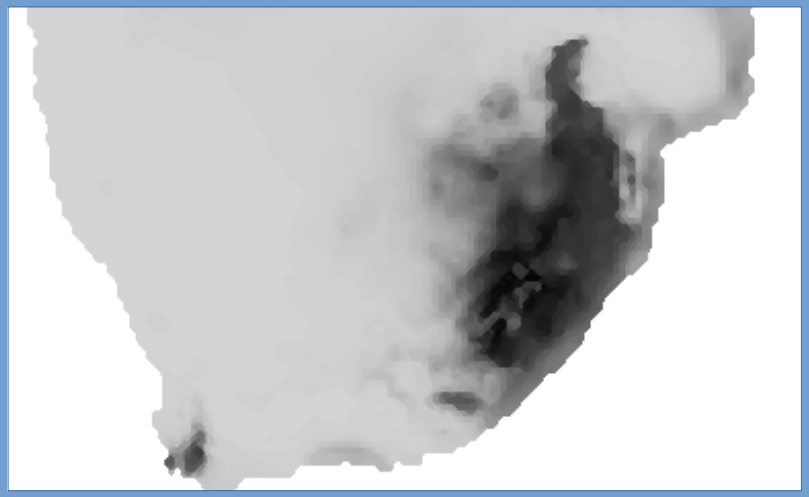
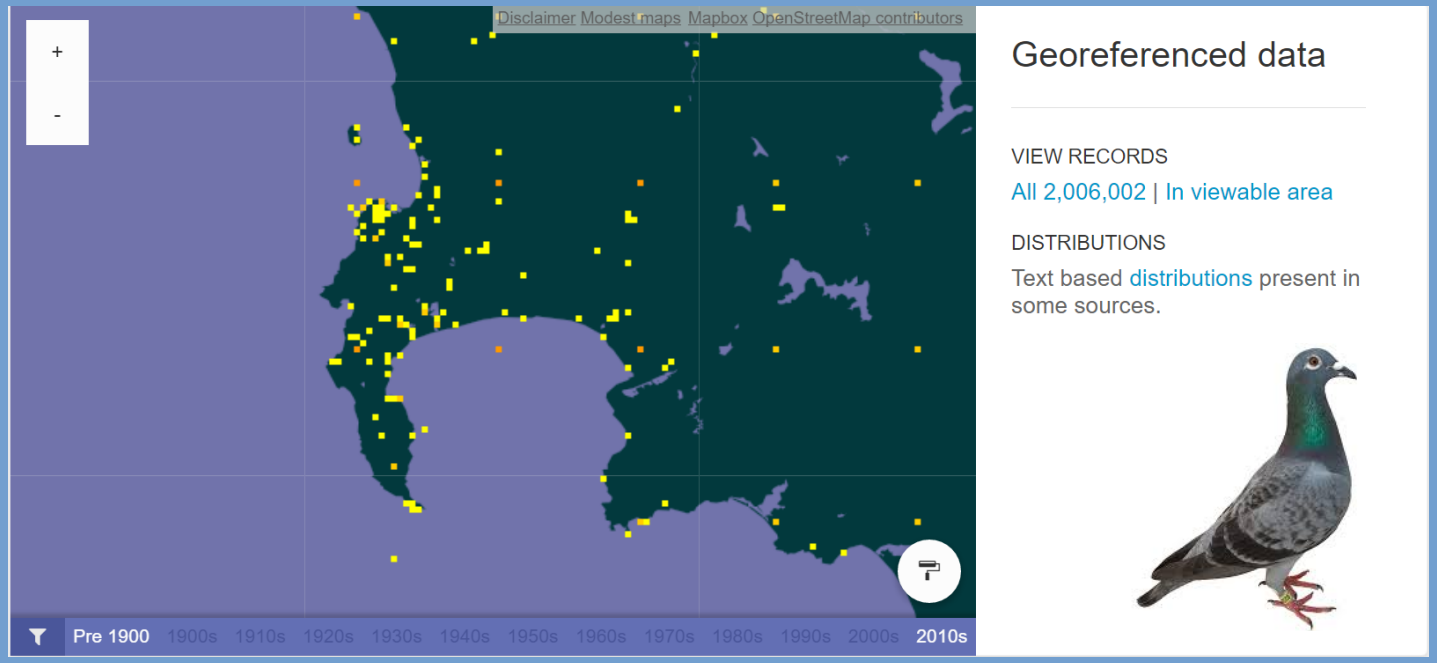
SEEC Stats Toolbox

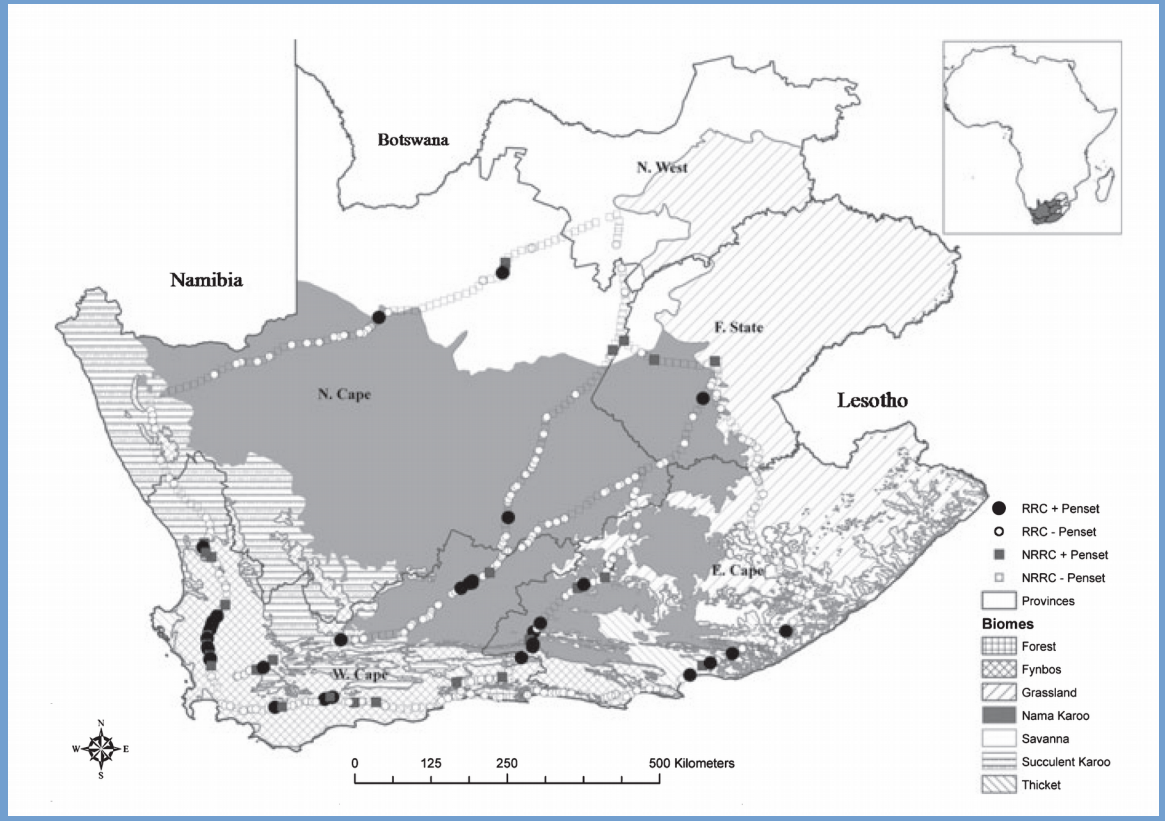
Species distribution modelling in R



Our imperfect knowledge of the natural world

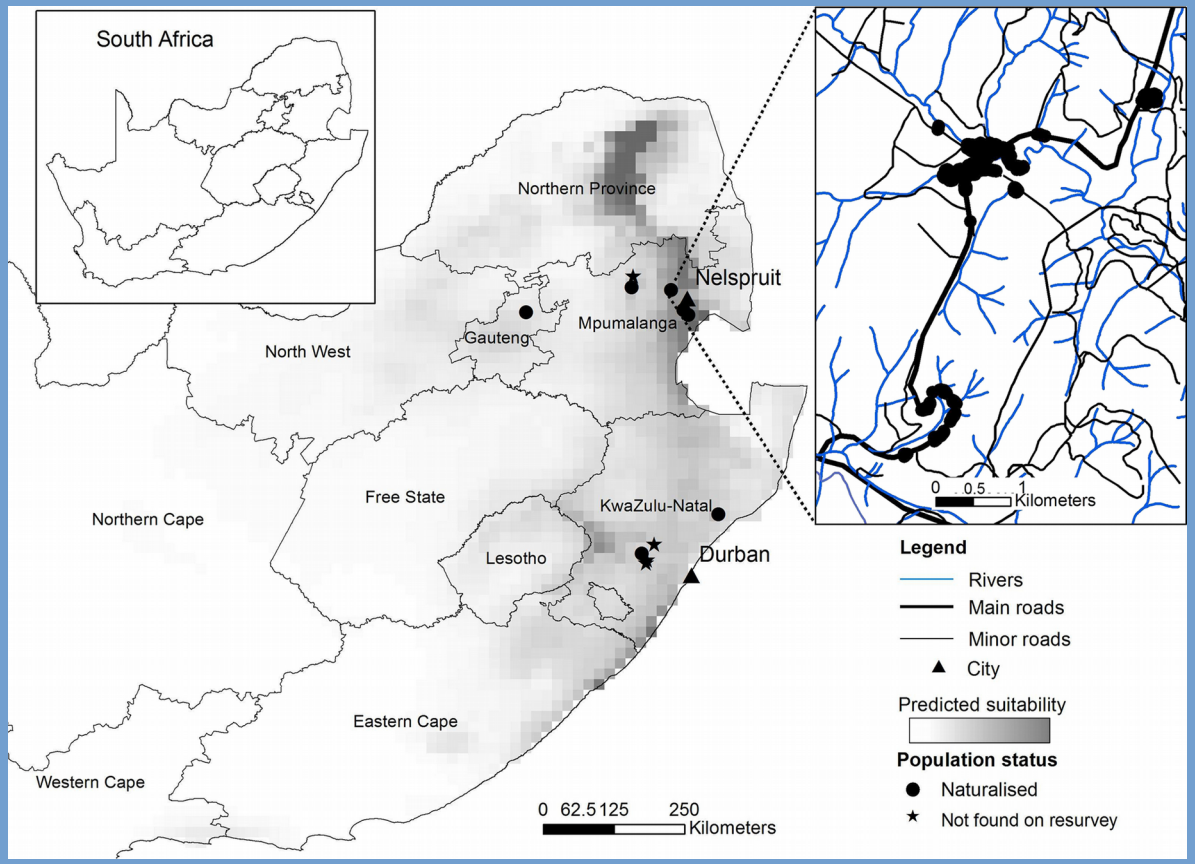


Our STILL imperfect knowledge of the natural world



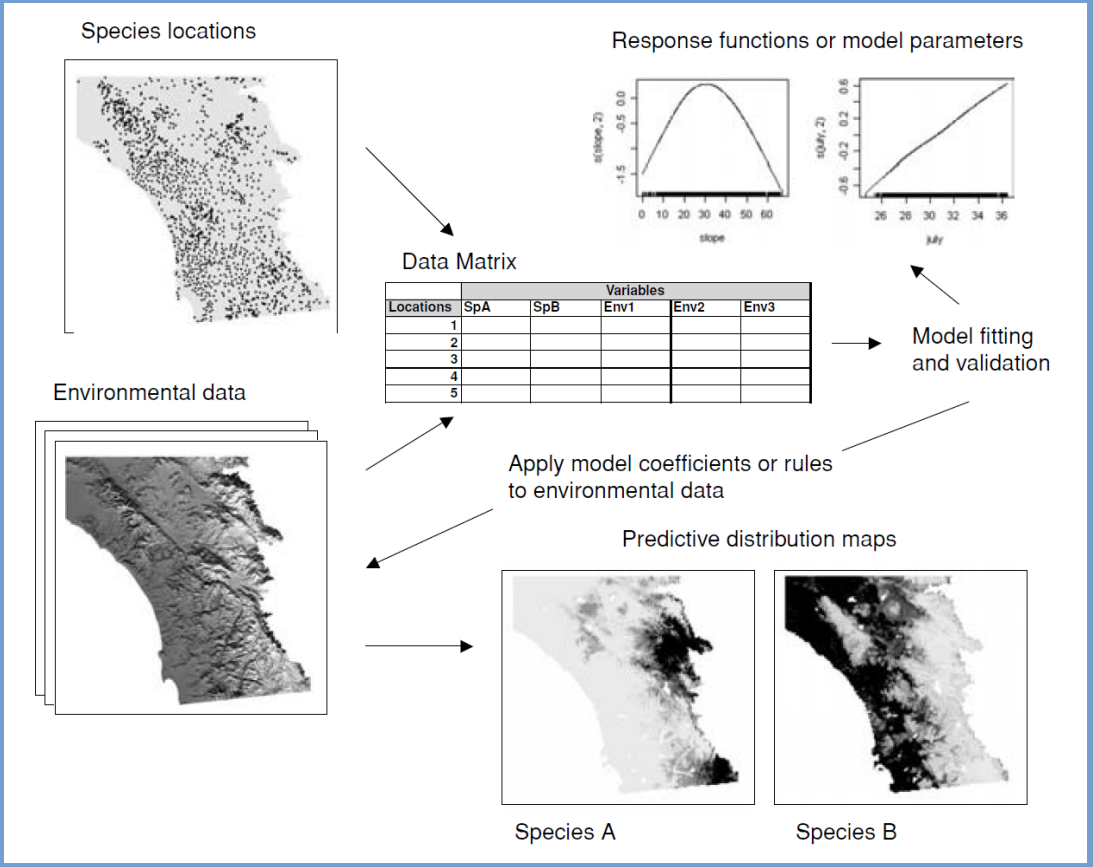
Rahlao et al. (2010). Weed Research 50: 537

What we actually want



Geerts et al. (2016) Biological Invasions. DOI 10.1007/s10530-016-1226-y

What is a species distribution model (SDM)?



Franklin (2009). Mapping species distributions. Cambridge University Press

Uses of SDMs

- Conservation prioritization
- Climate change predictions
- Rare species detection
- Invasive species screening
- Generating hypotheses - correlates of distribution
- Habitat suitability

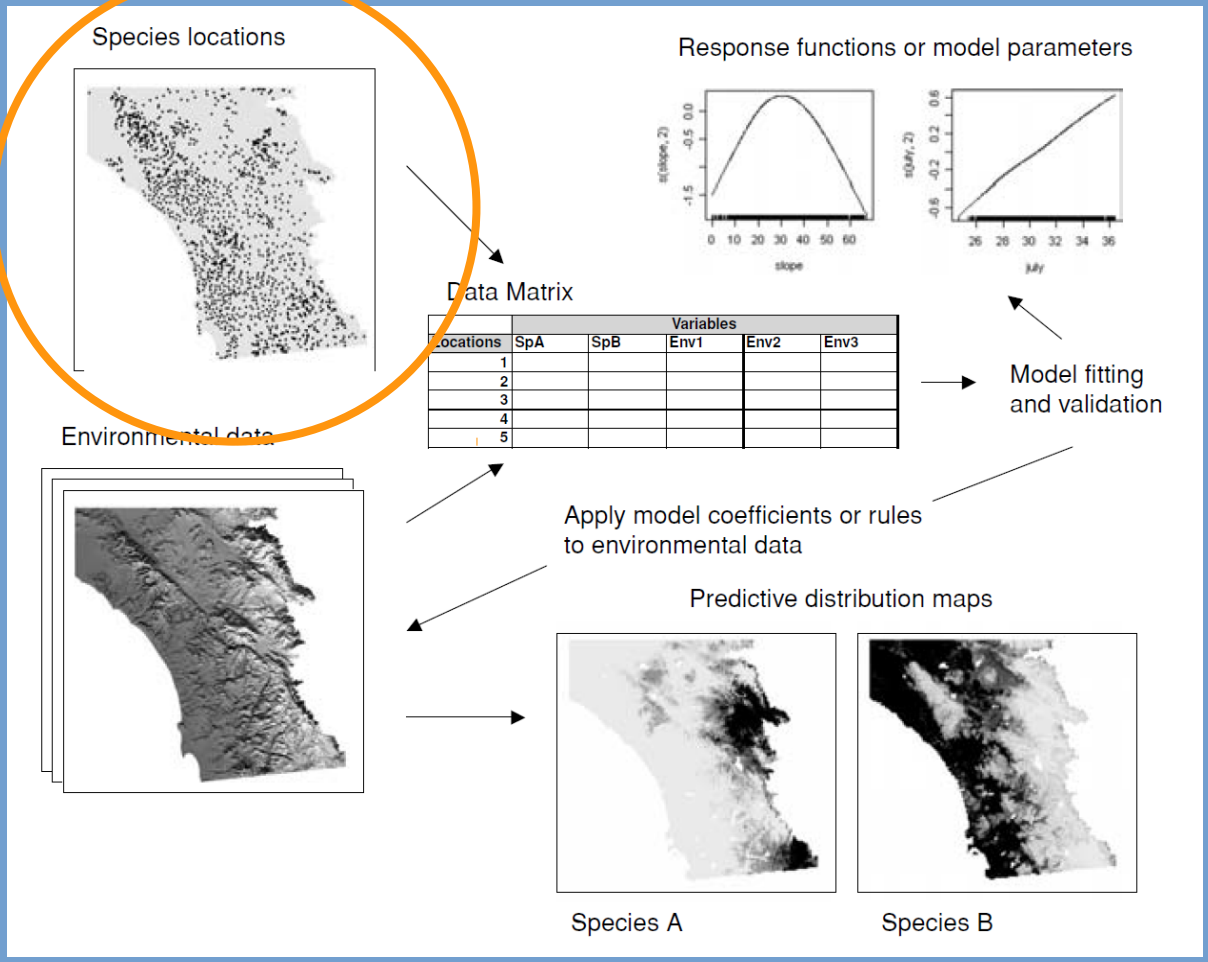
Eight steps to your own SDM

1. Occurrence data
2. Environmental data
3. Background samples
4. Study extent
5. Data cleaning
6. Modelling
7. Checking your model
8. Projecting your model

Southern Bald Ibis
Geronticus calvus (Boddaert, 1783)

www.hbw.com

Step 1: Occurrence data



Step 1: Occurrence data

- Sources:
  - Your own data
  - gbif.org/species
  - newposa.sanbi.org

- Useful to keep ID, locality, date... for data cleaning purposes

Step 1: Occurrence data

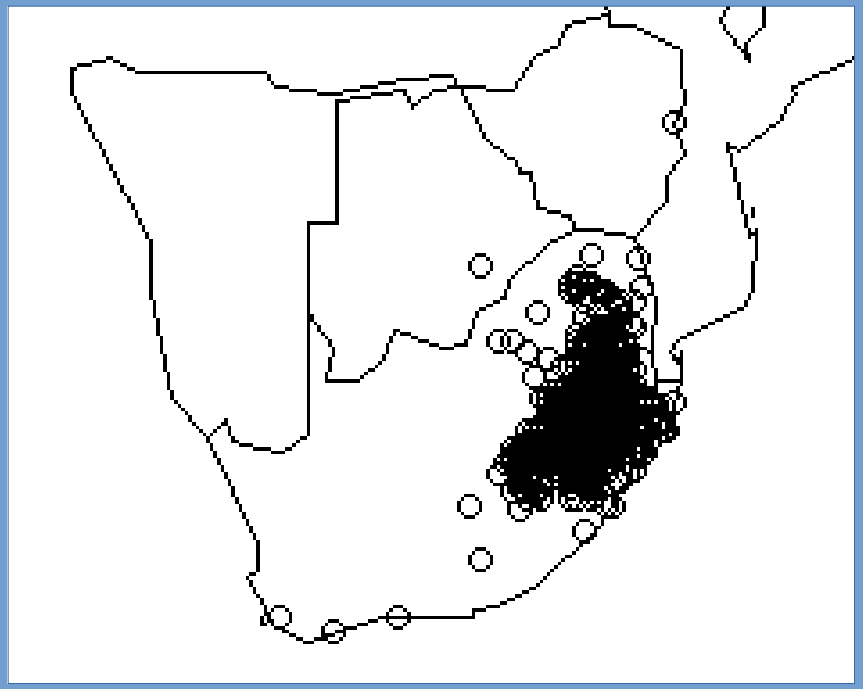
RStudio Source Editor

Stats_Toolbox_SDMs.R*

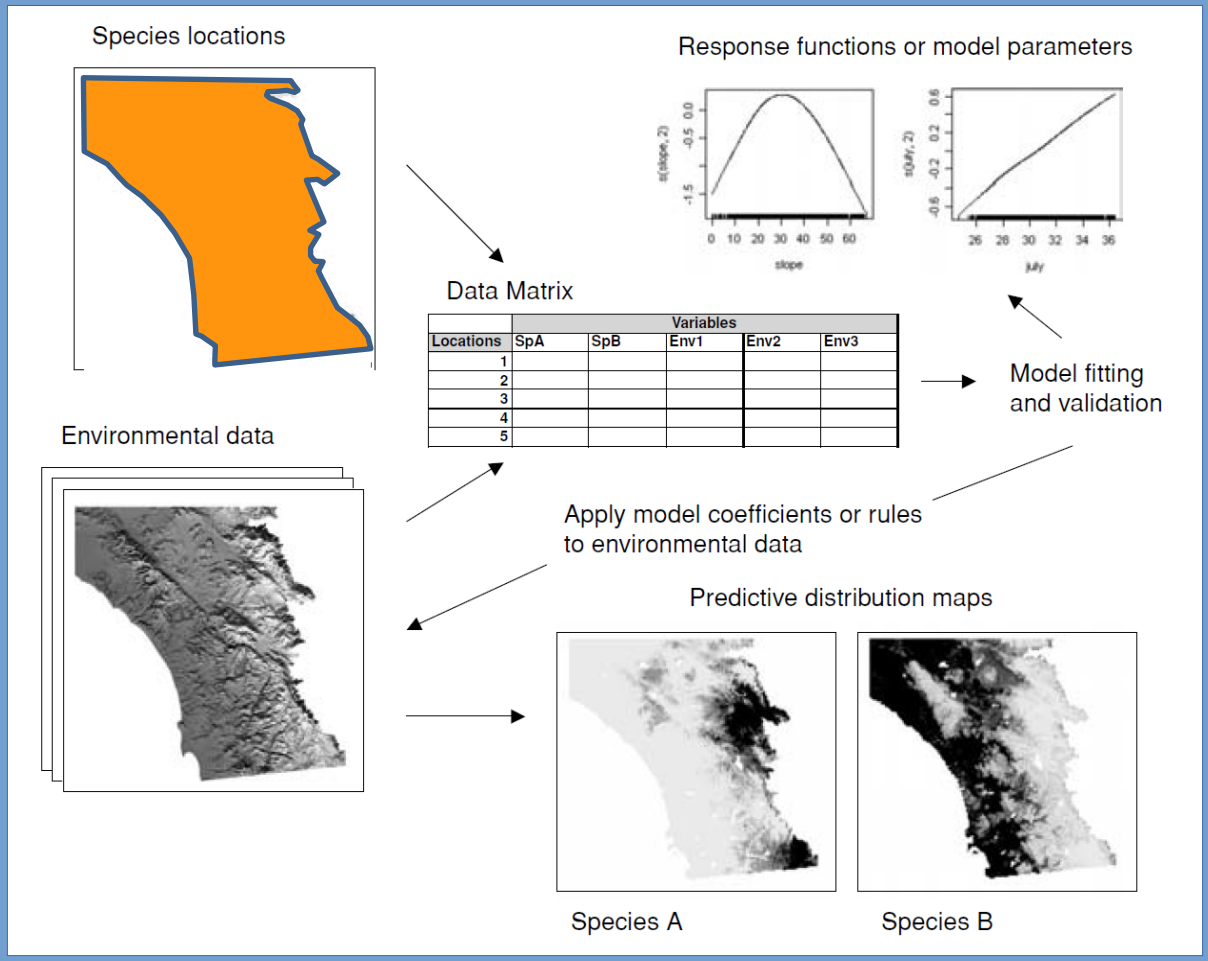
Source on Save | Run | Source

```
13
14  #-----------------------------------------------------------------------------------------------------------
15  #Get presence and background data:
16  #-----------------------------------------------------------------------------------------------------------
17  #We are going to get our data using the package rgbif. This allows us to download directly from GBIF.
18  library(rgbif)
19  #Search for the GBIF key for the species in question. This is a unique identifier that tells GBIF what data we want:
20  (key = name_suggest(q='Geronticus calvus', rank='species'))
21  key = key$key
22  #See how many records in GBIF are available. We only want records that have coordinates so we say "georeferenced=T":
23  occ_count(taxonKey = key, georeferenced=T)
24  #Download data from GBIF. Keep only records with coordinates:
25  presGBIF = occ_search(taxonKey=key, limit=10000, hasCoordinate=T)
26  presGBIF
27  #Keep only dataframe part of GBIF data:
28  pres = presGBIF$data
29  names(pres)
30  #Keep only some data fields:
31  pres = pres[,c('key','scientificName','decimalLongitude','decimalLatitude','basisOfRecord','year','countryCode','locality')]
32  #To save time, save these data to a CSV file so that you can use this instead if you run this script at a later stage:
33  write.csv(pres, 'pres.csv', row.names=F)
34
```

Step 1: Occurrence data

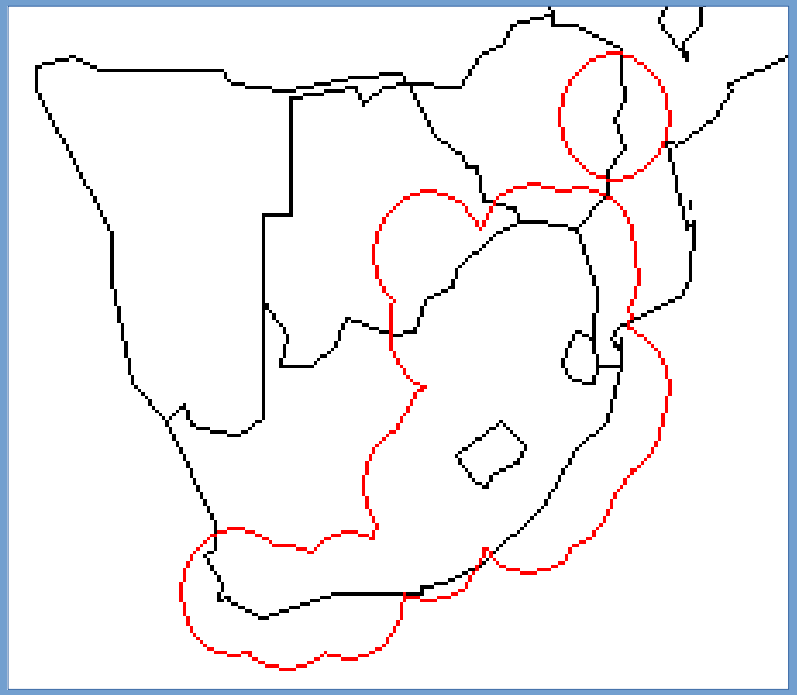


Step 2: Study extent

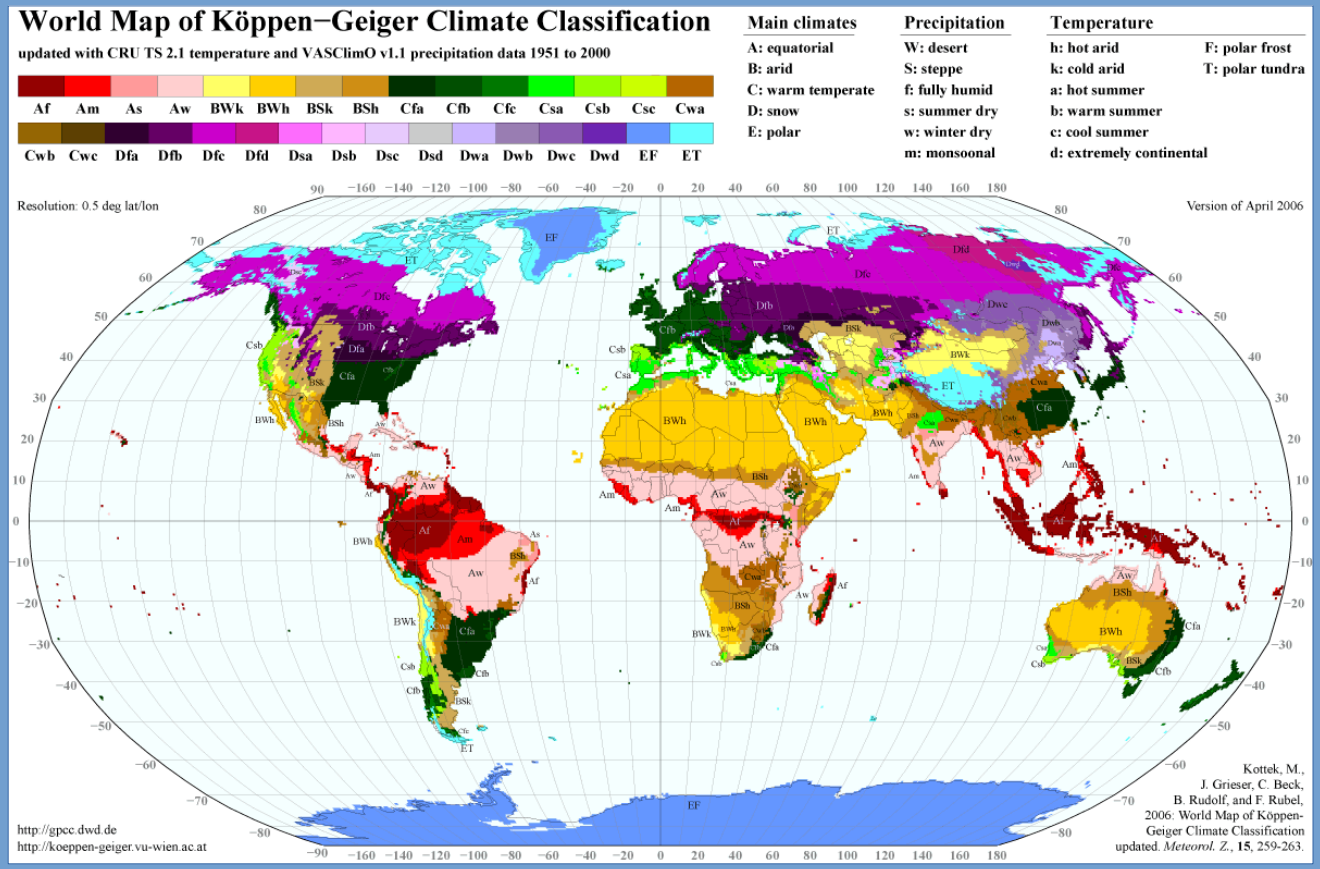


Step 2: Study extent

- Choosing an extent:
  - Size of the study area
  - Area from which background samples are selected



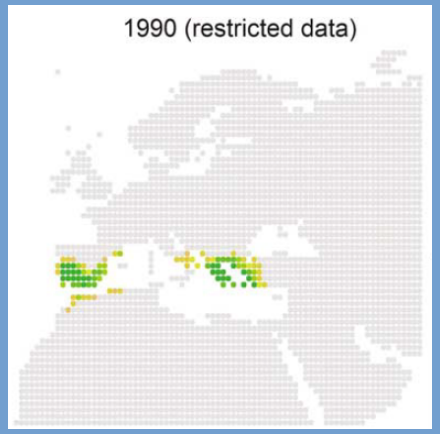
Step 2: Study extent

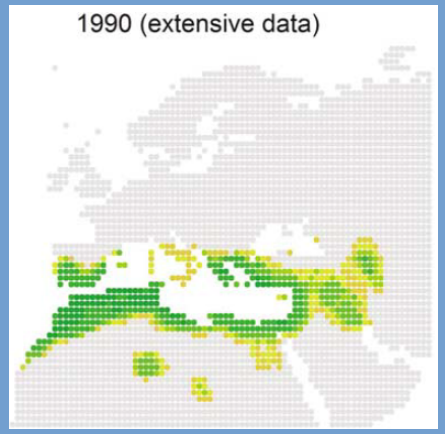


See Webber et al. (2011). Diversity and Distributions 17: 978

Step 2: Study extent

- Smaller extent:
  - Produces a smaller predicted area and perhaps emphasis on incorrect predictor variables.
- Larger extent:
  - Produces a larger predicted area, which may be unrealistic, and climate variables often dominate.





Barbet-Massin et al. 2010. Ecography 33: 878

Step 2: Study extent

- Depends on your objective: occupied or suitable habitats (Merow et al., 2013)
  - Occupied habitat – accessible area.
    - Identify where a species is currently found.
    - Identify environmentally limiting factors.
  - Suitable habitat – area that you wish to contrast with against the presences.
    - Climate change.
    - Invasive species.

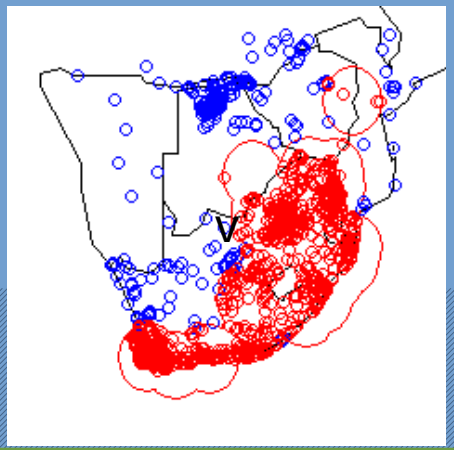
Step 2: Study extent

RStudio Source Editor

Stats_Toolbox_SDMs.R*

Source on Save | Run | Source

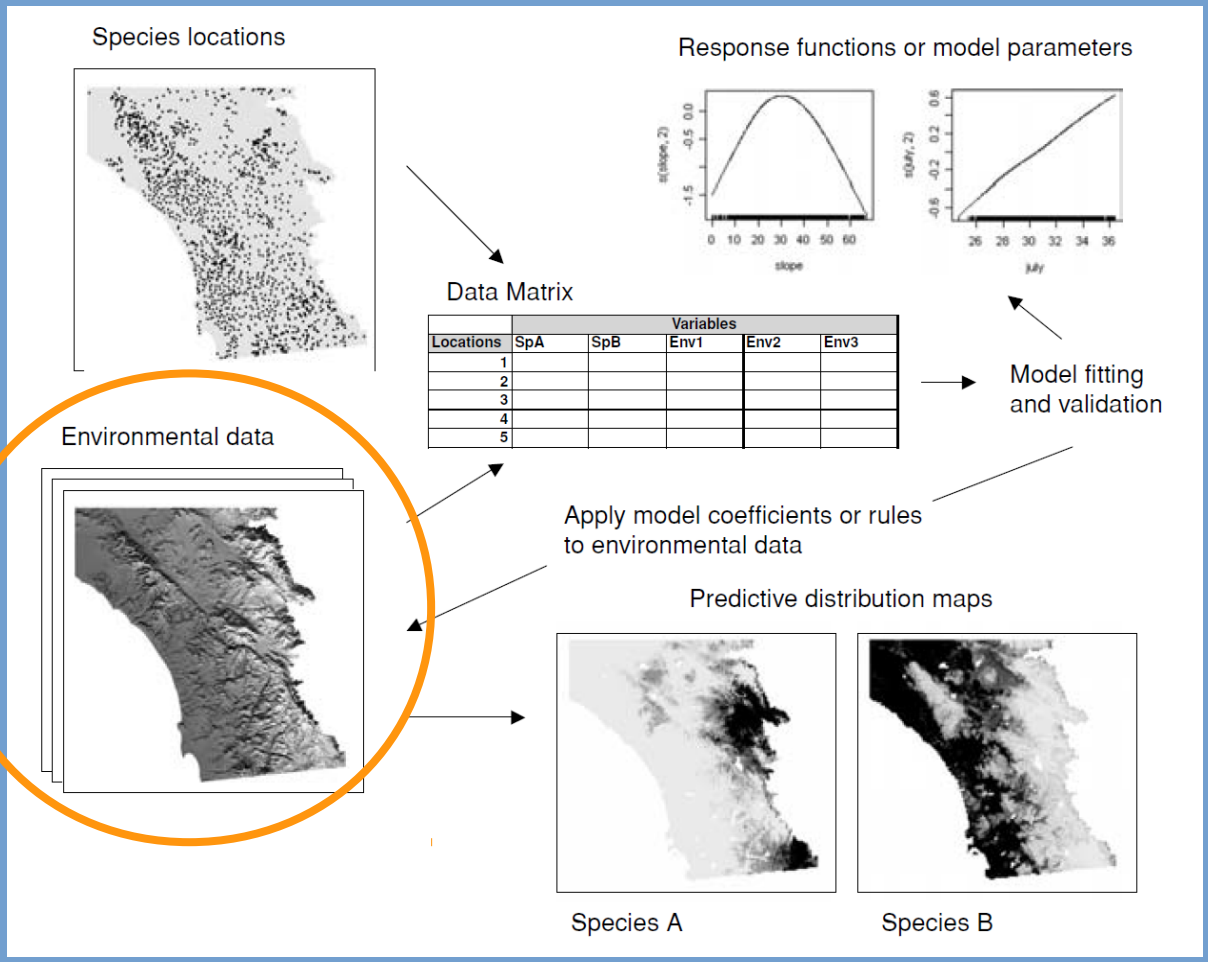
```
185 #We are going to select only pseudo-absences within 2 degrees (~240 km) of presences. You could also select only
186 #records within Koppen-Geiger climate zones that presences occur in, or something similar.
187 library(rgeos)
188 presSp = presClean #Create new dataframe from presences
189 coordinates(presSp) =~ x+y #Transform dataframe into a spatial points object
190 crs(presSp) = '+init=epsg:2052' #Give the spatial points object a projection (in this case Hartebeesthoek94)
191 presBuff  = gBuffer(presSp,width=2) #Create buffer polygon around presences
192 #Plot the buffer area on a map:
193 plot(sa)
194 plot(presBuff, add=T, border='red')
195 points(presSp) #Add presence records
196 #This next part will create a spatial points object from our pseudo-absences, overlay these on our buffer object
197 #and then we select only pseudo-absences falling within the buffer object:
198 backSp = backClean #Create new dataframe from pseudo-absences
199 coordinates(backSp) =~ x+y #Transform dataframe into a spatial points object
200 crs(backSp) = '+init=epsg:2052' #Give the spatial points object a projection (in this case Hartebeesthoek94)
201 overBufferback = over(backSp, presBuff)
202 backClean = backClean[!is.na(overBufferback),]
203 #Add the pseudo-absences to our map:
204 points(backSp, col='blue') #Add pseudo-absences, including those beyond buffer zone
205 points(x=backClean$x, y=backClean$y, col='red') #Add pseudo-absences, excluding those beyond buffer zone
206 points(presSp) #Add presence records
207
```



Step 2: Study extent

- Useful references:
  - Anderson & Raza (2010). Journal of Biogeography 37: 1378
  - Barve et al. (2011) Ecological Modelling 222: 1810
  - Merow et al. (2013). Ecography 36: 1058

Step 3: Environmental data

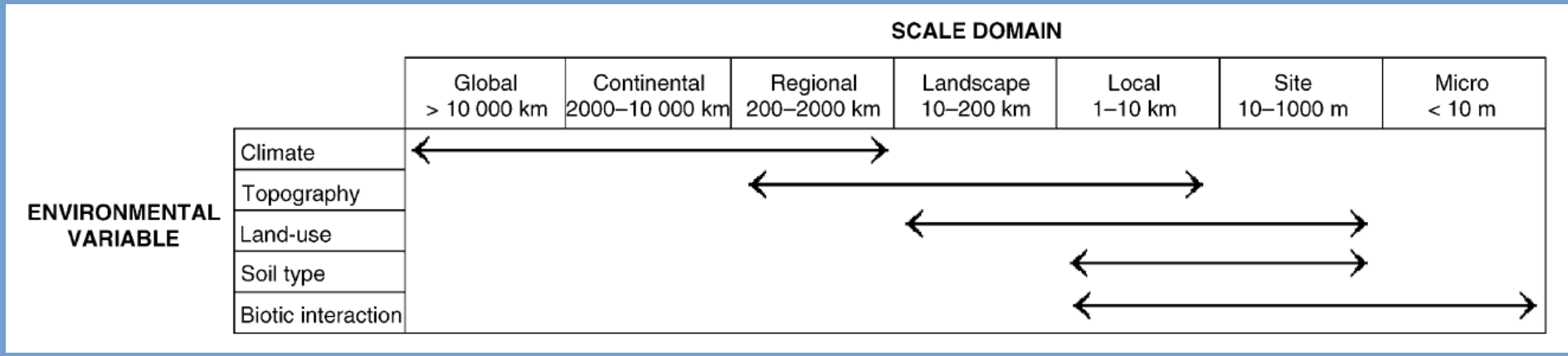


Step 3: Environmental data

- Sources:
  - Bioclim www.worldclim.org/bioclim
    - Can download directly in R using the "getData" function in the raster package
  - BioOracle www.oracle.ugent.be
  - Google is your friend

Step 3: Environmental data

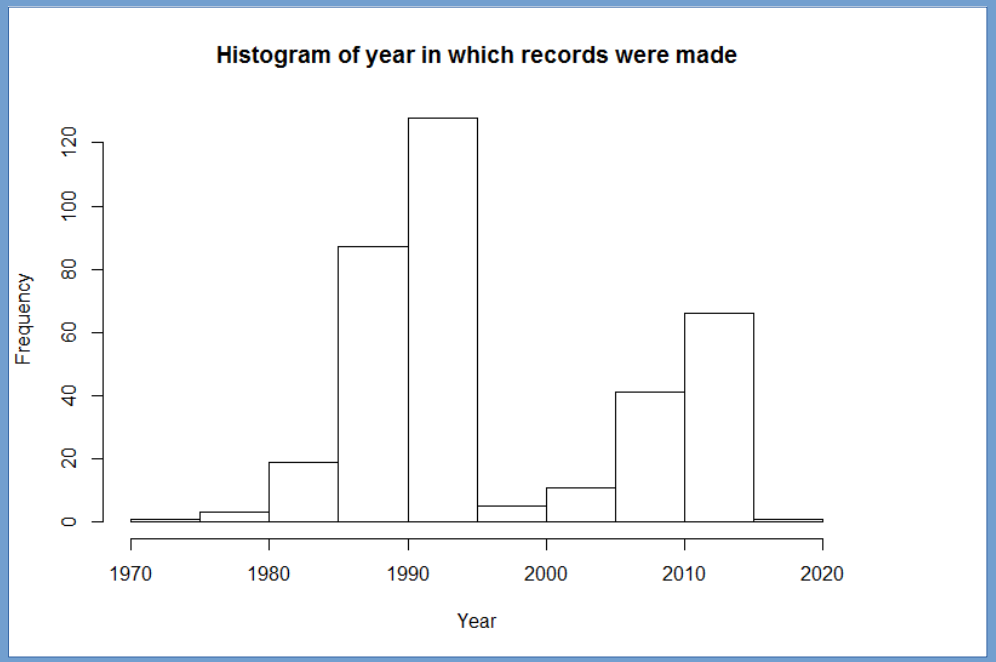
- Considerations:
  - Spatial resolution



Pearson & Dawson (2003). Global Ecology & Biogeography 12: 361

Step 3: Environmental data

- Considerations:
  - Temporal resolution



Step 3: Environmental data

- Considerations:
  - Selecting variables
    - Include variables that directly limit a species (e.g. min. temperature)
    - Include variables that are resources (e.g. nutrients)
    - Include variables that link to physiology (e.g. water availability)

Step 3: Environmental data

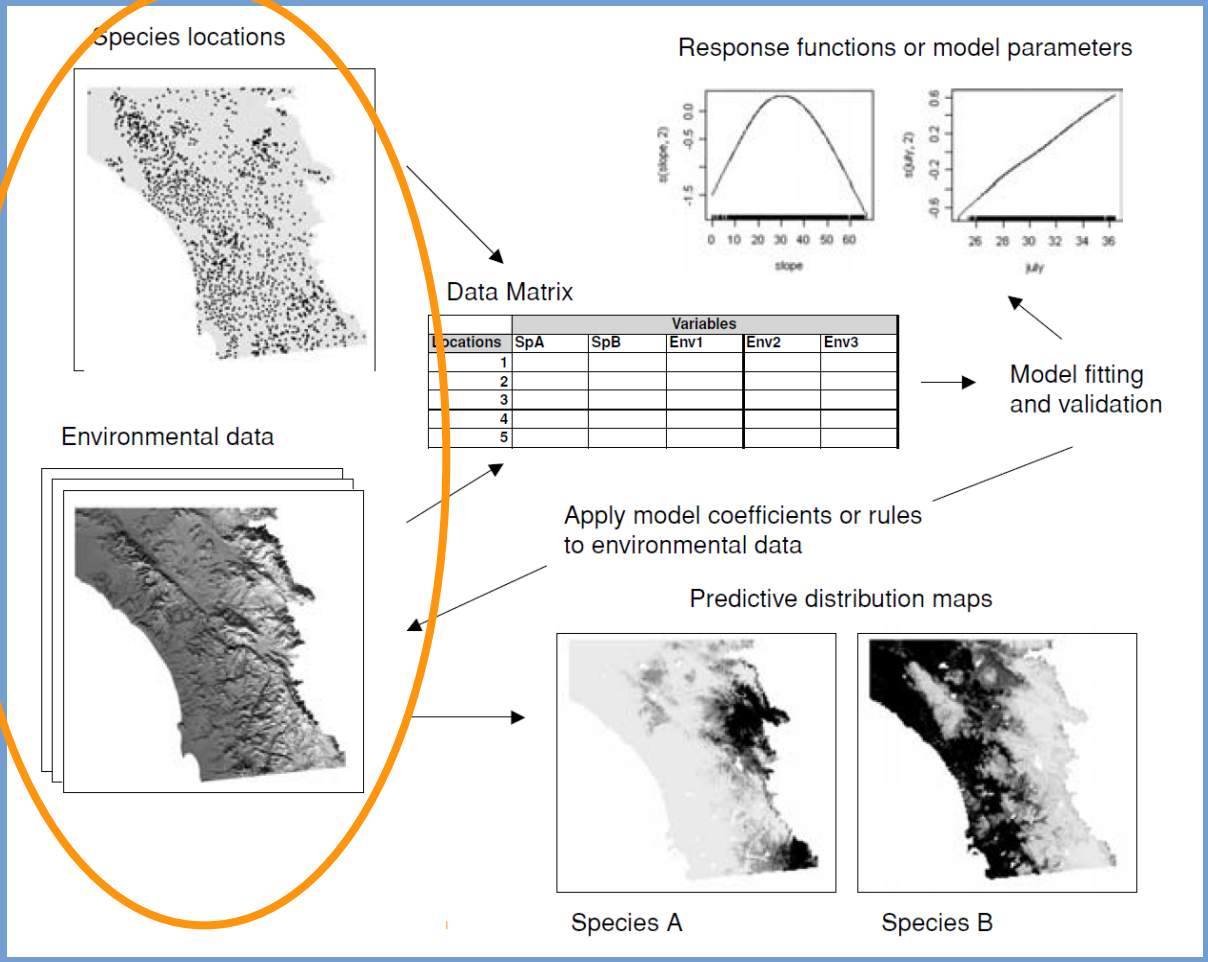
RStudio Source Editor

Stats_Toolbox_SDMs.R*

Source on Save | Run | Source

```
#-----------------------------------------------------------------------------------------------------------
#Get environmental layers:
#-----------------------------------------------------------------------------------------------------------
library(raster)

#One way to get WORLDCLIM data is to use the getData function in raster. We want the Bioclim variables at a
#resolution of 10 minutes
bio = getData(name='worldclim', var='bio', res=10, download=T) #This creates a raster stack with all the Bioclim variables.

#The alternative way of reading in rasters is shown below. You should also use the approach below once you have
#downloaded the Bioclim data using the getData function above. In other words, you can comment out the line
#above with the getData function for future use of this script.
(bioFiles = list.files('wc10/')) #List the files in the wc10 folder that was created using the getData function
(bioFiles = bioFiles[grep('.bil', bioFiles, fixed=T)]) #Select only the .bil files (these are the actual rasters)
(bioFiles = bioFiles[c(1,12:19,2:11)]) #Reorder file names to get into numerical order (BIO1, BIO2...)

bio = stack(paste0(getwd(), '/wc10/', bioFiles), RAT=F)
#Rescale Bioclim variables (BIOs 1 to 11) that have been multiplied by 10 (or 1000 for BIO4) (For example, BIO1, which is
#mean annual temperature, has values from -269 to 314, whereas these should be -26.9 to 31.4 degrees C.
#The *10 format is the default format for Bioclim):
for(r in c(1:3,5:11)){
  bio[[r]] <- bio[[r]]/10
}
bio[[4]] <- bio[[4]]/1000

#-----------------------------------------------------------------------------------------------------------
```

Step 4: Background samples

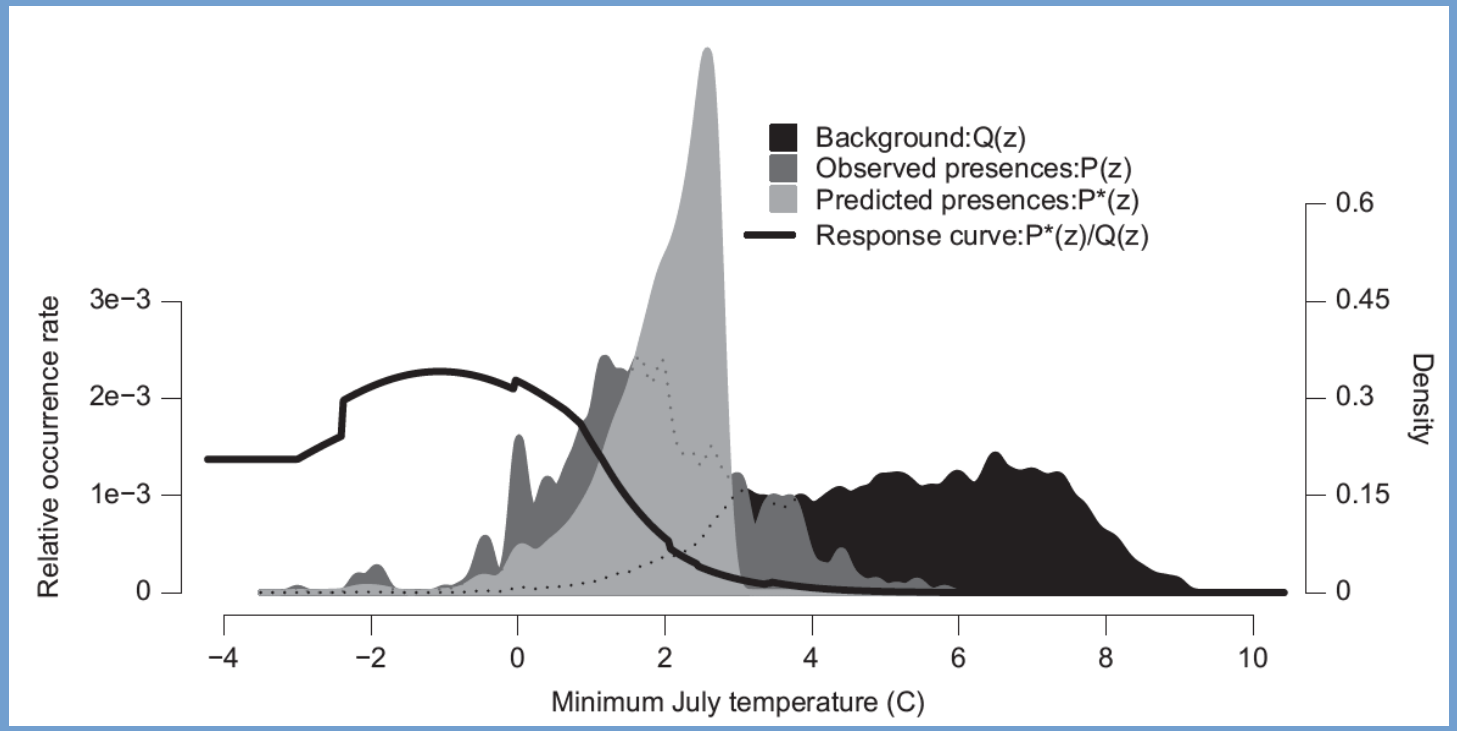


Step 4: Background samples

- When true absences unavailable
- *A priori* considered equally likely to contain individuals of a species

Merow et al. (2013). Ecography 36: 1058

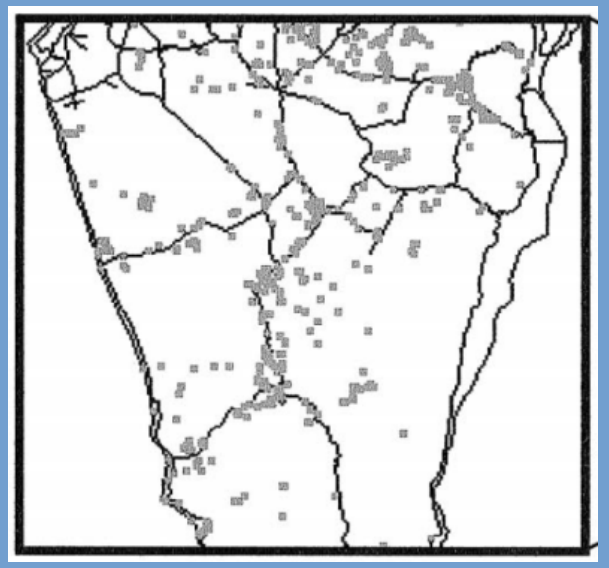
What background samples are used for



Merow et al. (2013). Ecography 36: 1058

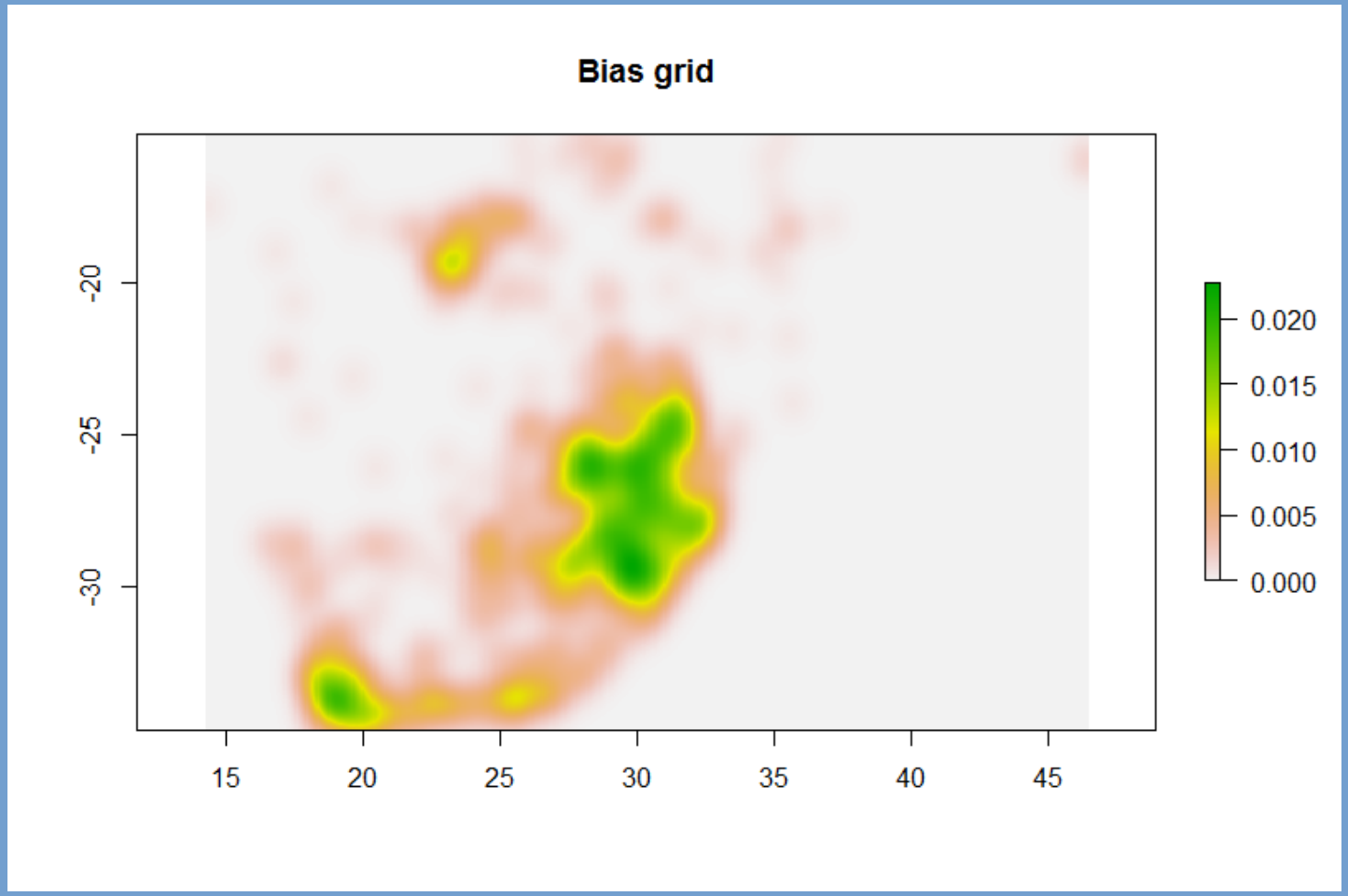
Step 4: Background samples

- Sampling bias
- Target Group Sampling
  - Use coordinates of related species or in same functional group to select background
  - Accounts for sampling bias
  - Create a bias grid for Maxent
- Useful refs:
  - Elith et al. (2010) Methods in Ecology and Evolution 1: 330
  - Merow et al. (2013)
  - Phillips et al. (2009) Ecological Applications 19: 181



Hijmans et al. (2000) Conservation Biology 14: 1755

Step 4: Background samples



Step 4: Background samples

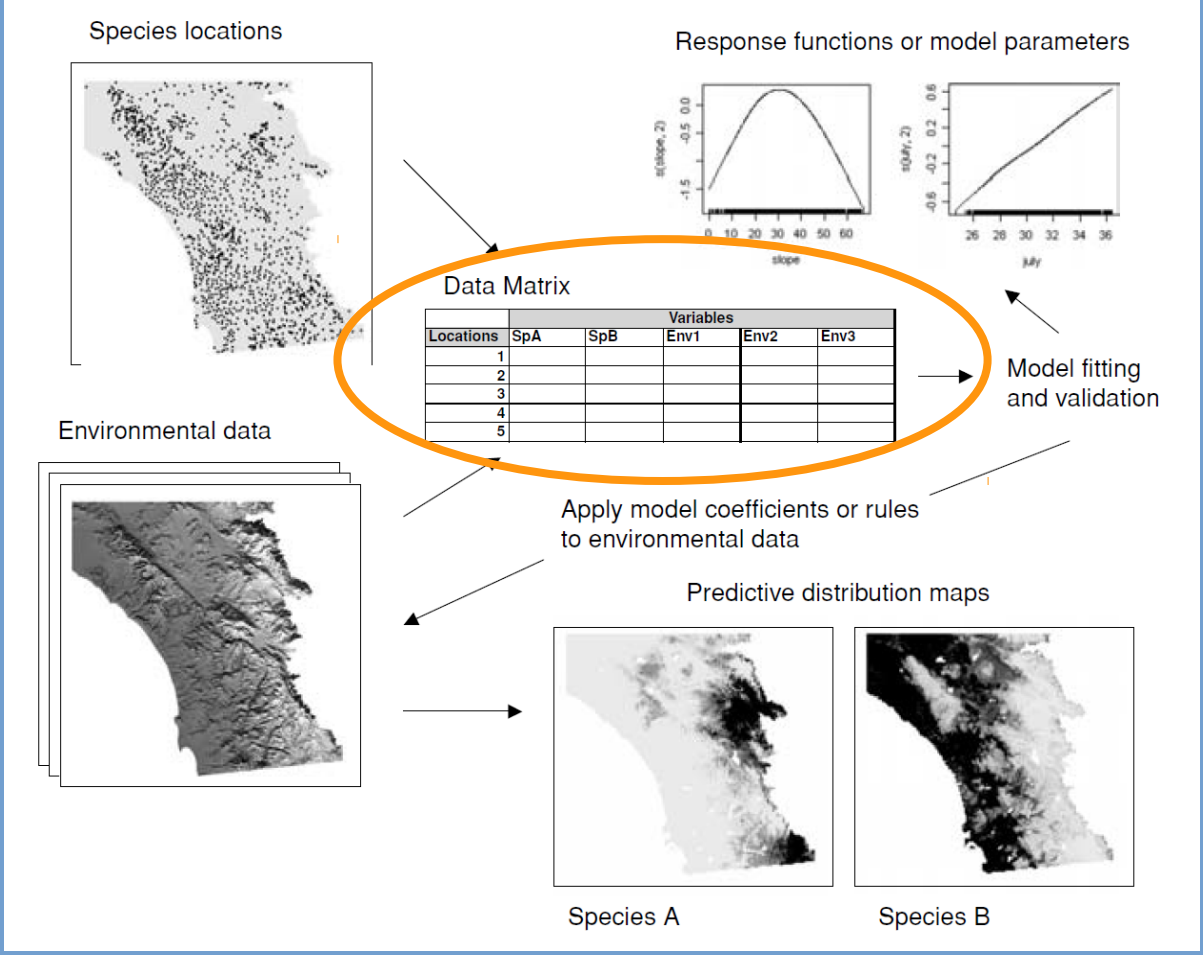
RStudio Source Editor

Stats_Toolbox_SDMs.R*

Source on Save | Run | Source

```
208 #Create a sampling bias grid for Maxent
209 #First we need to create a dataframe of all our observations (presences and pseudo-absences). This will give an indication
210 #of sampling effort:
211 allDat = rbind(presUncleaned, backUncleaned)
212 #Next we rasterize this dataframe:
213 allRas = rasterize(cbind(allDat$x, allDat$y), y=bio[[1]], field=1)
214 plot(allRas, xlim=c(15,35), ylim=c(-35,-15)) #Plot
215 #Get ID (cell number) of cells in which there are presences (of all species):
216 presRasID = which(values(allRas)==1)
217 #Get the coordinates of the above cells:
218 presRasCoords = coordinates(allRas)[presRasID,]
219 #Use a Gaussian kernel to smooth the number of records. Provides a density estimate that will be our bias grid:
220 library(MASS)
221 dens = kde2d(x=presRasCoords[,1], y=presRasCoords[,2], h=c(2,2), n=c(nrow(allRas),ncol(allRas)))
222 #Rasterize the density estimate:
223 biasGrid = raster(dens)
224 plot(biasGrid, main="Bias grid") #Plot
225 #Write this to a raster file:
226 writeRaster(biasGrid, 'biasGrid.tif', overwrite=T)
227
```

Step 5: Data cleaning



Step 5: Data cleaning

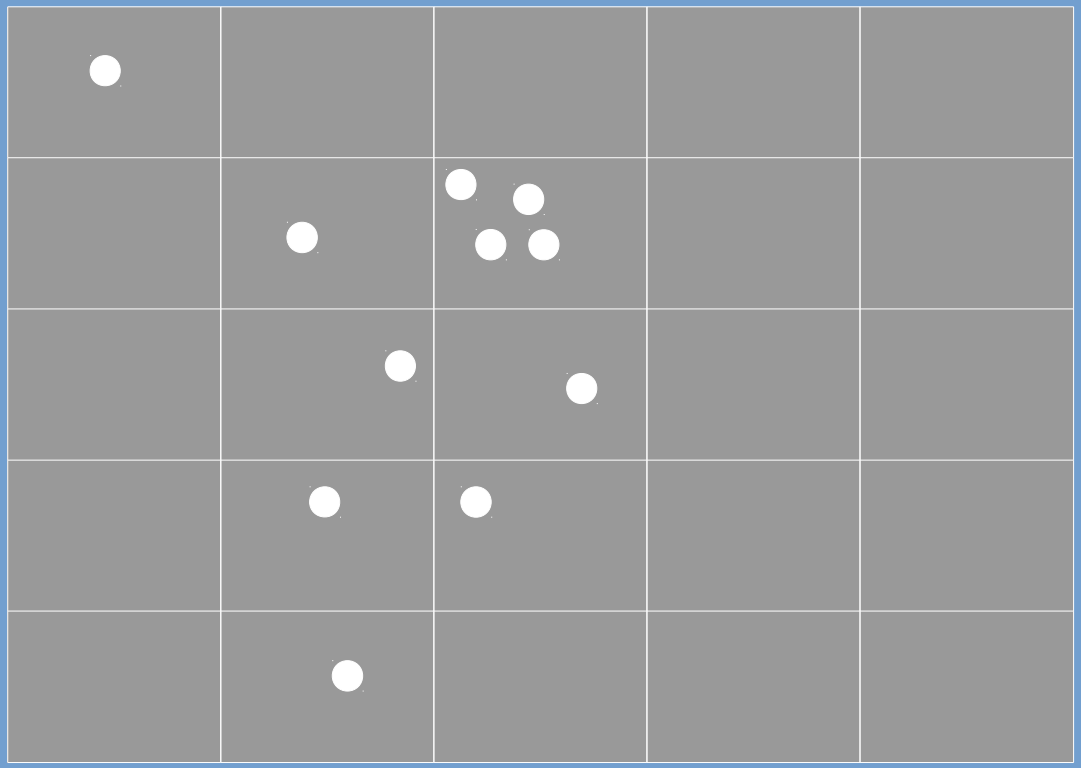
- Worthy of a lecture on its own
- Species names
  - Synonyms
  - Misapplication
  - Useful resources:
    - theplantlist.org
    - www.eol.org
    - www.ncbi.nlm.nih.gov
    - www.itis.gov
    - https://ropensci.org/tutorials/taxize_tutorial.html

Step 5: Data cleaning

- Coordinate errors
  - Spatial resolution
  - Zero lat/lon
  - Swapped lat and lon
  - How to detect:
    - Points in the sea (terrestrial organisms) or on land (marine organisms)
    - Country name mismatch
    - Elevational mismatch
    - Outliers with respect to environmental data

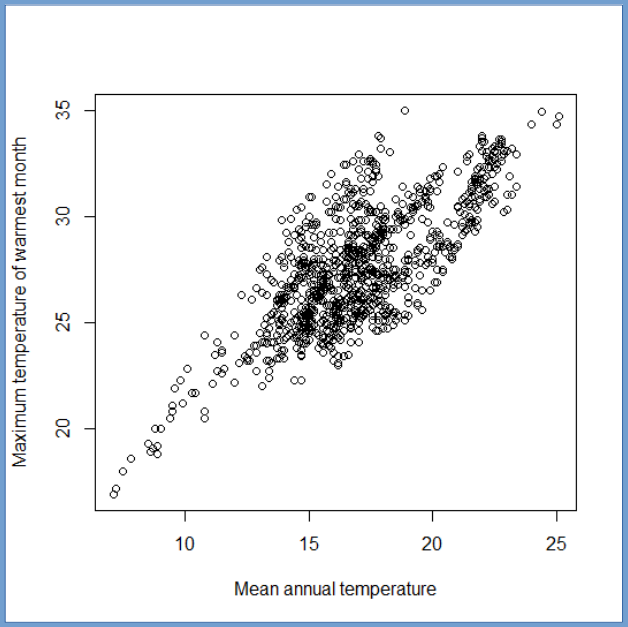
Step 5: Data cleaning

- Pseudoreplication



Step 5: Data cleaning

- Collinearity
  - Predictors that are highly correlated with one another ($r > 0.8$)
  - Can be a problem if one wants to understand environmental factors that limit species' distributions (Merow et al., 2013)
  - Not a problem if predicting distribution is the sole aim (Elith et al., 2011. Diversity and Distributions 17: 43)



Step 5: Data cleaning

- R package biogeo

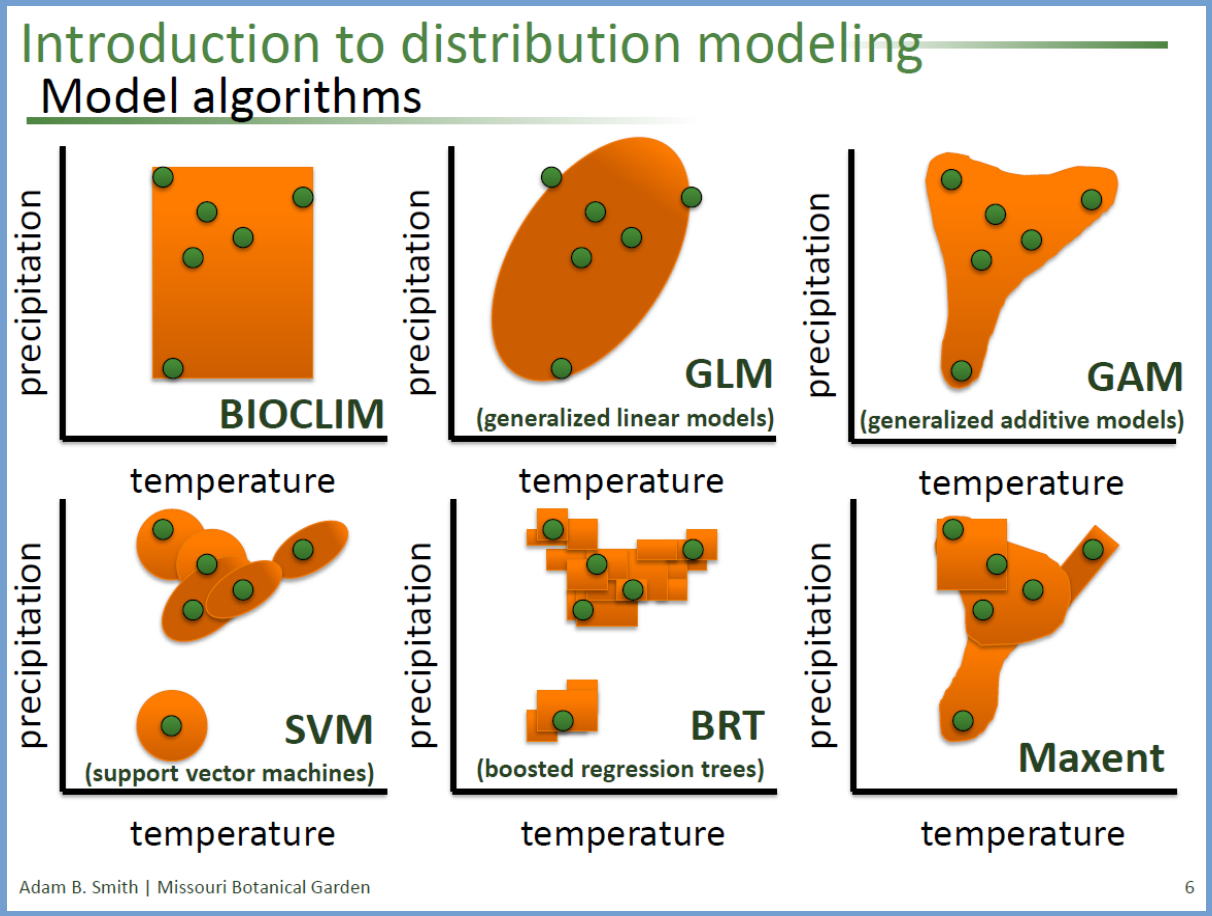
RStudio Source Editor

Stats_Toolbox_SDMs.R*

Source on Save | Run | Source

```
#Check for fossil records (if using GBIF data):
levels(as.factor(presClean$basisOfRecord))
#To remove fossil records:
presClean$Exclude[presClean$basisOfRecord=='FOSSIL_SPECIMEN'] = 1
presClean$Reason[presClean$basisOfRecord=='FOSSIL_SPECIMEN'] = 'Fossil' #Give a reason for exclusion

#Check species names:
levels(presClean$Species) #Everything is fine. All records have the same and correct species name

#Check temporal resolution of presence records:
hist(presClean$year, main='Histogram of year in which records were made', xlab='Year')

#Identify duplicates in grid cells (using the resolution of the environmental rasters you use. In this case = 10):
presClean = duplicatesexclude(presClean, res=10)
summary(as.factor(presClean$Exclude)) #0 = records that we will keep, 1 = records that will be removed
backClean = duplicatesexclude(backClean, res=10)
summary(as.factor(backClean$Exclude)) #0 = records that we will keep, 1 = records that will be removed

#Exclude zero lat & lon:
presClean$Exclude[presClean$x!=0 & presClean$y!=0] = 1
presClean$Reason[presClean$x!=0 & presClean$y!=0] = 'Zero lat or lon' #Give a reason for exclusion

#Move points in sea to nearest cell on land:
landDat = nearestcell(presClean, bio[[1]])
presClean <- landDat$dat
landDatB = nearestcell(backClean, bio[[1]])
backClean = landDatB$dat
#See points that were changed:
corrected = landDat$dat$Correction
iCorrected = grep("7", corrected)
landDat$dat[iCorrected,] #view records that were corrected

#Remove other points in the sea and any records for which we can't get environmental data (from "bio", our raster stack)
presClean = missingvalsexclude(bio[[1]], presClean)
summary(as.factor(presClean$Exclude)) #0 = records that we will keep, 1 = records that will be removed
backClean <- missingvalsexclude(bio[[1]], backClean)
```

Step 6: Running a model



http://www.earthskysea.org/!ecology/sdmShortCourseKState2012/sdmShortCourse_kState.pdf

Step 6: Running a model

- Maxent (R package dismo)
- Check out the vignette help document on SDM from dismo (is in the script)
- Useful refs:
  - Merow et al. (2013) Ecography 36: 1058
  - Yakulic et al. (2012) Methods in Ecology and Evolution 4: 236

Step 6: Running a model

- Some basic things to consider:

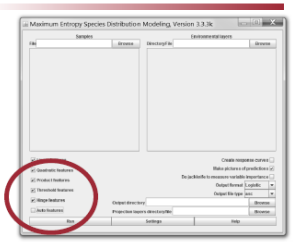
Maxent under the hood

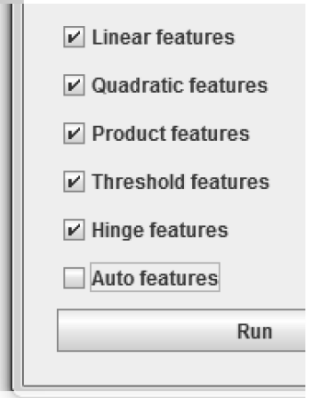
Constraints

"Feature"	Constraint	Name	min num of presences
	mean	linear	>0
	variance	quadratic	≥10
	proportion above/below threshold	step	≥15
	(as above) & mean	hinge	≥80
	covariance	product (2 variables)	≥80
	proportion in each category	categorical	>0

Pr(env | presence)

env





Use only hinge features to avoid overly-complex fits. Elith et al. 2010. The art of modeling range-shifting species. Methods Ecology & Evol 1:330-342.

Adam B. Smith | Missouri Botanical Garden

53

http://www.earthskysea.org/!ecology/sdmShortCourseKState2012/sdmShortCourse_kState.pdf

Step 6: Running a model

- Some basic things to consider:
  - Regularization coefficient (β) – penalizes overfitting, but it is user-specified.
  - Try a range of β values and evaluate model fit (e.g. using AUC) (Merow et al., 2013)

Step 6: Running a model

RStudio Source Editor

Stats_Toolbox_SDMs.R*

Source on Save | Run | Source

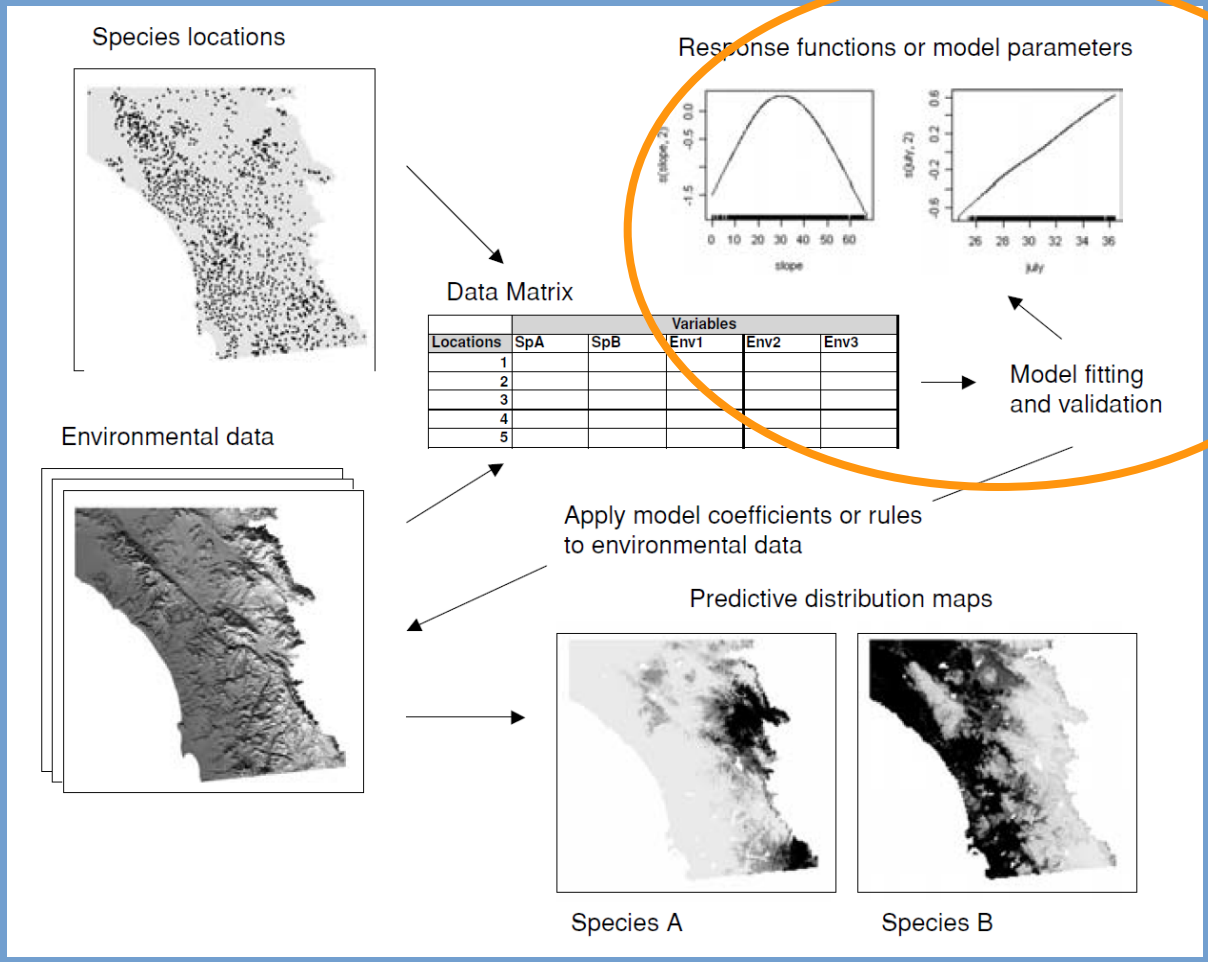
ed | Next | Prev | All | Replace | Replace | All

In selection | Match case | Whole word | Regex | Wrap

```
307  #Full model with all occurrences:
308  betaRCs <- seq(0.05,0.95,0.05)
309  bestAUC = read.csv('bestAUC.csv')$bestAUC
310  mxMod = maxent(x = modLayers,
311                 p = p,
312                 a = a,
313                 args = c('-p','nothreshold',paste0('beta_lqp=',betaRCs[bestAUC]),
314                          paste0('beta_hinge=',betaRCs[bestAUC]), 'outputformat=raw'),
315                 biasfile=biasGrid,
316                 removeDuplicates=T)
317
```

http://www.earthskysea.org/!ecology/sdmShortCourseKState2012/sdmShortCourse_kState.pdf

Step 7: Checking your model



Step 7: Checking your model

- Model accuracy
  - Area under the receiver operating characteristic curve (AUC)
  - Other measures too and suggested that you use some of these (Sensitivity, specificity, Boyce Index…)
  - Usually check by bootstrapping:
    - e.g. 100 model runs
    - 70% of data used to build a model (training data)
    - 30% used to evaluate (test data)

Step 7: Checking your model

RStudio Source Editor

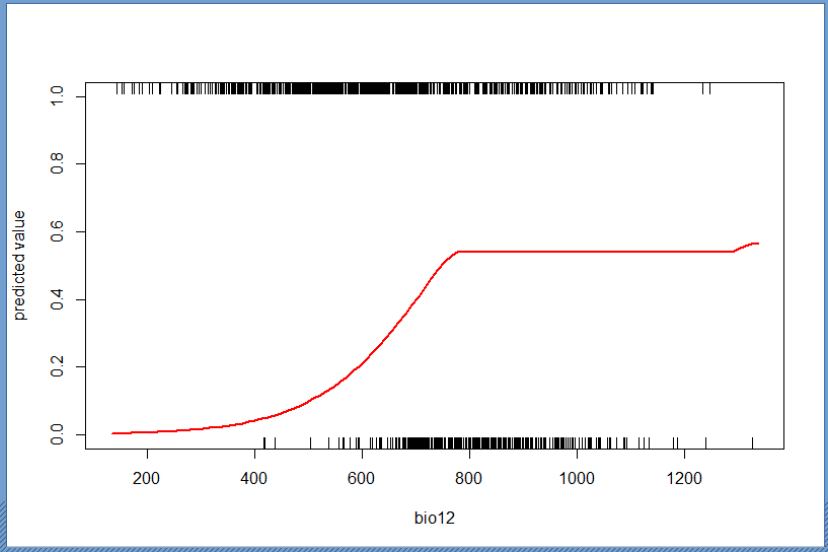
Stats_Toolbox_SDMs.R*

Source on Save | Run | Source

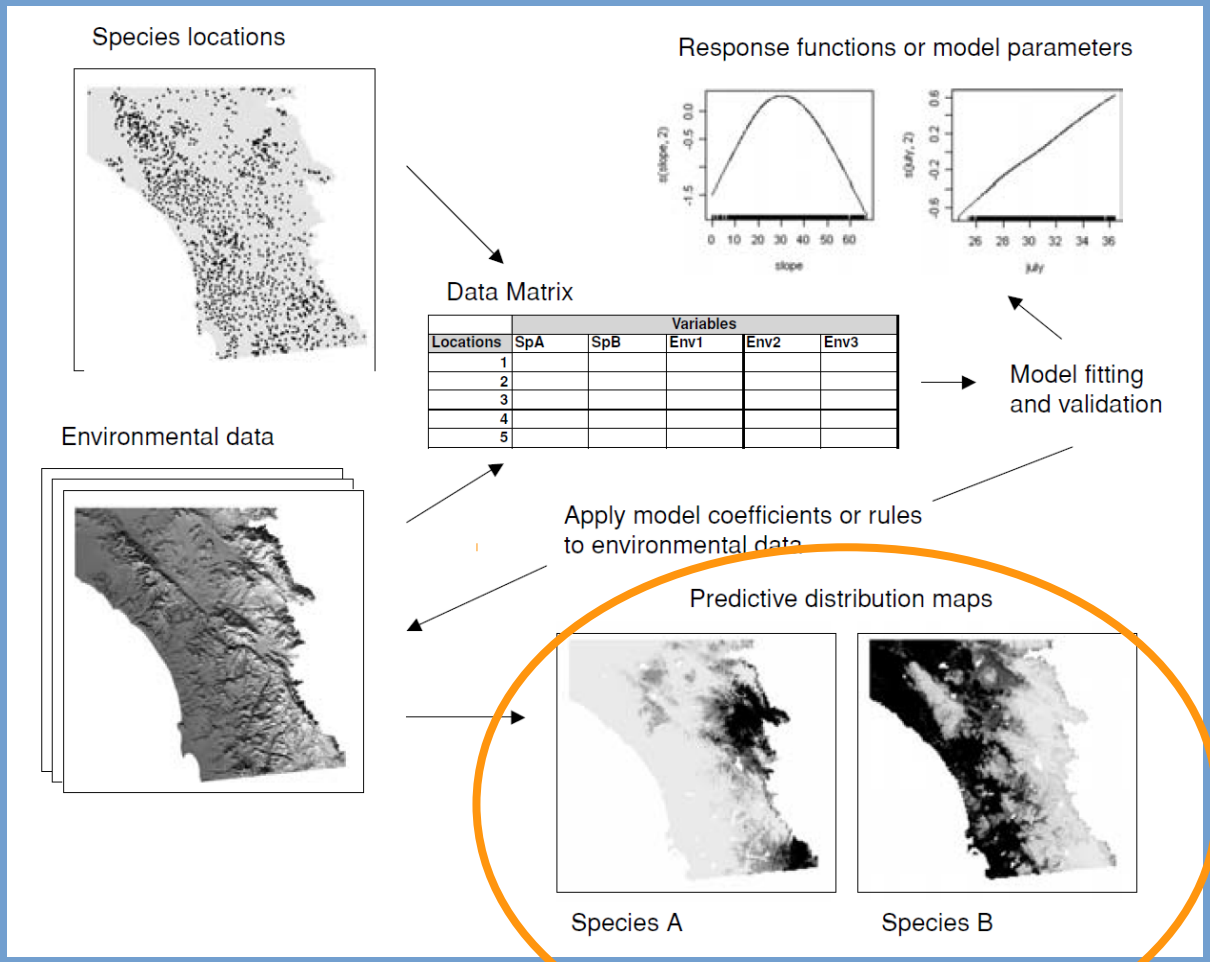
```
#-------------------------------------------------------------------------------
#Model evaluation (AUC)
#-------------------------------------------------------------------------------
AUCs <- {} #Empty vector to store AUC values
betaRCs <- seq(0.05,0.95,0.05) #Get a range of values for regularization coefficient
bestAUC = read.csv('bestAUC.csv')$bestAUC #Read in best value for above
for(i in 1:100){
  print(i)
  #Random number generator for selecting 70% of the presence data for model building:
  pSamp = sample(x = c(1:nrow(p)), size=round(nrow(p)*0.7,0), replace=F)
  ptrain = p[pSamp,] #Select 70% of the presence data for model building
  ptest = p[-pSamp,] #Use other 30% of presence for evaluating the model (test data)
  #Random number generator for selecting 70% of the background data for model building:
  aSamp = sample(x = c(1:nrow(a)), size=round(nrow(a)*0.7,0), replace=F)
  atrain = a[aSamp,] #70% of background data for model building
  atest = a[-aSamp,] #Use other 30% of presence for evaluating the model (test data)
  #Maxent model:
  mxMod = maxent(x = modLayers,
                 p = ptrain,
                 a = atrain,
                 args = c('-p','nothreshold',paste0('beta_lqp=',betaRCs[bestAUC]),paste0('beta_hinge=',betaRCs[bestAUC])),
                 biasfile=biasGrid, #Use a bias grid to reduce sampling bias
                 removeDuplicates=T)
  #Look at model evaluation (AUC):
  e = evaluate(mxMod, p=ptest, a=atrain, x=modLayers)
  #Store AUC values:
  AUCs[i] = e@auc
}

#Look at model evaluation (AUC):
mean(AUCs) #Get mean AUC values
1.96 * sd(AUCs)/sqrt(100) #95% CI

#-------------------------------------------------------------------------------
```

Step 7: Checking your model

- Response curves
  - Are they biologically realistic?
  - Is there enough sampling across the full range of the predictor?

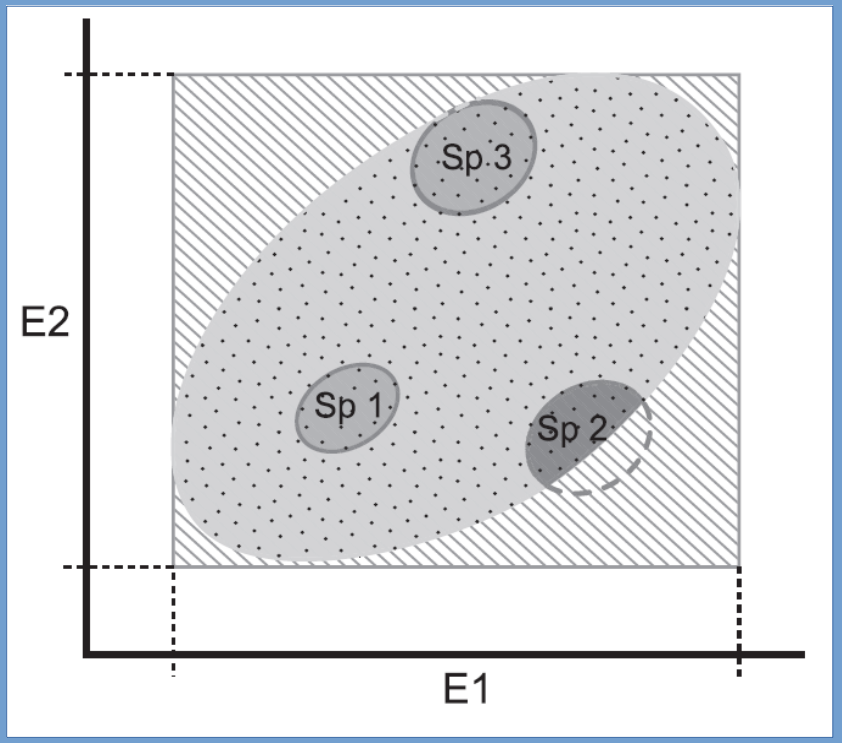


Step 8: Projecting your model



Step 8: Projecting your model

- Very easy to make a map, but is it a good map?
- Model extrapolation
  - novel environmental space

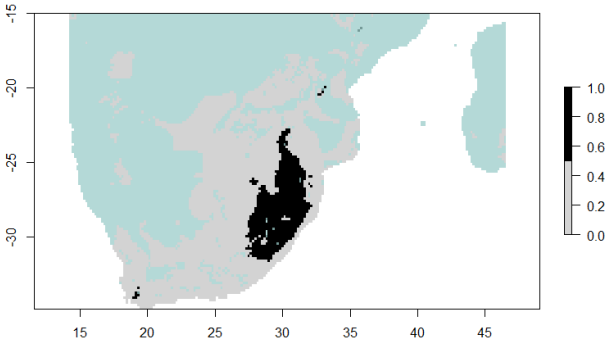


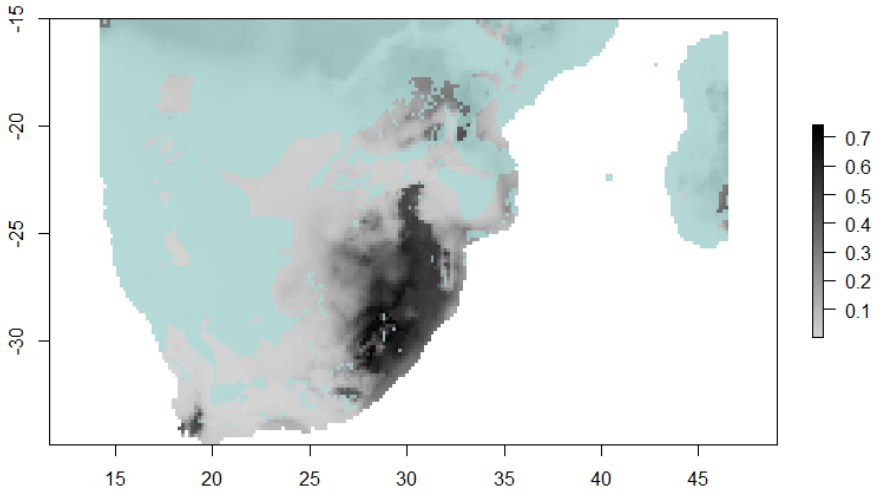
Zurell et al., 2012. Diversity and Distributions 18: 628

Step 8: Projecting your model

- Novel environmental space refs:
  - Multivariate environmental similarity surface (MESS) (Elith et al., 2010. Methods in Ecology & Evolution 1: 330)
  - Environmental overlap masks (Zurell et al., 2012. Diversity and Distributions 18: 628)

THE END


-15
-20
-25
-30
15
20
25
30
35
40
45
1.0
0.8
0.6
0.4
0.2
0.0


-15
-20
-25
-30
15
20
25
30
35
40
45
0.7
0.6
0.5
0.4
0.3
0.2
0.1

Other useful references

- Elith & Leathwick (2009). Annual Rev. Ecol. Evol. Syst. 40: 677
- Elith et al. (2011). Diversity and Distributions 17: 43
- Liu et al. (2013) Journal of Biogeography 40:778
- Renner et al. (2015). Methods in Ecology and Evolution 6: 366
- And many more… (so carry on reading!)

SEEC Stats Toolbox

Want to broaden your stats knowledge? Unsure of what you can do with your data? Still developing your proposal?

Join us for the **SEEC Stats Toolbox** seminars

Our next seminar:

Topic: **Occupancy Models**
Who: Prof Res Altwegg
When: **Thursday 25 August 2016 (1-2pm)**
Where: UCT campus (venue TBC)

More details: www.seec.uct.ca.za

SEEC - Statistics in Ecology, Environment and Conservation