PresentWLD_avg.asc")

#Projection for current historical data for the entire world
prob.rst.wld.ssp370 <- raster("./Maxent/EXP02/rhinolophys_FutureWLD_avg.asc")

#lets visualize
par(mfrow=c(2,2))
plot(prob.rst.aoi,main="Present AOI")
plot(prob.rst.aoi.ssp370,main="SSP370 AOI")
plot(prob.rst.wld,main="Present")
plot(prob.rst.wld.ssp370,main="SSP370")
```



Immediately, we can see that there are definite range shifts in the probability of distribution. We can also now just generate the same output but imposing the threshold.

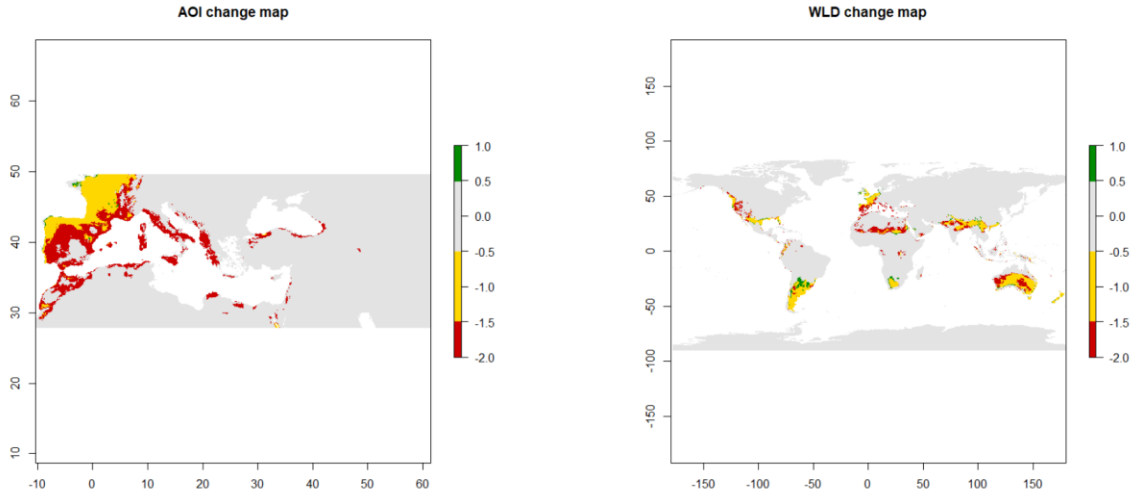
```
#thresholding the data
pres.rst.aoi <- prob.rst.aoi >= th
pres.rst.aoi.ssp370 <- prob.rst.aoi.ssp370 >= th
pres.rst.wld <- prob.rst.wld >= th
pres.rst.wld.ssp370 <- prob.rst.wld.ssp370 >= th

#lets visualize
par(mfrow=c(2,2))
plot(pres.rst.aoi,main="Present AOI",col=c("gray","red"))
plot(pres.rst.aoi.ssp370,main="SSP370 AOI",col=c("gray","red"))
plot(pres.rst.wld,main="Present",col=c("gray","red"))
plot(pres.rst.wld.ssp370,main="SSP370",col=c("gray","red"))
```



Which confirms our suspicion that there was some change in the suitability areas. But where did this range shift occur? And where was it a gain or a loss?

We can easily visualize this using the BIOMOD_RangeSize function in the Biomod2 package:

```
#we can use the biomod package:
range_aoi <- BIOMOD_RangeSize(CurrentPred=pres.rst.aoi,FutureProj=pres.rst.aoi.ssp370,SpChange.Save=NUL
range_wld <- BIOMOD_RangeSize(CurrentPred=pres.rst.wld,FutureProj=pres.rst.wld.ssp370,SpChange.Save=NUL

col.lst <- c("red3" , "gold", "grey89" , "green4") #define plot colours

par(mfrow=c(1,2))
plot(range_aoi$Diff.By.Pixel,col=col.lst,main="AOI change map") #plot range change
plot(range_wld$Diff.By.Pixel,col=col.lst,main="WLD change map") #plot range change
```



Information the range size and how much it varied in which direction is stored within the object that the BIOMOD_RangeSize function calculates.

```
> range_aoi
$Compt.By.Models
      Loss Stable0 Stable1 Gain PercLoss PercGain SpeciesRangeChange CurrentRangeSize FutureRangeSize.NoDisp
layer 14037  149447    8277  207   62.907    0.928           -61.979           22314                 8277
      FutureRangeSize.FullDisp
layer                     8484

$Diff.By.Pixel
class      : RasterStack
dimensions : 261, 862, 224982, 1  (nrow, ncol, ncell, nlayers)
resolution : 0.08333333, 0.08333333  (x, y)
extent     : -10.33333, 61.5, 27.83333, 49.58333  (xmin, xmax, ymin, ymax)
crs        : NA
names      : layer
min values :    -2
max values :     1


> range_wld
$Compt.By.Models
      Loss Stable0 Stable1  Gain PercLoss PercGain SpeciesRangeChange CurrentRangeSize FutureRangeSize.NoDisp
layer 88596 2905195  151248 23367   36.939    9.743           -27.196          239844                151248
      FutureRangeSize.FullDisp
layer                   174615

$Diff.By.Pixel
class      : RasterStack
dimensions : 2160, 4320, 9331200, 1  (nrow, ncol, ncell, nlayers)
resolution : 0.08333333, 0.08333333  (x, y)
extent     : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
crs        : NA
names      : layer
min values :    -2
max values :     1
```

	R variables	Description	PixelCounts	Metrics	Percentage	Formula	R variables
Habitat change	Gain	Gain	23367	Habitat gain	9,743%	Gain/(Disa+Stable1)	PercLoss
	Stable0	Maintained (Absent -> Absent)	2905195				
	Stable1	Maintained (Present -> Present)	151248	Habitat loss	36,939%	Disa/(Disa+Stable1)	PercGain
	Disa	Loss	88596				
	CurrentRangeSize	Current range size	239844	Species range change	-27,196%	Habitat Gain - Habitat loss	SpeciesRangeChange
	FutureRangeSize0Disp	Future range size (no migration)	151248				
	FutureRangeSize1Disp	Future range size (migration)	174615				

We can confirm then that there was a loss of ~27% of the total area suitable for this species which means that while this species is negatively affected by climate change in the coming ~50 years, it is not critically at risk.

All this analysis could be done in many ways, both in R but also using a GIS software. Once we have the final probability raster's, it all becomes about using different operations that are available everywhere. **Let us save our outputs to the disc, define their projections so we can easily open them in a GIS and use it to create the maps for your report.**

Exporting final data to more GIS friendly files:

```
## Defining the coordinate system AND saving the files as a geotif

#all data is actually in WGS84, meaning, geographic coordinates
WGS84 <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")

#probability maps
projection(prob.rst.aoi) <- WGS84
projection(prob.rst.aoi.ssp370) <- WGS84
projection(prob.rst.wld) <- WGS84
projection(prob.rst.wld.ssp370) <- WGS84
#presence absence maps
projection(pres.rst.aoi) <- WGS84
projection(pres.rst.aoi.ssp370) <- WGS84
projection(pres.rst.wld) <- WGS84
projection(pres.rst.wld.ssp370) <- WGS84
#change maps
#in this case we need to fetch first the raster from the list object
rst.aoi.change <- range_aoi$Diff.By.Pixel
rst.wld.change <- range_wld$Diff.By.Pixel
projection(rst.aoi.change) <- WGS84
projection(rst.wld.change) <- WGS84
```

And now, using all that we have learned, we can just save the files to a new folder to make it easier to find. **Remember to create the "FinalOutputs" folder in your workspace.**

```r
#create a nice final directory
dir.create("./FinalOutputs")

#writing the probability maps
writeRaster(prob.rst.aoi,"./FinalOutputs/EXP02_ProbMap_AOI.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))

writeRaster(prob.rst.aoi.ssp370,"./FinalOutputs/EXP02_ProbMap_AOI_SSP370.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))

writeRaster(prob.rst.wld,"./FinalOutputs/EXP02_ProbMap_WLD.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))

writeRaster(prob.rst.wld.ssp370,"./FinalOutputs/EXP02_ProbMap_WLD_SSP370.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))
```

```r
#writing the presence maps
writeRaster(pres.rst.aoi,"./FinalOutputs/EXP02_PresMap_AOI.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))

writeRaster(pres.rst.aoi.ssp370,"./FinalOutputs/EXP02_PresMap_AOI_SSP370.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))

writeRaster(pres.rst.wld,"./FinalOutputs/EXP02_PresbMap_WLD.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))

writeRaster(pres.rst.wld.ssp370,"./FinalOutputs/EXP02_PresMap_WLD_SSP370.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))

#writing the change maps
writeRaster(rst.aoi.change,"./FinalOutputs/EXP02_Change_AOI.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))

writeRaster(rst.wld.change,"./FinalOutputs/EXP02_Change_WLD.tiff",
            overwrite=T,
            options=c("COMPRESS=LZW"))
```

And now, your new folder should have everything there:

EXP02_Change_AOI	18/11/2020 22:38	Ficheiro TIF	48 KB
EXP02_Change_WLD	18/11/2020 22:38	Ficheiro TIF	1 169 KB
EXP02_PresbMap_WLD	18/11/2020 22:40	Ficheiro TIF	1 120 KB
EXP02_PresMap_AOI	18/11/2020 22:40	Ficheiro TIF	45 KB
EXP02_PresMap_AOI_SSP370	18/11/2020 22:40	Ficheiro TIF	39 KB
EXP02_PresMap_WLD_SSP370	18/11/2020 22:40	Ficheiro TIF	1 099 KB
EXP02_ProbMap_AOI	18/11/2020 22:39	Ficheiro TIF	843 KB
EXP02_ProbMap_AOI_SSP370	18/11/2020 22:39	Ficheiro TIF	850 KB
EXP02_ProbMap_WLD	18/11/2020 22:39	Ficheiro TIF	15 614 KB
EXP02_ProbMap_WLD_SSP370	18/11/2020 22:40	Ficheiro TIF	15 768 KB

And you should also be able to use all these files seamlessly in a GIS for further analysis and to make even nicer maps. **PS: Adapting these to go fetch the MESS & MOD data should be trivial by now. Perhaps you can also bring those to the GIS and use them to explain your results.**

Remember the "Golder rules of Cartography":

https://www.wwu.edu/huxley/spatial/tut/ALL_GOOD_MAPS.pdf

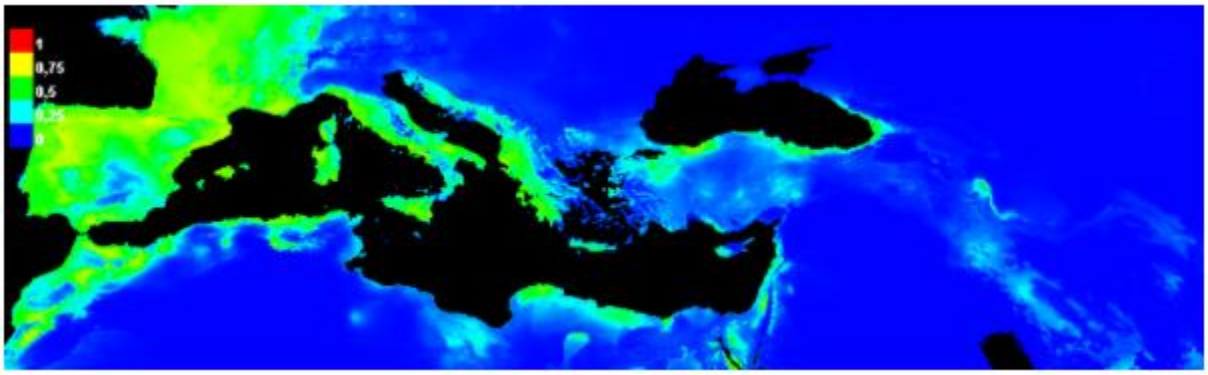
1. Title
2. Scale
3. Orientation
4. Border
5. Legend
6. Authorships and data provenance
7. Detail (if needed)
8. Effective graphical design
9. Visual hierarchy
10. **PURPOSE**

These instructions will certainly be useful for the maps in the report.

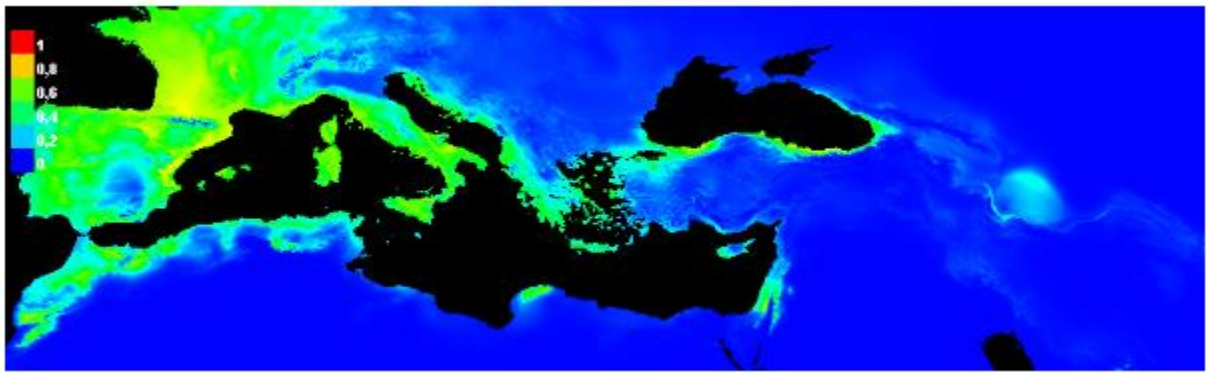
A curve ball:

During the preparation of this tutorial I ran the model with the wrong variables and this is the result I got:

With the right set of variables: AUC= 0.899



With the wrong set of variables: AUC = 0.901



The models are very similar with the accuracy of the wrong model being slightly better. How do you interpret this?

If you want to take a shot at explaining: n.q.cesar.sa@cml.leidenuniv.nl

And notice:

Variable	Percent contribution	Permutation importance
_Bio04	52.9	50.8
_Bio12	36.3	31.9
_Bio01	6.6	12.9
_Bio15	3.2	2.6
_Bio19	1.1	1.9

Variable	Percent contribution	Permutation importance
Var_Bio04	61.7	60.6
Var_Bio09	31.9	21.2
Var_Bio02	4	4.7
Var_Bio07	2.4	13.5

Models trained with the wrong variables can have better accuraccy and also lead researchers towards the wrong conclusions..

Common R commands:

List (Idaho university): https://www.webpages.uidaho.edu/~stevel/251/comR.pdf

This is not an extensive list at all, just some examples that might be useful for the exercise.

Dataframes: (df)

- This is the most common object in R. It is the equivalent to a an excel table where (in general) rows represent samples and columns different measurements of those samples.

Importing data as dataframe: http://www.r-tutor.com/r-introduction/data-frame/data-import

summary(df): produces a summary of the data that is in the dataframe

head(df): prints the first (10) rows of the dataframe, lets you quickly investigate big datasets (tail(df) the same but from the bottom).

rownames and colnames(df): prints the row names (often just a sequential 1 to N rows but not always) or the column names of your dataframe.

Changing the dataframe:

df[i, j] <- let's you change the data in row i and colum j.

df$Newcolumnname <- lets you add a new column to your dataframe. Notice that the values of this column will be added following different rules (https://www.datamentor.io/r-programming/data-frame/)

rbind & cbind <- let's you "collate" a data frame with another dataframe as a row or as a column. Notice that rbind expects that both dataframes have the same column names while cbind expects that both dataframe have the same number of columns.

Loading data as dataframe from csv/txt etc:

Raster: (rst)

- Different R packages deal with raster types in different ways. The most commonly used package is probably the raster package, so these examples relate to it.

Loading raster data from files in the computer:

- raster("path to the file in text", band=1) loads the raster, by default the first band. It is a single layer raster.
- stack("path to the file in text") loads all bands of the raster into a multilayer raster

names(rst) <- returns the names of the raster layers. Also lets you change the names of the layer.

projection(rst) <- Returns the current projection of the R object (also applies for spatial data frames). Attention: you can use this to change the projection but not to REPROJECT it's equivalent to the define projection in ArcGIS.

CRS(R spatial object) <- it's a function of the sp package that creates a CRS class object which can be used to define projection of objects)

projectRaster(from, to, res, method="bilinear",…) <- this function actually projects a raster in coordinate system X to coordinate system Y. Can be called in a different way also. This is the function you need to use if you want to go from e.g. WGS84 to a projected coordinate system.

writeRaster(rst,filename,format,bylayer,suffix,…) <- this functions saves the raster data into your folders. This function has many different options:

- bylayer = True ; tells R to store each different layer into a different raster file
- suffix = "numbers" or "names" ; tells R to store each layer using the order in the multilayer raster or the actual layer name.
- You can also change the type of "background value", or "nodata" value using the background = <some value>
- Most importantly, you can tell R to compress your file by adding (example for GeoTiff files):

```
writeRaster(NDI,
            "PATH2OUTPUTDIRECTORYANDFILENAME.tif",
            options=c("COMPRESS=LZW"),
            overwrite=TRUE)
```

Many raster functions allow to to store your raster at the same time you are using them. For example, rasterize function transforms spatial data (e.g. points) to raster data: (source)

```
rasterize(myEM.shp,
          raster(cli.pca.crop.pt,layer=1)*0,
          field=names(myEM.shp)[2],
          #background=0,
          filename="path2folder/my_PT_EMpredict.tif",
          options=c("COMPRESS=LZW"),
          overwrite=TRUE)
```

Finally, the raster package allows you to use models trained in R (e.g. some arbitrary linear model):

```
model.glm <- glm(Presence~.,data=train.df,family = "binomial")
```

```
my.prob.raster <- predict(my.pred.rst.stack,
                          model=model.glm,type="response")
```