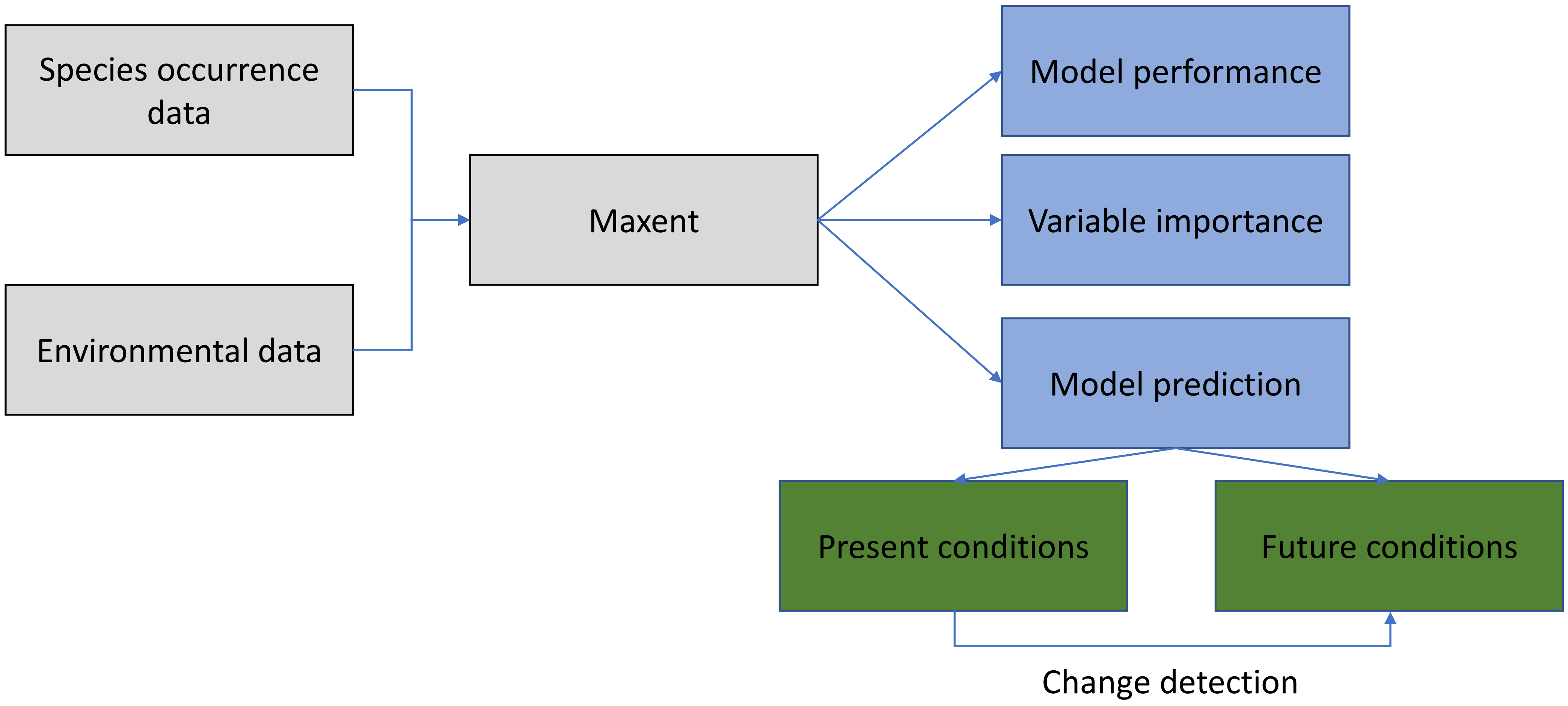
SDM Practical

SDM Pratical overview:



SDM – Design your experiment

- <u>Think before starting</u>!
  - What do you want to know about the species?
  - How do you verify that?

- <u>Set an hypothesis</u>:
  - E.g. Climate change will decrease the area available for species X
    - Thus: $HO\text{: }There\ is\ no\ change\ in\ the\ suitable\ area\ of\ species\ X$

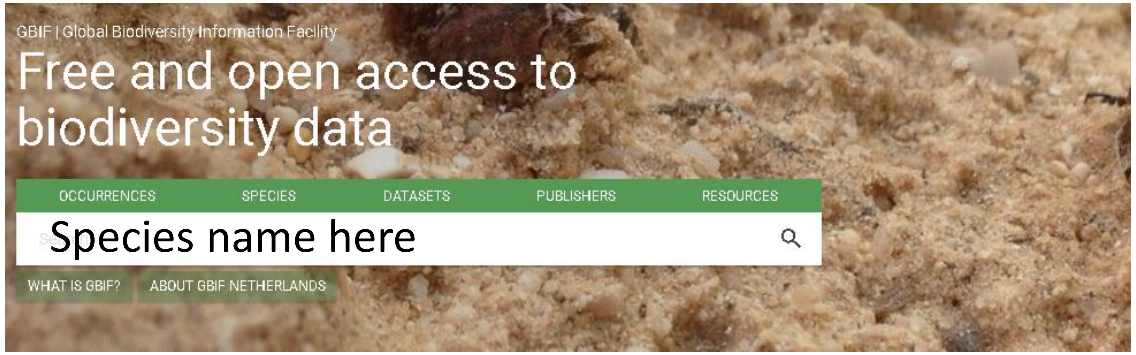
      $HA\text{: }There\ is\ a\ decrease\ of\ the\ suitable\ area\ of\ species\ X$

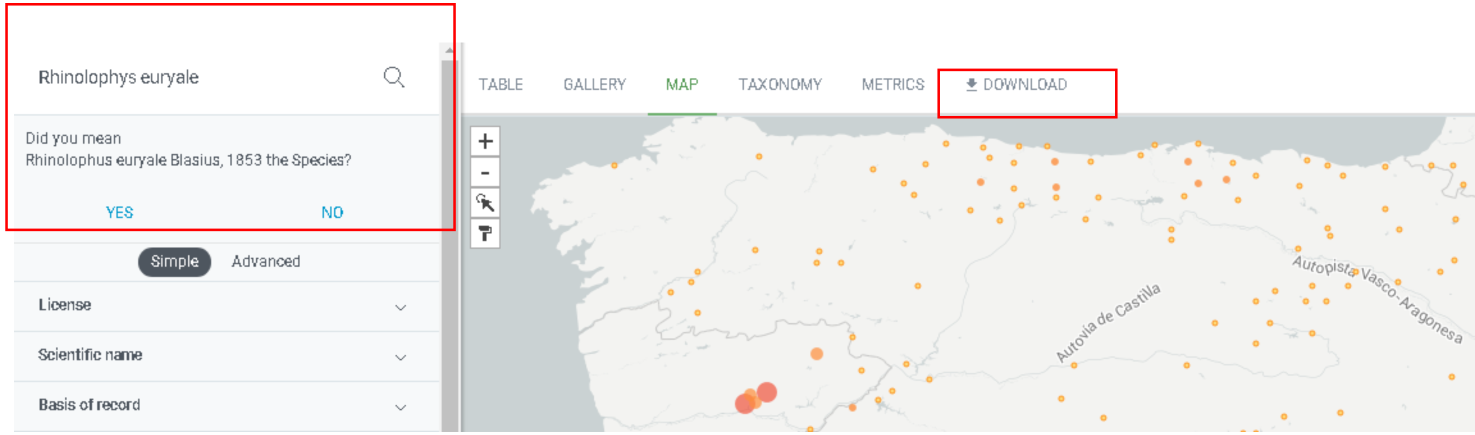
- <u>Draft the steps </u>to test said hypothesis:

Be as much formal as you possibly can

SDM Pratical: Getting # ready

- Goto: https://www.gbif.org/



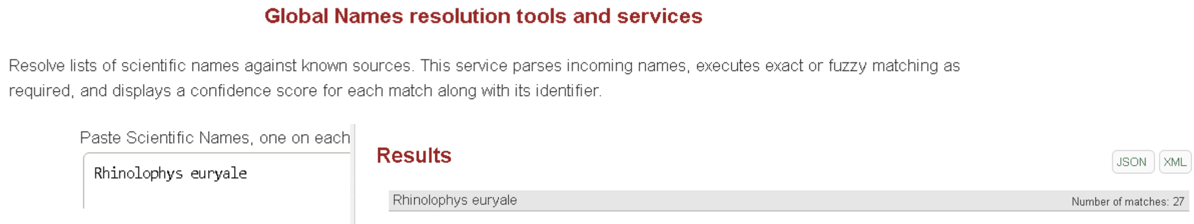


On the options on the left, make sure to only select occurrences with geographic information (and perhaps only human observation)

Does it have synonyms?

https://resolver.globalnames.org/

27 synonyms for this bat species



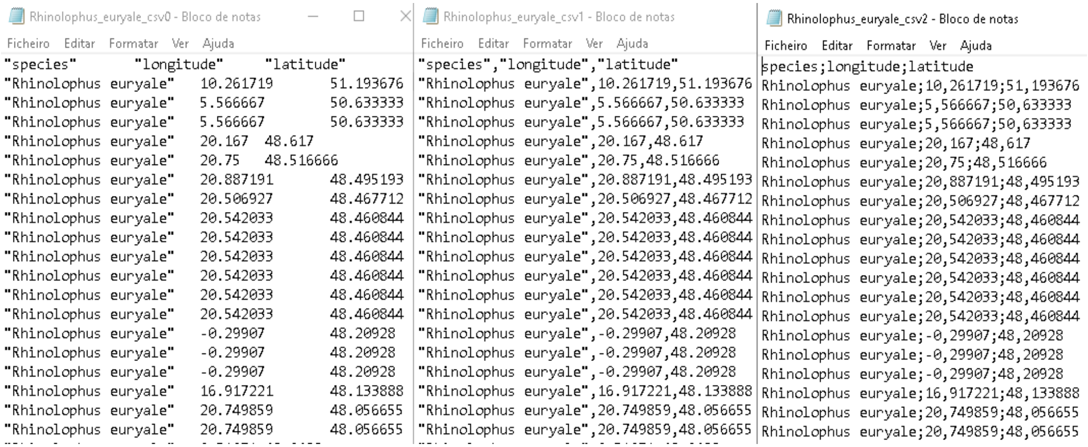
SDM Pratical: Getting # ready

- CSV – comma separated variables
  - Comma's have different meanings:
  - EU (in general): Comma is a decimal separator
  - USA: point is the decimal separator
  - Also tab delimited (decimal can change here)

- R can read all types

- Correcting from csv1 to csv2
  - Open in notepad
  - Substitute , for ;
  - Substitute . for ,

- There are other methods (check the manual)



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"	16.917221	48.133888
"Rhinolophus euryale"	20.749859	48.056655
"Rhinolophus euryale"	20.749859	48.056655
```

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",16.917221,48.133888
"Rhinolophus euryale",20.749859,48.056655
"Rhinolophus euryale",20.749859,48.056655
```

Rhinolophus_euryale_csv2 - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

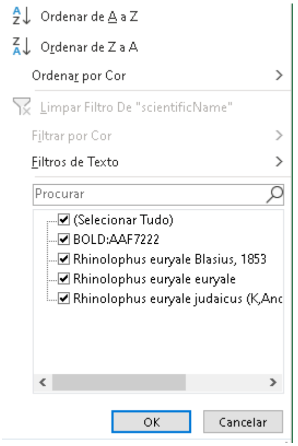
```
species;longitude;latitude
Rhinolophus euryale;10,261719;51,193676
Rhinolophus euryale;5,566667;50,633333
Rhinolophus euryale;5,566667;50,633333
Rhinolophus euryale;20,167;48,617
Rhinolophus euryale;20,75;48,516666
Rhinolophus euryale;20,887191;48,495193
Rhinolophus euryale;20,506927;48,467712
Rhinolophus 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
Rhinolophus euryale;16,917221;48,133888
Rhinolophus euryale;20,749859;48,056655
Rhinolophus euryale;20,749859;48,056655
```

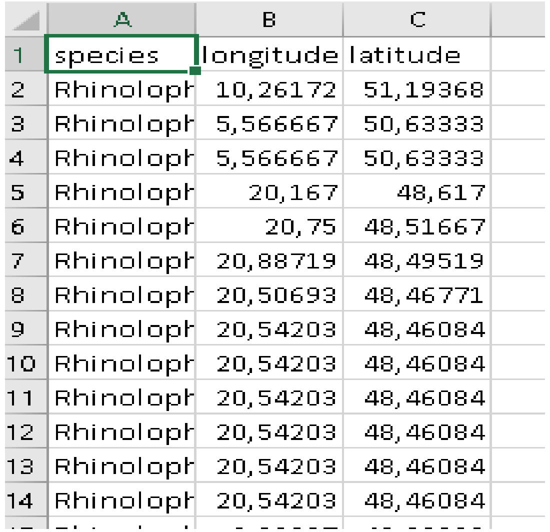
ArcGIS, R, excel, etc, each of them might recognize numbers different <u>so keep all files</u>

SDM Pratical: Getting # ready

- Maxent expects the data in CSV
  - Species name; longitude; latitude
    - MAKE SURE YOU HAVE ONE NAME PER SPECIES
    - When you download from GBIF, it usually brings multiple names
  - Yes, you can have multiple species in one file
    - Yes you can have multiple subsets of the same species in one file: e.g. amanita1, amanita2
      - They will all be modelled independently

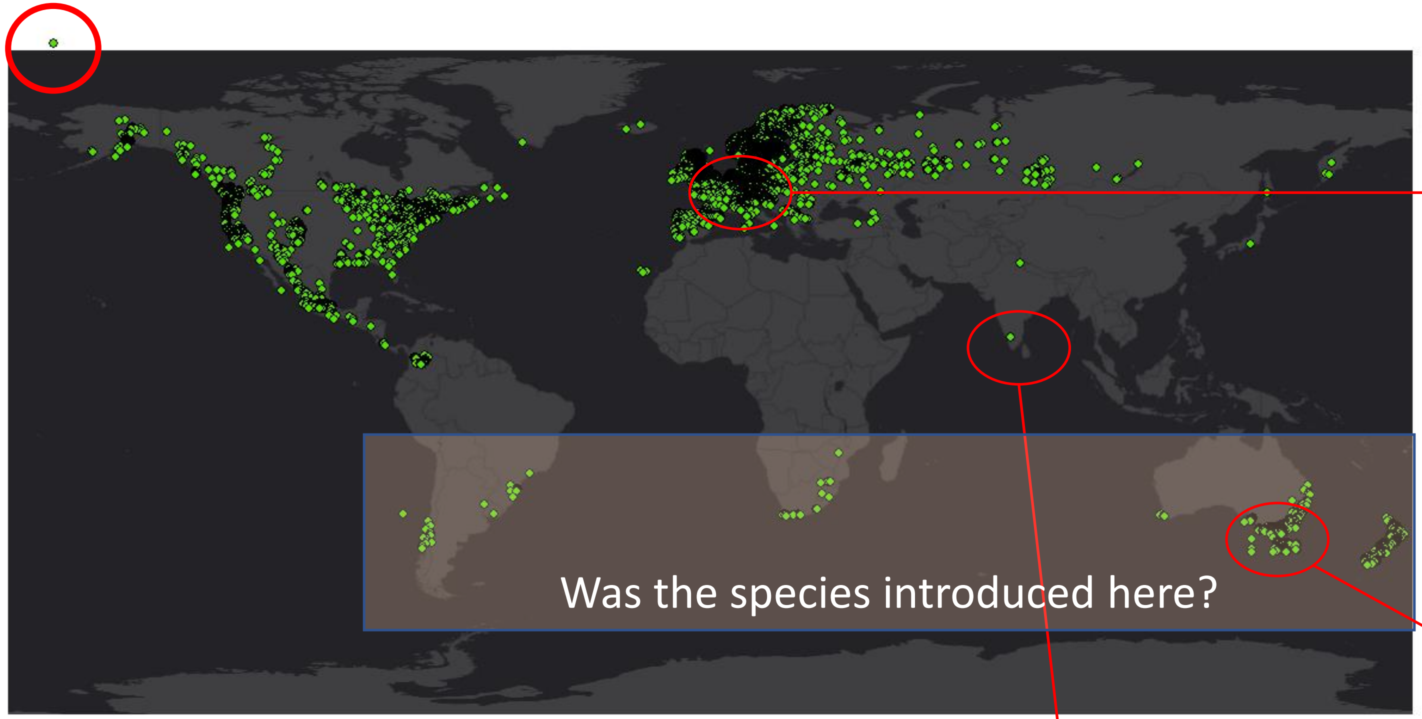
- IMPORTANT STEP:
  - **<u>Load your CSV into a GIS and explore your #</u>**

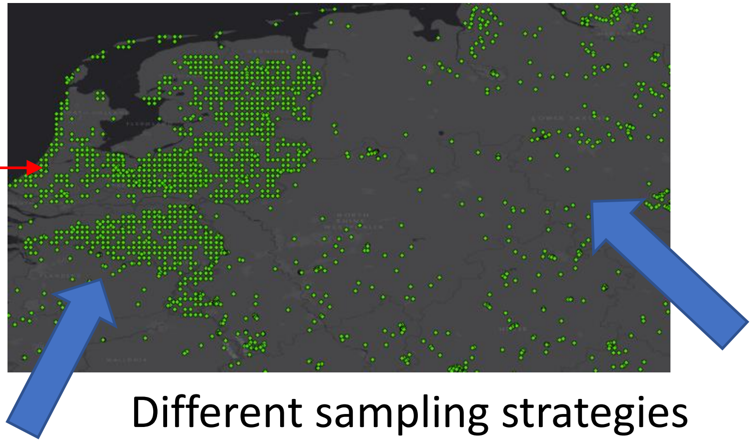




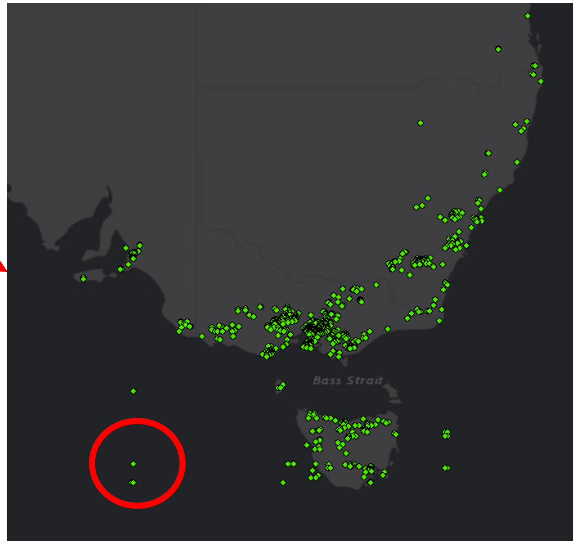
	A	B	C
1	species	longitude	latitude
2	Rhinoloph	10,26172	51,19368
3	Rhinoloph	5,566667	50,63333
4	Rhinoloph	5,566667	50,63333
5	Rhinoloph	20,167	48,617
6	Rhinoloph	20,75	48,51667
7	Rhinoloph	20,88719	48,49519
8	Rhinoloph	20,50693	48,46771
9	Rhinoloph	20,54203	48,46084
10	Rhinoloph	20,54203	48,46084
11	Rhinoloph	20,54203	48,46084
12	Rhinoloph	20,54203	48,46084
13	Rhinoloph	20,54203	48,46084
14	Rhinoloph	20,54203	48,46084

SDM Pratical: Getting # ready





Different sampling strategies



Floating mushrooms

In a GIS you can easily clean the obvious errors

- Further explore if the data "makes sense"
- When satisfied, export back to csv (check GIS notes)

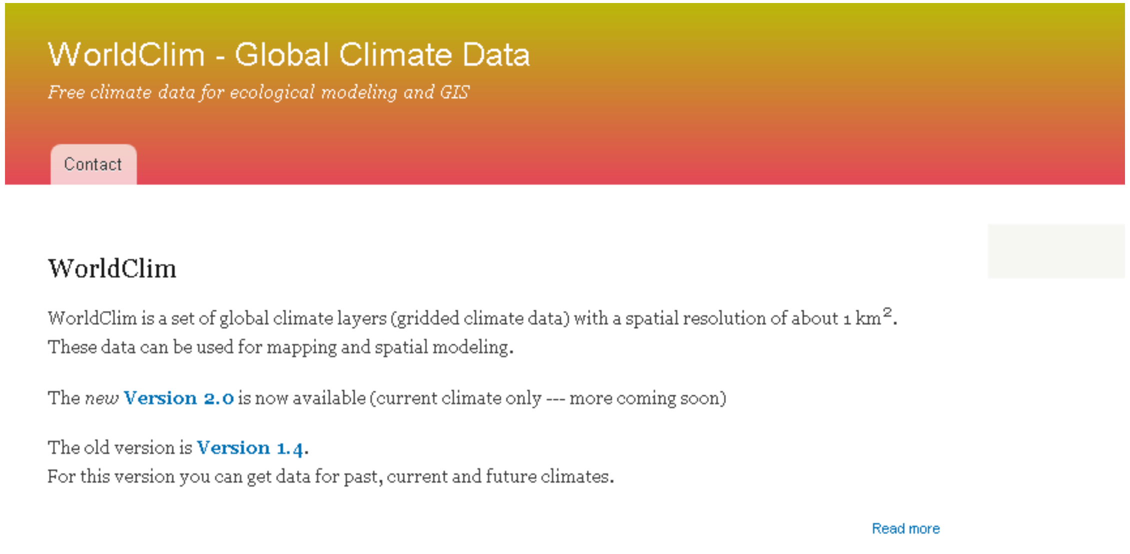
Hum…

Getting climate variables ready

- Goto: https://www.worldclim.org/
  - Worldclim version 2: 1970~2000
    - Tif format (more GIS friendly)
    - Only “Present”
  - Worldclim version 1.4: 1960~1990
    - Bil format
    - Also a section for Future and Current

  - You can use either worldclim version but future scenarions are only possible on the previous version

  - Next:
    - Variable description
    - Coordinate system and resolution



You can also directly download in R:

Example @ https://www.gis-blog.com/r-raster-data-acquisition/

Getting climate variables ready

Main product of Wordclim dataset

- Monthly means of each of these estimates

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

Coordinate system: World Geodetic System 84(WGS84)

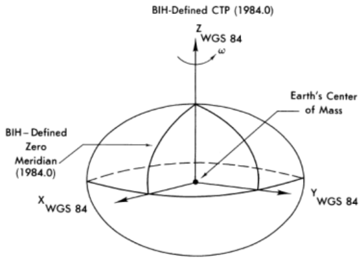


Figure 1.1. WGS 84 Reference Frame

Units are degrees, minutes, seconds

Proper term: Geographic coordinates

These are secondary products that "sumarize" the main products

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

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

Resolution at the equator!
Conversion to m explained in: https://en.wikipedia.org/wiki/Decimal_degrees
X (m) = 30(s) * Perimeter@equator/360(º) = 927.66m @ equator AKA 1km layer
Download the 5 minute layer! (10km @ equator)

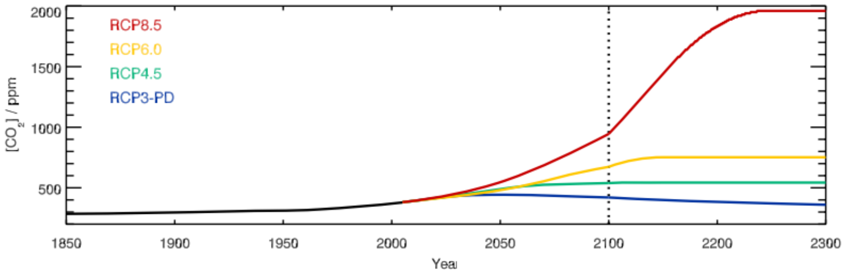
Getting climate variables ready

2050 → Year the scenario represents

GCM	code	rcp26	rcp45	rcp60	rcp85
ACCESS1-0 (#)	AC		tn, tx, pr, bi		tn, tx, pr, bi
BCC-CSM1-1	BC	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
CCSM4	CC	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
CESM1-CAM5-1-FV2	CE		tn, tx, pr, bi		
CNRM-CM5 (#)	CN	tn, tx, pr, bi	tn, tx, pr, bi		tn, tx, pr, bi
GFDL-CM3	GF	tn, tx, pr, bi	tn, tx, pr, bi		tn, tx, pr, bi
GFDL-ESM2G	GD	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	
GISS-E2-R	GS	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
HadGEM2-AO	HD	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
HadGEM2-CC	HG		tn, tx, pr, bi		tn, tx, pr, bi
HadGEM2-ES	HE	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
INMCM4	IN		tn, tx, pr, bi		tn, tx, pr, bi
IPSL-CM5A-LR	IP	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
MIROC-ESM-CHEM (#)	MI	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
MIROC-ESM (#)	MR	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
MIROC5 (#)	MC	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
MPI-ESM-LR	MP	tn, tx, pr, bi	tn, tx, pr, bi		tn, tx, pr, bi
MRI-CGCM3	MG	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi
NorESM1-M	NO	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi	tn, tx, pr, bi

Greenhouse gas scenarios: four representative concentration pathways (RCPs)
Time periods: 2050 (average for 2041-2060) and 2070 (average for 2061-2080)
Variables:

- tn - monthly average minimum temperature (degrees C * 10)
- tx - monthly average maximum temperature (degrees C * 10)
- pr - monthly total precipitation (mm)
- bi - 'bioclimatic' variables



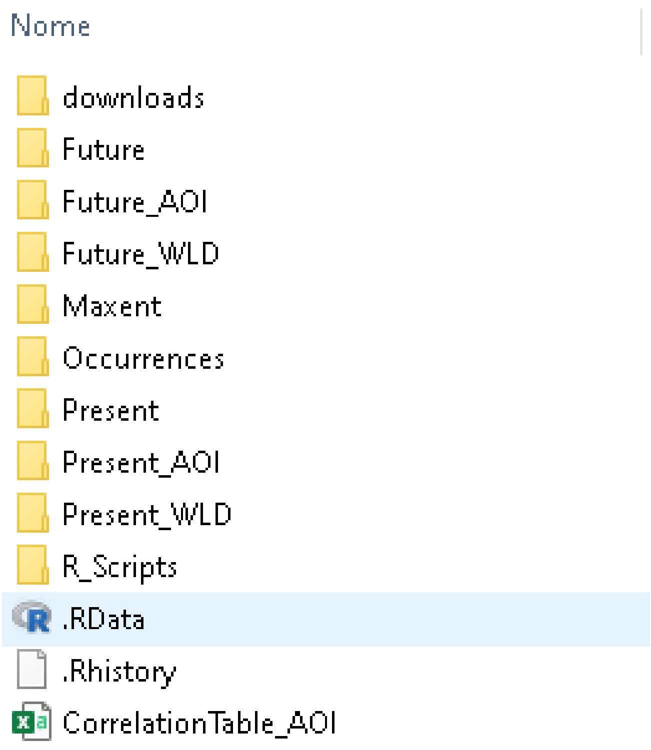
The HadGEM2-ES implementation of CMIP5 centennial simulations

Download the data corresponding to the scenarios you want to test

Notice: you must use the same variables for present and forecasting

Getting climate variables ready

- Be organized!
  - Think of folders as a "database"
  - If you use R script: 00_SettingUpWorkEnvironment.R you will get an environment like mine
- Unzip each .zip file to the appropriate folders
  - My case: Future data to Future folder, Present data to Present folder
- First step: subsetting the data to your Area of Interest
  - Alternative 1: Clip your AOI based on the range of your presence data
  - Alternative 2: Clip your AOI based on a polygon shapefile created in Arcmap
  - Alternative 3: Create your own extent
- You should train your model in an AOI and then predict back to the entire planet
  - If you want to see the future in a specific AOI, you can also clip the future layers
  - We show that on the tutorial but it's optional



Getting climate variables ready

- Overall procedure:
  1. Load the entire group of rasters in the present

  2. Load an object (occurrence points or a shapefile) to crop the rasters to na AOI

  3. Select the variables of interest – using an ecological criteria

  4. Test the selected data for Autocorrelation and VIF
     1. If >.7 or VIF >10, then you have to remove somethig

  5. Save all the data to a new folder in .asc format
     1. Maxent needs it!

  6. Start-up maxent, load the data, set the settings and run

Getting climate variables ready

Check file: 01_CroppingEnvVariables.R

- **Loading rasters:**

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

#lists all file names and paths to the files, saves it to a list
rst.names <- list.files("./Present/",pattern=".bil")[c(1,12:19,2:11)]
rst.fld <-  list.files("./Present/",pattern=".bil",full.names = T)[c(1,12:19,2:11)]
#explore your list to see if you only have rasters on it
rst.fld
#loads all rasters into a stack object (a multi-dimensional raster)
stk.present <- stack(rst.fld)
```

Setwd tells R to consider c:/practical as the “working folder”

This allows you to use relative paths by adding a dot: ./<folder>

If you use a diferent folder then you should change this

List.files command creates a “list” of the files in a specific folder.

By using pattern=“.bil” this list will only consider files with the “.tif” extension

Adding full.names=T means it will actually return the entire path (in this case: c:/Practical/Present /<filename>.tif

The order of the original variables is mixed – so we have to shuffle them
Originally: (Bio1, bio10 … Bio 19, bio 2… bio9)
But we want: Bio1, bio2, …. Bio19

The direction of the slash is significant!!!

Single \ - tells Rm next character is special
Double: \\ - Tells R, next character is a backslash
Forward: / - recognized as a backslash

For R you must use: \\ or / when using a folder path or you will have an error

```
Error: unexpected input in "\"
> setwd("C:\Practical\")
Error: '\P' is an unrecognized escape in character string starting ""C:\P"
> setwd("C:/Practical/")
> |
```

Getting climate variables ready

Now you repeat the same for the scenarios you want to use

```
#loading future variables is a bit more complicated due to the subfolders
path.fut.26 <- list.files("./Future/he26bi50/",pattern=".tif",full.names = T)
path.fut.45 <- list.files("./Future/he45bi50/",pattern=".tif",full.names = T)
path.fut.60 <- list.files("./Future/he60bi50/",pattern=".tif",full.names = T)
path.fut.85 <- list.files("./Future/he85bi50/",pattern=".tif",full.names = T)

stk.fut.26 <- stack(path.fut.26)
stk.fut.45 <- stack(path.fut.45)
stk.fut.60 <- stack(path.fut.60)
stk.fut.85 <- stack(path.fut.85)

#checks the order of the layers loaded and renames them to bioXX, each XX repr
#a bioclimatic layer
names(stk.present)
names(stk.present) <- c("Bio01","Bio02","Bio03","Bio04",
                        "Bio05","Bio06","Bio07","Bio08",
                        "Bio09","Bio10","Bio11","Bio12",
                        "Bio13","Bio14","Bio15","Bio16",
                        "Bio17","Bio18","Bio19")

```

This is to rename the variables of the present period to the bioclim naming – <u>if you do it now, it helps you later when selecting the variables. Everything will be in proper order</u>

Getting climate variables ready

In the case of the future variables they are often loaded out of order, you should check in your case!!

```
> path.fut.26
 [1] "./Future/he26bi50/he26bi501.tif"  "./Future/he26bi50/he26bi5010.tif" "./Future/he26bi50/he26bi5011.tif"
 [4] "./Future/he26bi50/he26bi5012.tif" "./Future/he26bi50/he26bi5013.tif" "./Future/he26bi50/he26bi5014.tif"
 [7] "./Future/he26bi50/he26bi5015.tif" "./Future/he26bi50/he26bi5016.tif" "./Future/he26bi50/he26bi5017.tif"
[10] "./Future/he26bi50/he26bi5018.tif" "./Future/he26bi50/he26bi5019.tif" "./Future/he26bi50/he26bi502.tif"
[13] "./Future/he26bi50/he26bi503.tif"  "./Future/he26bi50/he26bi504.tif"  "./Future/he26bi50/he26bi505.tif"
[16] "./Future/he26bi50/he26bi506.tif"  "./Future/he26bi50/he26bi507.tif"  "./Future/he26bi50/he26bi508.tif"
[19] "./Future/he26bi50/he26bi509.tif"
```

```
names(stk.fut.26)
names(stk.fut.45)
names(stk.fut.60)
names(stk.fut.85)
path.fut.26
#the easiest way to  is just to re-load the variables again with the proper order
path.fut.26 <- list.files("./Future/he26bi50/",pattern=".tif",full.names = T)[c(1,12:19,2:11)]
path.fut.45 <- list.files("./Future/he45bi50/",pattern=".tif",full.names = T)[c(1,12:19,2:11)]
path.fut.60 <- list.files("./Future/he60bi50/",pattern=".tif",full.names = T)[c(1,12:19,2:11)]
path.fut.85 <- list.files("./Future/he85bi50/",pattern=".tif",full.names = T)[c(1,12:19,2:11)]

stk.fut.26 <- stack(path.fut.26)
stk.fut.45 <- stack(path.fut.45)
stk.fut.60 <- stack(path.fut.60)
stk.fut.85 <- stack(path.fut.85)

#check if they loaded fine
names(stk.fut.26)
names(stk.fut.45)
names(stk.fut.60)
names(stk.fut.85)

#and then rename them easily
list.of.names <- c("Bio01","Bio02","Bio03","Bio04",
                   "Bio05","Bio06","Bio07","Bio08",
                   "Bio09","Bio10","Bio11","Bio12",
                   "Bio13","Bio14","Bio15","Bio16",
                   "Bio17","Bio18","Bio19")

names(stk.fut.26) <- list.of.names
names(stk.fut.45) <- list.of.names
names(stk.fut.60) <- list.of.names
names(stk.fut.85) <- list.of.names
```

Now the 10th layer is n the 10th position

```
> path.fut.26
 [1] "./Future/he26bi50/he26bi501.tif"  "./Future/he26bi50/he26bi502.tif"  "./Future/he26bi50/he26bi503.tif"
 [4] "./Future/he26bi50/he26bi504.tif"  "./Future/he26bi50/he26bi505.tif"  "./Future/he26bi50/he26bi506.tif"
 [7] "./Future/he26bi50/he26bi507.tif"  "./Future/he26bi50/he26bi508.tif"  "./Future/he26bi50/he26bi509.tif"
[10] "./Future/he26bi50/he26bi5010.tif" "./Future/he26bi50/he26bi5011.tif" "./Future/he26bi50/he26bi5012.tif"
[13] "./Future/he26bi50/he26bi5013.tif" "./Future/he26bi50/he26bi5014.tif" "./Future/he26bi50/he26bi5015.tif"
[16] "./Future/he26bi50/he26bi5016.tif" "./Future/he26bi50/he26bi5017.tif" "./Future/he26bi50/he26bi5018.tif"
[19] "./Future/he26bi50/he26bi5019.tif"
```

When everything is in proper order, we can name it – Maxent likes it when variables for projections share names with the training variables

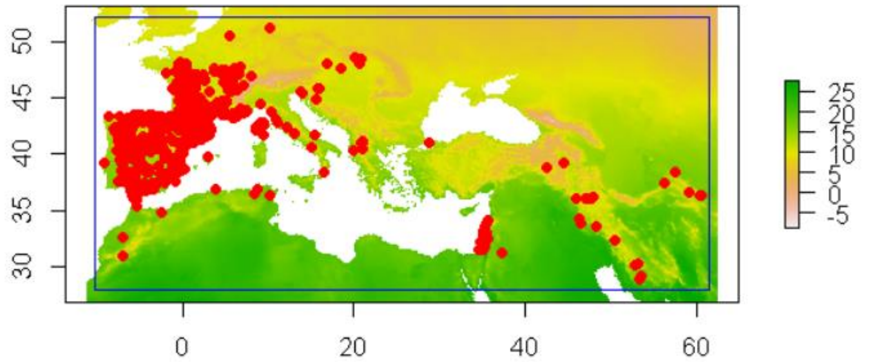
Getting climate variables ready

- Creating an extent object
  - We will clip to the extent of the area where we have occurrences
    - For a polygon area, the exercise is the same (ish).

Here i had a problem – some coordinate values were wrong, if you have an error, eliminate those rows in Excel, save as csv, and repeat

- First load the occurrence data:

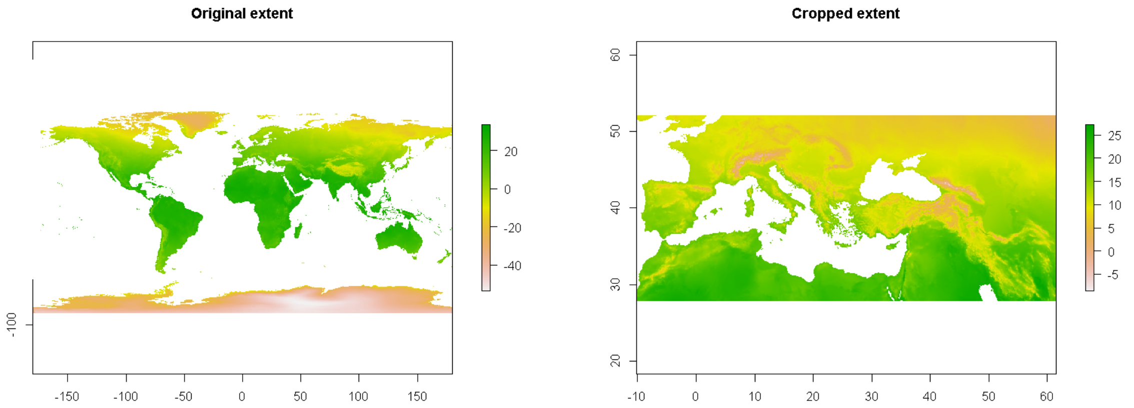
```
names(stk.fut.26)
# Load species occurrence file
# notice im using read.csv2, which expects a EU type of table. If you want to use the NA style,
#then you must switch read.csv2 with read.csv
#You can also use custom delimitrs
sp <- read.csv2("./Occurrences/Rhinolophus_euryale_csv2.csv",header=T) #load csv of occurrence
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")


#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(stk.present$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
```



Getting climate variables ready

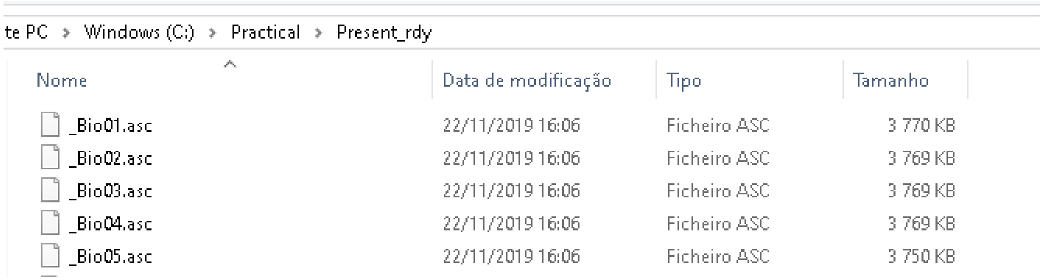
- Clipping the rasters:

```
#cropping the present data
stk.present.AOI.crop <- crop(stk.present,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(stk.present$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
```



- The crop command clips all rasters

```
#now we can save them to another folder in a format
#that maxent can read
#saving the cropped present data in .asc format
writeRaster(stk.present.AOI.crop,
            "./Present_AOI/.asc",
            overwrite=T,
            bylayer=T,
            suffix="names")
```



With what we have we could go to maxent, but first we need to check for autocorrelation effects in our data

Getting climate variables ready

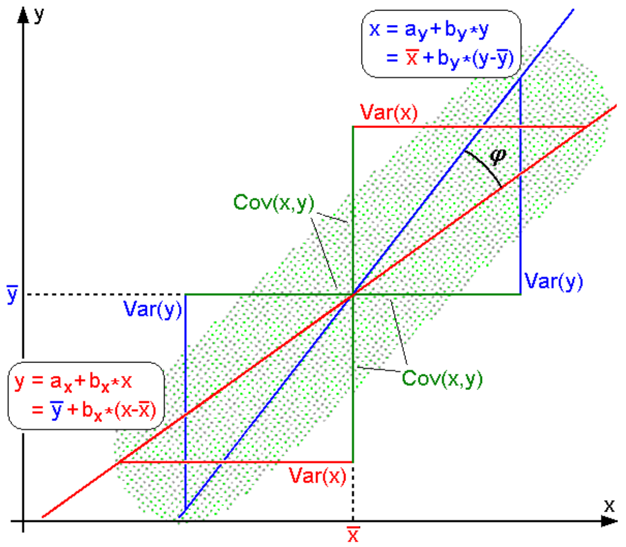
- Now we repeat the same with the present data (world, cropped by future scenarios – antartica is not there in the future data
  - And then you save it to the proper folder (check the script!)

```

#cropping the future data
stk.fut.26.AOI.crop <- crop(stk.fut.26,bbox)
stk.fut.45.AOI.crop <- crop(stk.fut.45,bbox)
stk.fut.60.AOI.crop <- crop(stk.fut.60,bbox)
stk.fut.85.AOI.crop <- crop(stk.fut.85,bbox)

#you can see the
par(mfrow=c(1,2))
plot(stk.fut.26$Bio01)
plot(stk.present$Bio01)

#cropping the present world data
stk.present <- crop(stk.present,stk.fut.26)
```

Getting climate variables ready

- Remember autocorrelation and multicollinearity

- Pairwise correlation



- Model multicollinearity
  - Select n variables – X1… Xn
  - Sucessively make a linear model:
    - X1 ~ X2…Xn
    - X2 ~ X1 + X3… Xn
    - IF any VIF > 10, remove said variable
  - Repeat above step with new model, excluding variable X that was removed.
  - Don't worry, R has a package for it

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

Getting climate variables ready

- Pairwise correlation test
    - Pearson coefficient

```
### the autocorrelation testing is important ONLY for the areas where
### model is trained, so, for this section, we use only the cropped e

### pairwise testing
#first we convert the cropped raster to a data.frame
stk.present.AOI.crop <- na.omit(as.data.frame(stk.present.AOI.crop))
#now this stores the pearson correlation in a matrix
cor.tab <-cor(stk.present.AOI.crop)
#remember to change to write.csv if needed
write.csv2(cor.tab,"CorrelationTable_AOI.csv")
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	Bio01	Bio02	Bio03	Bio04	Bio05	Bio06	Bio07	Bio08	Bio09	Bio10	Bio11	Bio12	Bio13	Bio14	Bio15	Bio16	Bio17	Bio18	Bio19
Bio01	1	0,686055	0,660273	-0,23315	0,875827	0,820743	0,071517	0,025992	0,844474	0,891814	0,913634	-0,56364	-0,44576	-0,63376	0,685389	-0,45453	-0,63032	-0,68202	-0,26329
Bio02	0,686055	1	0,659006	0,111116	0,81381	0,293616	0,509326	-0,10757	0,650355	0,730932	0,506533	-0,60923	-0,51346	-0,63433	0,645142	-0,52374	-0,644	-0,64656	-0,37722
Bio03	0,660273	0,659006	1	-0,64641	0,409402	0,724167	-0,28811	-0,22129	0,608793	0,356148	0,803072	-0,15062	-0,07727	-0,28211	0,561358	-0,07839	-0,26755	-0,35339	0,075049
Bio04	-0,23315	0,111116	-0,64641	1	0,243434	-0,71377	0,909285	0,18432	-0,18764	0,228713	-0,60605	-0,40359	-0,41508	-0,25636	-0,14273	-0,42122	-0,28258	-0,19758	-0,44821
Bio05	0,875827	0,81381	0,409402	0,243434	1	0,452432	0,539321	0,068535	0,764088	0,989195	0,614411	-0,75876	-0,64563	-0,76733	0,639536	-0,65728	-0,77509	-0,78853	-0,47539
Bio06	0,820743	0,293616	0,724167	-0,71377	0,452432	1	-0,50698	-0,05363	0,682023	0,494753	0,971028	-0,14168	-0,05604	-0,27523	0,507805	-0,05796	-0,25494	-0,34915	0,090673
Bio07	0,071517	0,509326	-0,28811	0,909285	0,539321	-0,50698	1	0,116884	0,094506	0,488915	-0,32306	-0,59959	-0,57111	-0,48176	0,13863	-0,58056	-0,50842	-0,43246	-0,5451
Bio08	0,025992	-0,10757	-0,22129	0,18432	0,068535	-0,05363	0,116884	1	-0,28637	0,10579	-0,06057	-0,15776	-0,19362	0,020961	-0,22475	-0,19881	0,004904	0,186607	-0,36424
Bio09	0,844474	0,650355	0,608793	-0,18764	0,764088	0,682023	0,094506	-0,28637	1	0,76456	0,774035	-0,48841	-0,36614	-0,64124	0,672729	-0,37387	-0,62398	-0,75172	-0,12991
Bio10	0,891814	0,730932	0,356148	0,228713	0,989195	0,494753	0,488915	0,10579	0,76456	1	0,635333	-0,74352	-0,63235	-0,75114	0,608282	-0,64363	-0,75762	-0,77618	-0,46092
Bio11	0,913634	0,506533	0,803072	-0,60605	0,614411	0,971028	-0,32306	-0,06057	0,774035	0,635333	1	-0,28708	-0,18652	-0,41157	0,611413	-0,19089	-0,39586	-0,47981	-0,0184
Bio12	-0,56364	-0,60923	-0,15062	-0,40359	-0,75876	-0,14168	-0,59959	-0,15776	-0,48841	-0,74352	-0,28708	1	0,941355	0,843861	-0,42529	0,95245	0,871175	0,804919	0,844799
Bio13	-0,44576	-0,51346	-0,07727	-0,41508	-0,64563	-0,05604	-0,57111	-0,19362	-0,36614	-0,63235	-0,18652	0,941355	1	0,651168	-0,17839	0,996067	0,683382	0,667598	0,896504
Bio14	-0,63376	-0,63433	-0,28211	-0,25636	-0,76733	-0,27523	-0,48176	0,020961	-0,64124	-0,75114	-0,41157	0,843861	0,651168	1	-0,67518	0,668518	0,992771	0,909616	0,511984
Bio15	0,685389	0,645142	0,561358	-0,14273	0,639536	0,507805	0,13863	-0,22475	0,672729	0,608282	0,611413	-0,42529	-0,17839	-0,67518	1	-0,19707	-0,67729	-0,61078	-0,07466
Bio16	-0,45453	-0,52374	-0,07839	-0,42122	-0,65728	-0,05796	-0,58056	-0,19881	-0,37387	-0,64363	-0,19089	0,95245	0,996067	0,668518	-0,19707	1	0,699409	0,680928	0,903838
Bio17	-0,63032	-0,644	-0,26755	-0,28258	-0,77509	-0,25494	-0,50842	0,004904	-0,62398	-0,75762	-0,39586	0,871175	0,683382	0,992771	-0,67729	0,699409	1	0,903737	0,550295
Bio18	-0,68202	-0,64656	-0,35339	-0,19758	-0,78853	-0,34915	-0,43246	0,186607	-0,75172	-0,77618	-0,47981	0,804919	0,667598	0,909616	-0,61078	0,680928	0,903737	1	0,399641
Bio19	-0,26329	-0,37722	0,075049	-0,44821	-0,47539	0,090673	-0,5451	-0,36424	-0,12991	-0,46092	-0,0184	0,844799	0,896504	0,511984	-0,07466	0,903838	0,550295	0,399641	1

- Think of which variables are good for your species
- Confirm if there is more than 0.7 and smaller than -.07 pairwise correlation between them
- To many pairwise correlations will cause multicollinearity

	Var 1	Var 2	Va3
Var 1	Var 1 ~ var 1	.....	...
Var 2	Var2 ~var 1	Var 2 ~ var 2	....
Var3	Var 3 ~ Var 1	Var 3 ~ var 2	.....

Getting climate variables ready

- For exemple, my first selection:

	Bio01	Bio04	Bio07	Bio12	Bio15	Bio19
Bio01	1	-0,23315	0,071517	-0,56364	0,685389	-0,26329
Bio04	-0,23315	1	0,909285	-0,40359	-0,14273	-0,44821
Bio07	0,071517	0,909285	1	-0,59959	0,13863	-0,5451
Bio12	-0,56364	-0,40359	-0,59959	1	-0,42529	0,844799
Bio15	0,685389	-0,14273	0,13863	-0,42529	1	-0,07466
Bio19	-0,26329	-0,44821	-0,5451	0,844799	-0,07466	1

- Now I need to test for multicollinearity:

```
#multicollinearity testing
library(usdm)

#e.g. i select Bio01; Bio04; Bio07; Bio 12; Bio 15 and bio 19
head(df.crop.stack)
df.crop.stack.selection <- df.crop.stack[,c(1,4,7,12,15,19)] #select only the variables i am interested
head(df.crop.stack.selection)

vif(df.crop.stack.selection, maxobservations=nrow(df.crop.stack.selection))
```

	Variables	VIF
1	Bio01	3.623841
2	Bio04	13.534628
3	Bio07	13.308972
4	Bio12	9.459776
5	Bio15	2.617598
6	Bio19	5.691570

	Variables	VIF
1	Bio01	3.285936
2	Bio04	1.934874
3	Bio12	9.448515
4	Bio15	2.331450
5	Bio19	5.659382

- Bio 04 and 07 are so strongly correlated that they affect the entire model – I have to remove one
  - Bio 04: Temp. seasonality
  - Bio 07: Temp. anual range
  - I like seasonality more for Mediterranean climates

Last step before modelling

- Now we have:
  - Selected the occurrence data we want to use
  - Selected the variables we want to use and cropped them to our choosen study area

  - What's missing?
    - Crop the future scenario data and export it to a folder

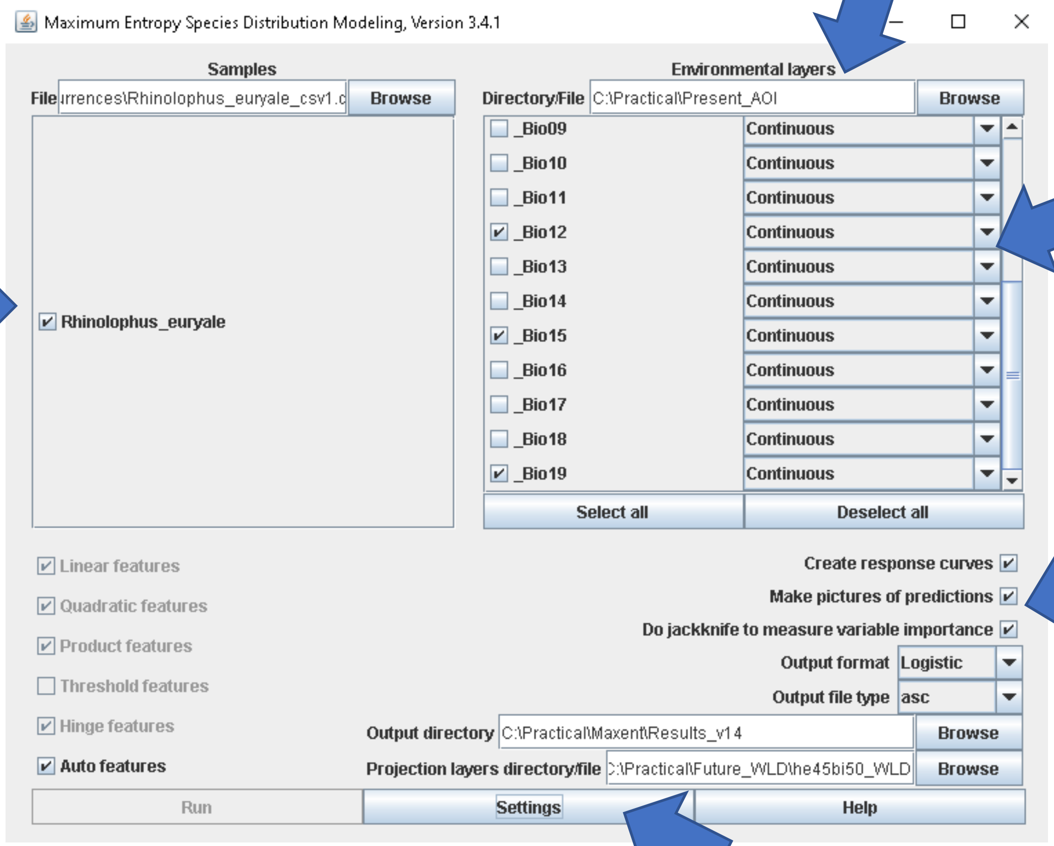
```r
#Saving the cropped variables to the correct folder, in the .asc format

#we had already saved the cropped rast
path2rst.HadGEM2ES_RCP85 <- list.files("./Future/", full.names = T)
path2rst.HadGEM2ES_RCP85
#in this case the order of variables is changed, its better to keep everything in the
#same order. It's easy to adapt the code to read in our preferred order
#WARNING: this step might be different in your case, CHECK IT FIRST
rst.fut.stack <- stack(path2rst.HadGEM2ES_RCP85[c(1,12:19,2:11)])
rst.fut.stack.crop <- crop(rst.fut.stack,bbox)
names(rst.fut.stack.crop)

#now we can jus save the variables to a new folder
#but first, we rename the variables to the same as the present vars
names(rst.fut.stack.crop)<- names(rst.AOI.crop)

writeRaster(rst.fut.stack.crop,
            "./Future_rdy/.asc",
            overwrite=T,
            bylayer=T,
            suffix="names")

#now you can go to maxent
```

Think:

- If you train and project the model in one AOI you can't find new areas around the world where the species might be
- If you crop your future data, you will also not be able to predict anything regarding the world

Maxent finally

Load the enviromental variables – point to the folder



Load the occurrences (beware of the type of CSV)

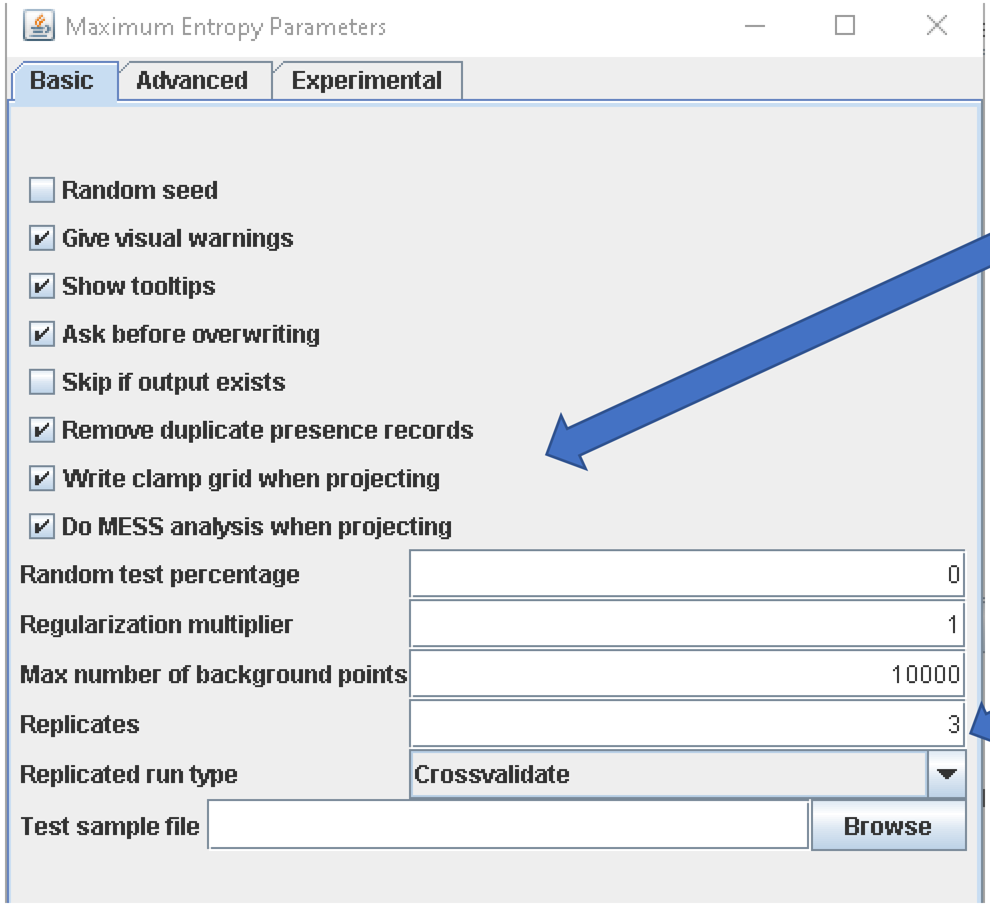
De-select all and then select your variables

Activate everything and choose logistic output

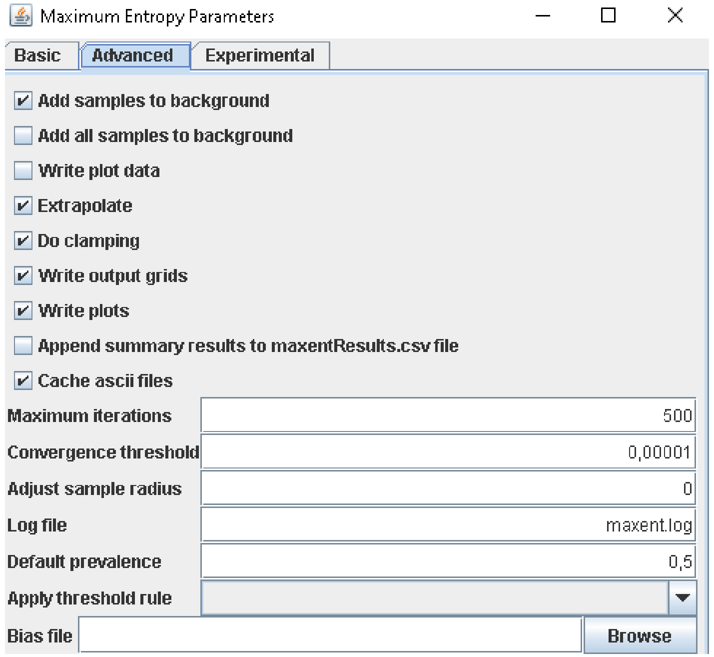
Press settings for extra options

Take note of the warnings you will get and let me know what you think they are x)

Maxent finally



Make sure options are like these – What is "MESS"?



This is the number of repetitions of the models, in "real" research you would have to do thousands.

Good luck!