Species Distribution Modeling In R

Announcements

- This is our last official lecture! Thank you for such a great semester :)
- Final Projects due on May 13th
    - Remember it doesn’t have to be anything crazy haha
    - Concise & meaningful >>> long and drawn out
    - Feel free to revise your project scope to meet reasonable timeframe expectations
- Next few weeks will be focused on final project time and finishing up labs.
- Lab this week should be short, use this slide deck as a tutorial!

1. What is a Species Distribution Model?

What is a Species Distribution Model (SDM)?

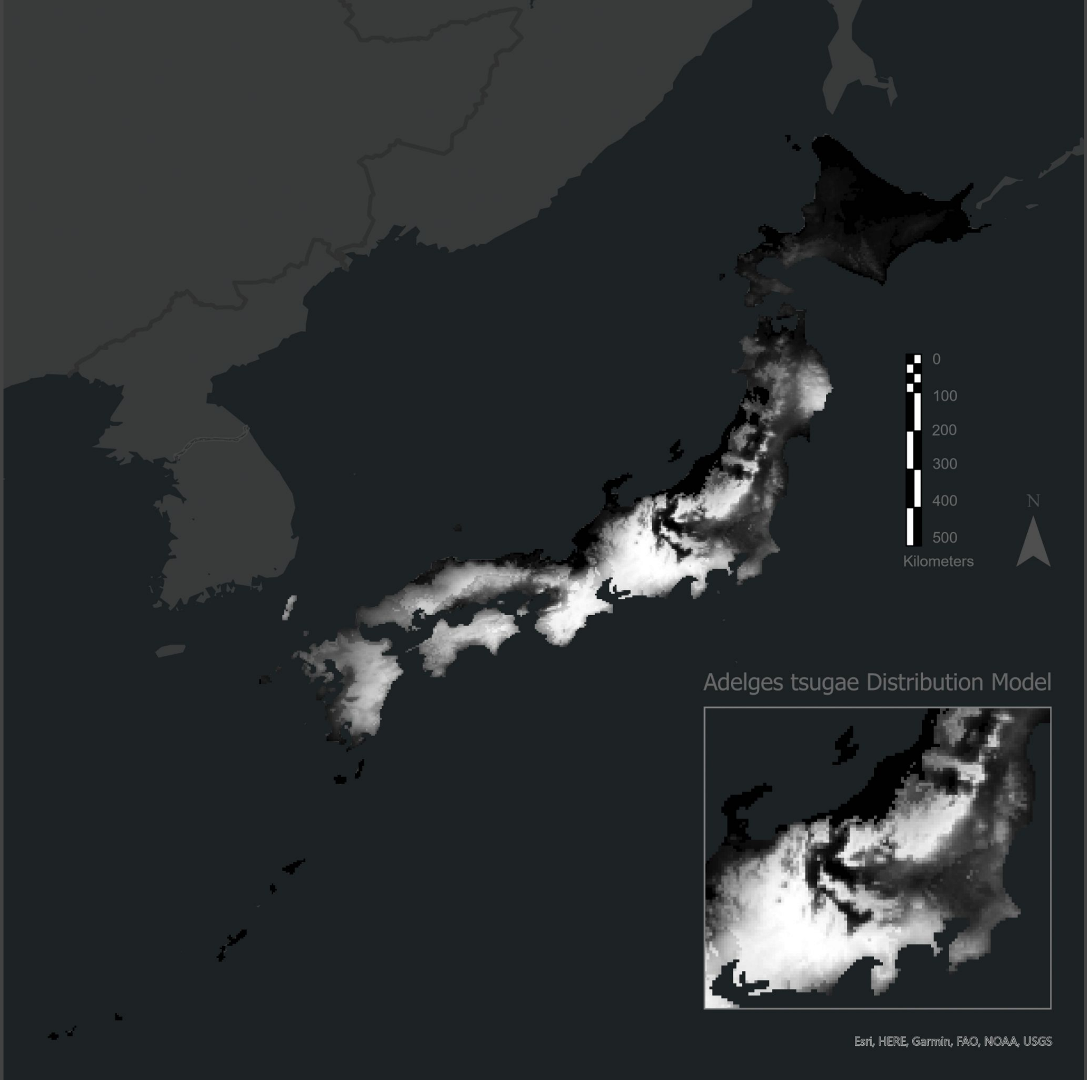
- A mathematical tool used in biogeography to predict the geographical distribution of a species based on environmental conditions and spatial data.
- These models are useful for describing suitable habitats for species.
- SDMs function by correlating the known locations of a species with environmental variables such as temperature, rainfall, and elevation.
- These models can predict areas that are likely to support the species but where it has not yet been observed.

Why are SDMs useful?

- Targeting locations for conservation & restoration
- Assessing the impacts of environmental changes
- Aiding in conservation planning
- Predicting shifts in species distributions due to climate change.

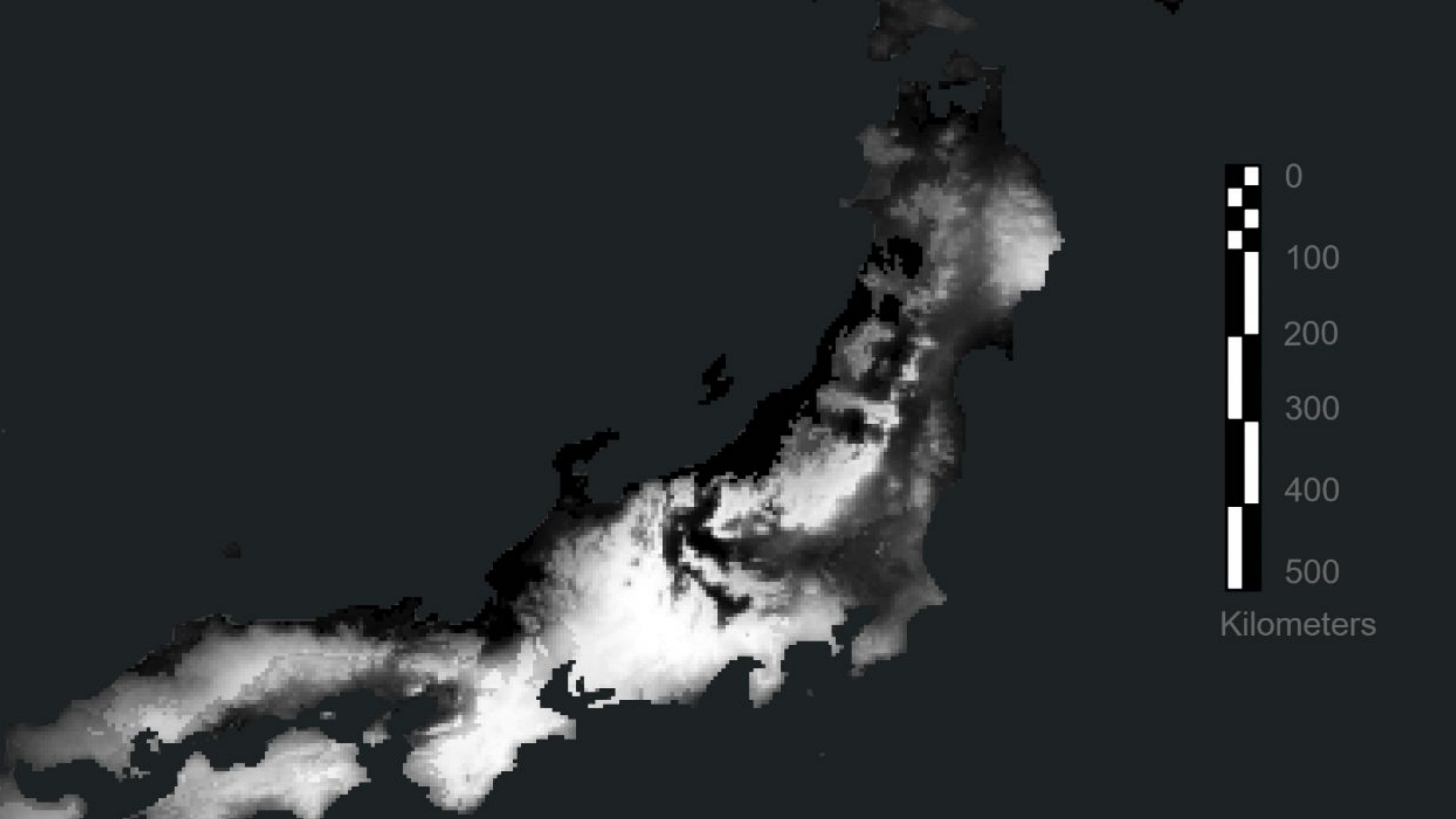
SDMs vs Occupancy Models

- Species Distribution Model
  - **Probability value represents the likelihood that the environmental conditions at a given location are suitable for a species.**
  - Uses presence-only data or presence-absence data, along with environmental covariates like climate and topography.
- Occupancy Model
  - **Estimate the probability of a species being present in a given area** while accounting for imperfect detection during surveys.
  - Primarily require presence-absence data collected over multiple visits to a site to explicitly model detectability.



Adelges tsugae Distribution Model

Esri, HERE, Garmin, FAO, NOAA, USGS


0
100
200
300
400
500
Kilometers

2. Basic Species Distribution Models

Species Distribution Model Types

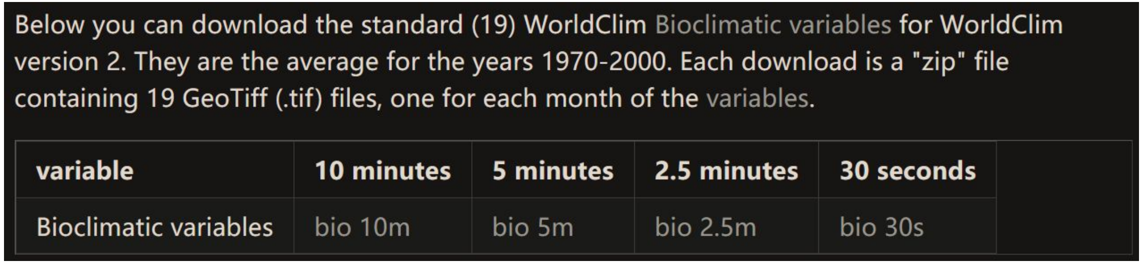
- Today!
    - **Bioclim Models**
    - Maxent (Maximum Entropy)
- Others
    - Domain Models
    - Random Forests (RF)
    - Support Vector Machines (SVM)
    - Artificial Neural Networks (ANN)

Bioclim Models

- Bioclim is the classic 'climate-envelope-model'.
- Requires minimal input and optimization
- Caveats
  - Because of it has less specificity, it doesn’t perform as well as other more specialized models
  - Not good for predicting climate change effects
- Still used for basic predictions, but is not very robust
- Packages
  - dismo contains bioclim() function
- All you need are occurrence points and a raster stack of environment layers.

Bioclim Model Example - Data Download

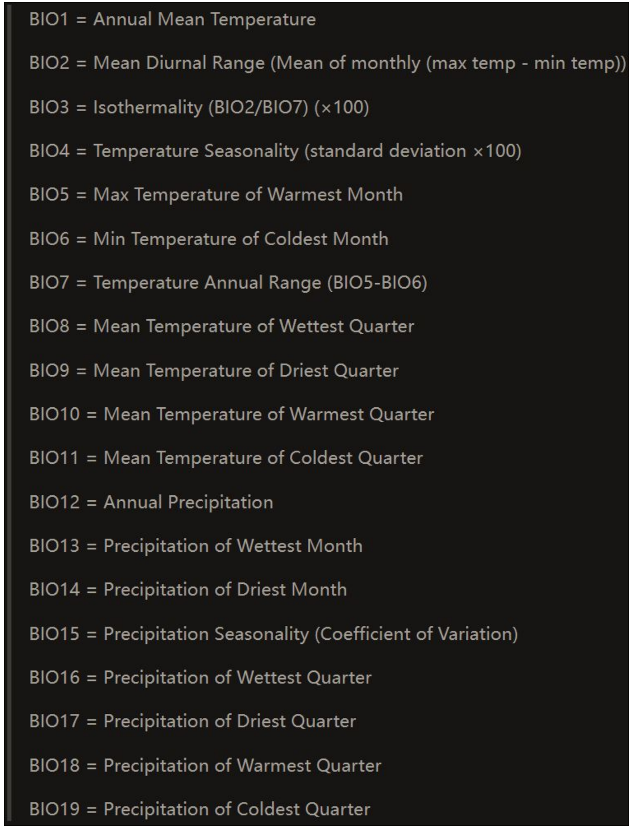
- WorldClim is a great resource for bioclimatic datasets



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

- Variables such as:
  - Max/Min Temperature of Warmest/Coldest Months
  - Precipitation of Wettest & Driest Months/Quarters
  - Annual Precipitation & Annual Mean Temperature



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)

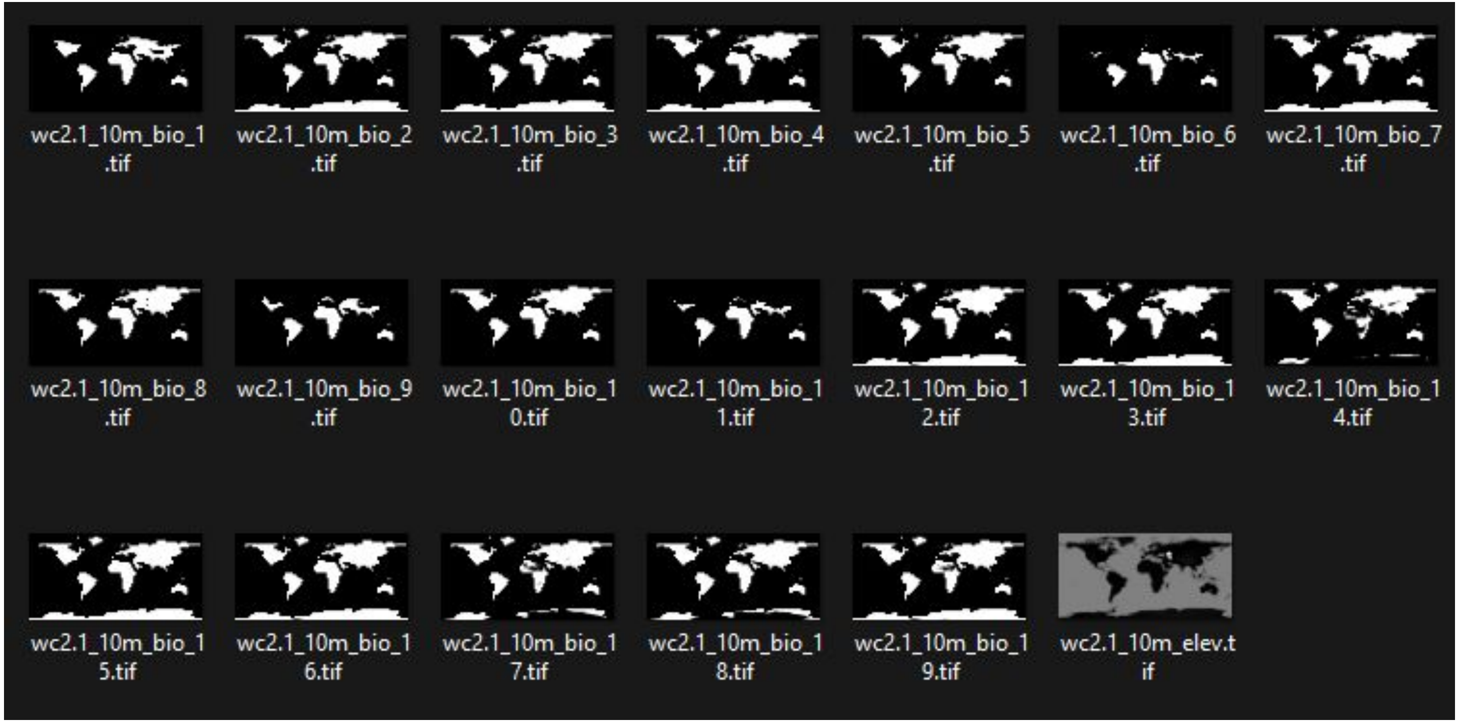
BIO16 = Precipitation of Wettest Quarter

BIO17 = Precipitation of Driest Quarter

BIO18 = Precipitation of Warmest Quarter

BIO19 = Precipitation of Coldest Quarter

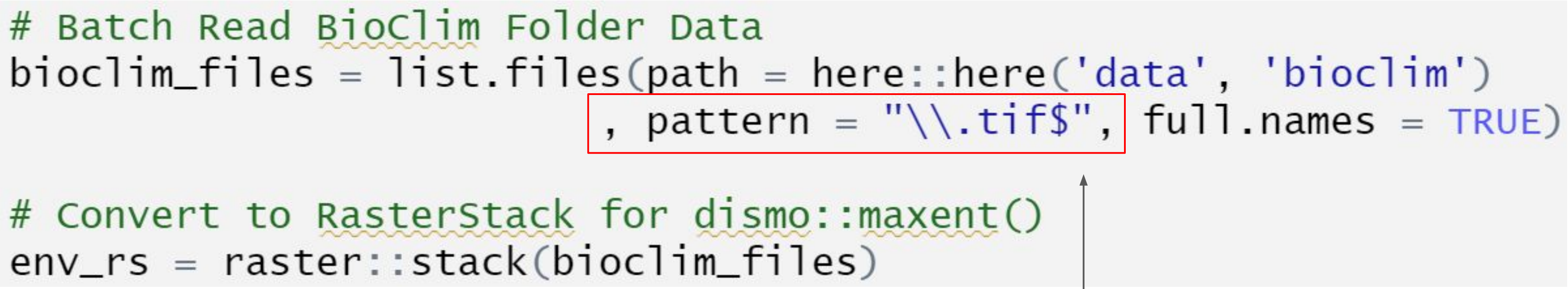
Bioclim Model Example - Data Inspection



Lot of files to load in…..

Bioclim Model Example - Efficiently Reading in Data

- You probably don’t want to be reading in all ~19 bioclimatic rasters layer individually, so we can read in the whole folder with list.files() & stack()!
- Note that we must use the raster package, as the dismo package which runs many types of SDMs has not yet been updated to use terra.



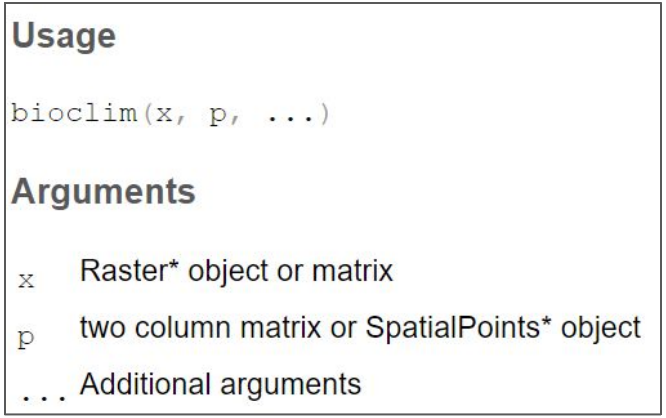
```
# Batch Read BioClim Folder Data
bioclim_files = list.files(path = here::here('data', 'bioclim')
                         , pattern = "\\.tif$", full.names = TRUE)

# Convert to RasterStack for dismo::maxent()
env_rs = raster::stack(bioclim_files)
```

Optional: Filters out non-tif files from list

Bioclim Model Example - Prepare Presence PTS

- The dismo bioclim() function will run a bioclimatic SDM for us!



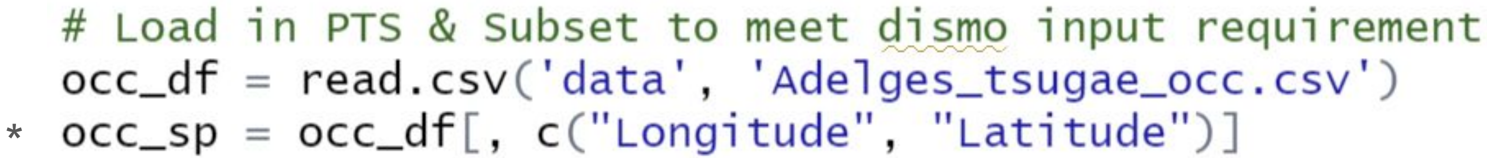
Usage

```
bioclim(x, p, ...)
```

Arguments

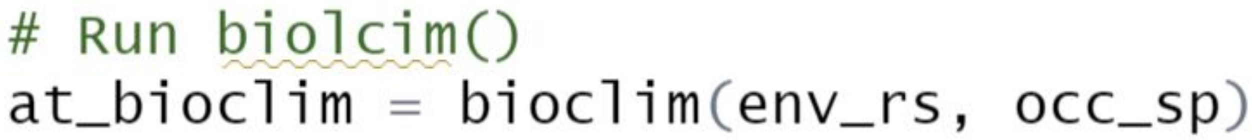
`x`	Raster* object or matrix
`p`	two column matrix or SpatialPoints* object
`...`	Additional arguments

Looks like we need to only provide LAT/LONG data!



```
  # Load in PTS & Subset to meet dismo input requirement
  occ_df = read.csv('data', 'Adelges_tsugae_occ.csv')
* occ_sp = occ_df[, c("Longitude", "Latitude")]
```

Bioclim Model Example - Run Bioclim Model

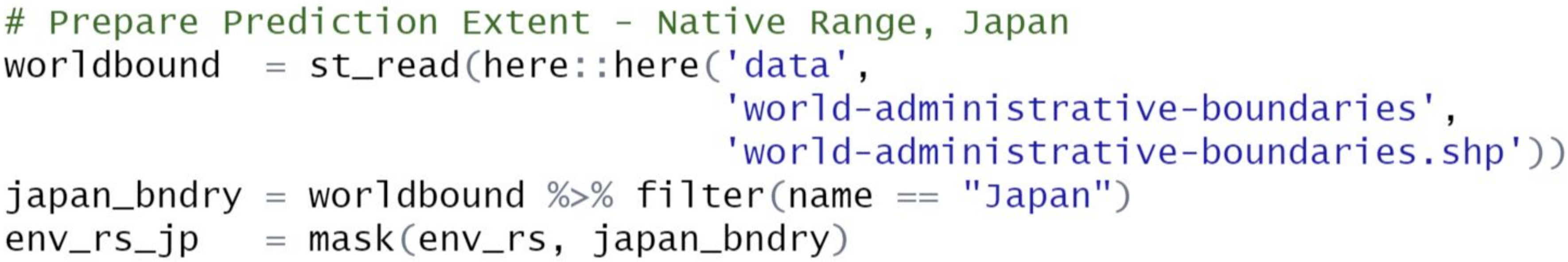


```
# Run biolcim()
at_bioclim = bioclim(env_rs, occ_sp)
```

- This is awesome! We ran the model :)
- Where is the SDM?
  - Well we just made the model, but this model is flexible and can be applied to different datasets.
  - We need to choose a study area to predict where the potential species distribution is located!

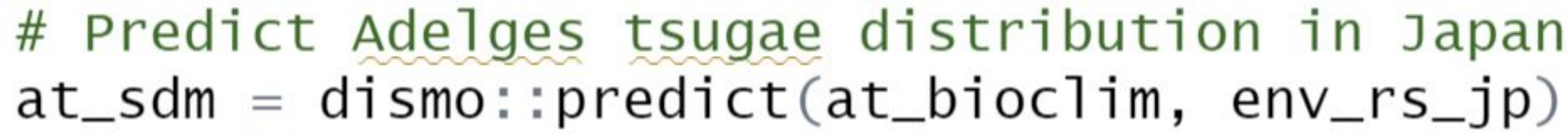
Bioclim Model Example - Prepare a Prediction Extent

- To create a prediction extent, we can clip the environmental raster stack to our study area - in this case Japan!
- The SDM we created in the last slide is an interpretation of our species' living preferences, so we want to give it a matching set of layers to **infer what other locations would be suitable based on its closeness to the species' preferences**.



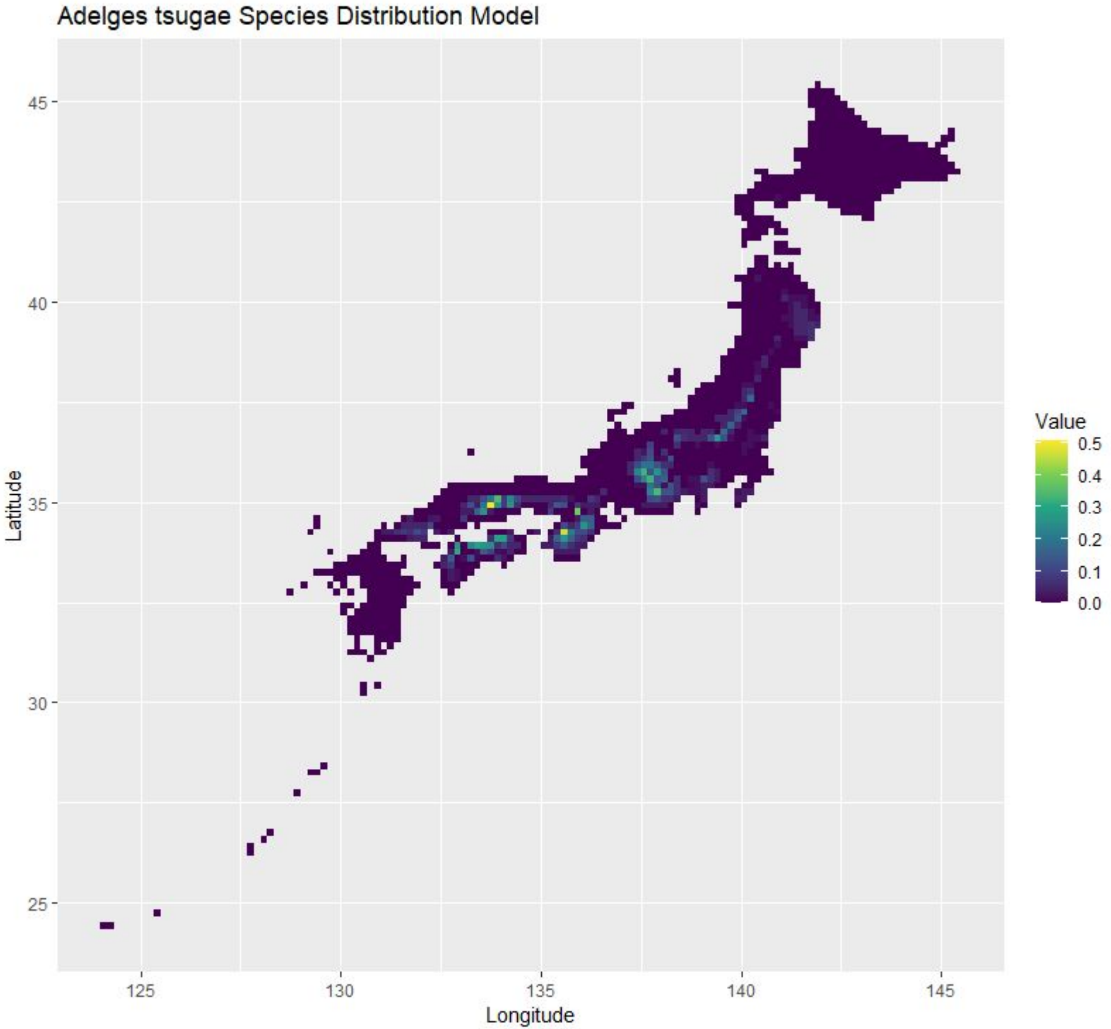
```r
# Prepare Prediction Extent - Native Range, Japan
worldbound  = st_read(here::here('data',
                                 'world-administrative-boundaries',
                                 'world-administrative-boundaries.shp'))
japan_bndry = worldbound %>% filter(name == "Japan")
env_rs_jp   = mask(env_rs, japan_bndry)
```

Bioclim Model Example - PREDICT!



```
# Predict Adelges tsugae distribution in Japan
at_sdm = dismo::predict(at_bioclim, env_rs_jp)
```

- We can use dismo::predict() to predict the likelihood that the environmental conditions throughout different locations in Japan are suitable for Adelges tsugae.
- Let’s plot it!


Adelges tsugae Species Distribution Model
Latitude
45
40
35
30
25
125
130
135
140
145
Longitude
Value
0.5
0.4
0.3
0.2
0.1
0.0

3. MaxEnt

Species Distribution Model Types

- Today!
  - Bioclim Models
  - **Maxent (Maximum Entropy)**
- Others
  - Domain Models
  - Random Forests (RF)
  - Support Vector Machines (SVM)
  - Artificial Neural Networks (ANN)

Maximum Entropy Modeling (Maxent)

- Methodology: Maxent estimates the probability distribution of species' occurrences based on environmental constraints, using the principle of maximum entropy.
- It operates under the **assumption that the observed distribution of a species is the most spread out, or has maximum entropy, given the environmental constraints**.
- Use: Particularly popular for species distribution modeling with presence-only data. It's robust, efficient, and can handle sparse data scenarios well.
- Packages
    - **dismo** provides an interface to run Maxent models if you have Maxent software
    - **ENMeval** package can mathematically determine optimal settings for Maxent models!
    - Maxent Download: https://biodiversityinformatics.amnh.org/open_source/maxent/

MaxEnt Settings - Feature Combinations

- MaxEnt models can be specified by different combinations of five feature combinations:
    - L = linear
    - Q = quadratic
    - H = hinge
    - P = product
    - T = threshold

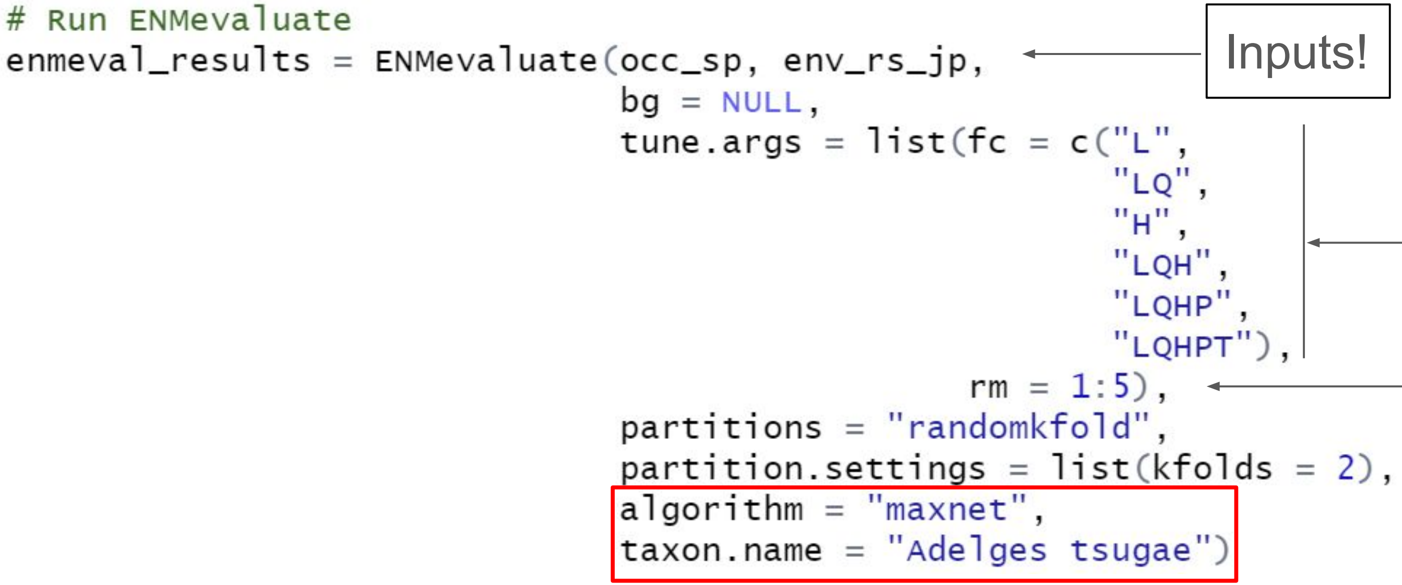
MaxEnt Settings - Beta Multiplier

- Regularization is used in MaxEnt models to prevent overfitting, ensuring that the model generalizes well to new, unseen data.
- Regularization: In MaxEnt, regularization typically involves adding a penalty to the complexity of the model.
- **Beta Multiplier (β) modifies the impact of this regularization**.
- Specifically, it scales the penalty applied to the entropy of the distribution:
  - High β = penalize complexity (number of variables) more.
    - A too-high beta can lead to a model that is too simple and fails to capture important patterns.
  - Low β = the model places less emphasis on penalizing complexity.
    - A too-low beta might create a model that is overly complex and fits the noise in the training data, rather than underlying patterns applicable more broadly.

Optimized MaxEnt Settings - ENMeval Package

- ENMeval determines the best arguments for your species' dataset such as the beta multiplier and the feature combination.
- ENMeval determines which feature combination is best suited and is setup as inputs for our MaxEnt model. For example, LQH can be a statistically better model choice than just a pure L feature model!

1. Run ENMeval to determine best model settings



```r
# Run ENMevaluate
enmeval_results = ENMevaluate(occ_sp, env_rs_jp,
                              bg = NULL,
                              tune.args = list(fc = c("L",
                                                      "LQ",
                                                      "H",
                                                      "LQH",
                                                      "LQHP",
                                                      "LQHPT"),
                                               rm = 1:5),
                              partitions = "randomkfold",
                              partition.settings = list(kfolds = 2),
                              algorithm = "maxnet",
                              taxon.name = "Adelges tsugae")
```

Inputs!

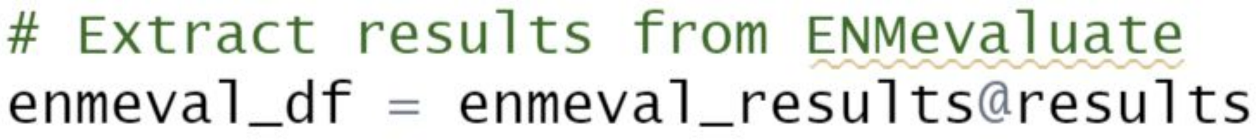
Feature Combinations

Range of beta multiplier values you want to test.

```
enmeval_results    Large ENMevaluation ( 584.2 MB)
```

What do I do with this?

2. Extract best model settings

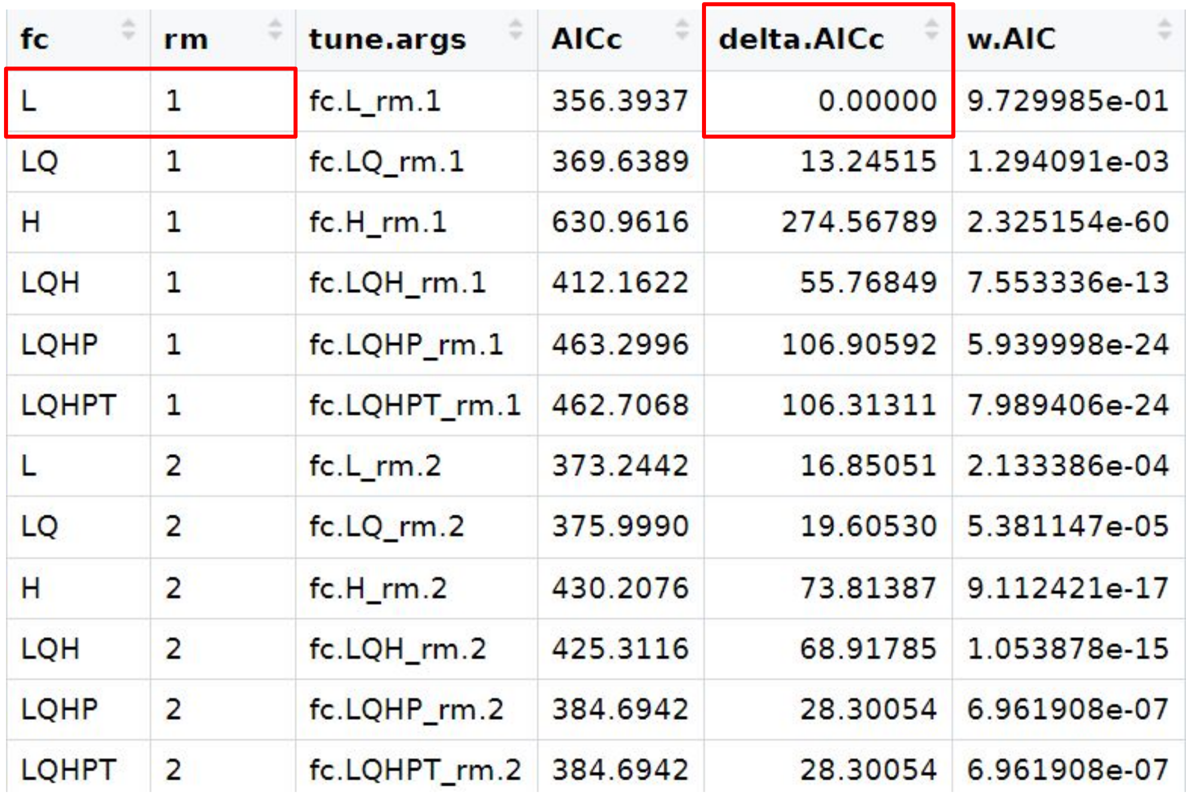


```
# Extract results from ENMevaluate
enmeval_df = enmeval_results@results
```

- Extract results with this weird method haha
- Returns a usable dataframe :)

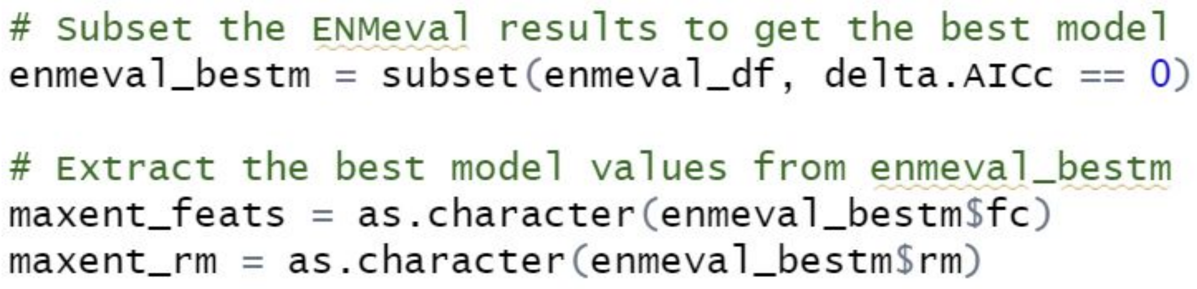
3. Inspect ENMeval to determine best settings

- ENMeval runs through all relevant permutations of feature combinations & beta multiplier (rm)!
- **delta.AICc = 0 is the best model setting!**
- This tells us the model performs best when linear, and is hurt when other combinations are added.



fc	rm	tune.args	AICc	delta.AICc	w.AIC
L	1	fc.L_rm.1	356.3937	0.00000	9.729985e-01
LQ	1	fc.LQ_rm.1	369.6389	13.24515	1.294091e-03
H	1	fc.H_rm.1	630.9616	274.56789	2.325154e-60
LQH	1	fc.LQH_rm.1	412.1622	55.76849	7.553336e-13
LQHP	1	fc.LQHP_rm.1	463.2996	106.90592	5.939998e-24
LQHPT	1	fc.LQHPT_rm.1	462.7068	106.31311	7.989406e-24
L	2	fc.L_rm.2	373.2442	16.85051	2.133386e-04
LQ	2	fc.LQ_rm.2	375.9990	19.60530	5.381147e-05
H	2	fc.H_rm.2	430.2076	73.81387	9.112421e-17
LQH	2	fc.LQH_rm.2	425.3116	68.91785	1.053878e-15
LQHP	2	fc.LQHP_rm.2	384.6942	28.30054	6.961908e-07
LQHPT	2	fc.LQHPT_rm.2	384.6942	28.30054	6.961908e-07

4. Extract the best model settings

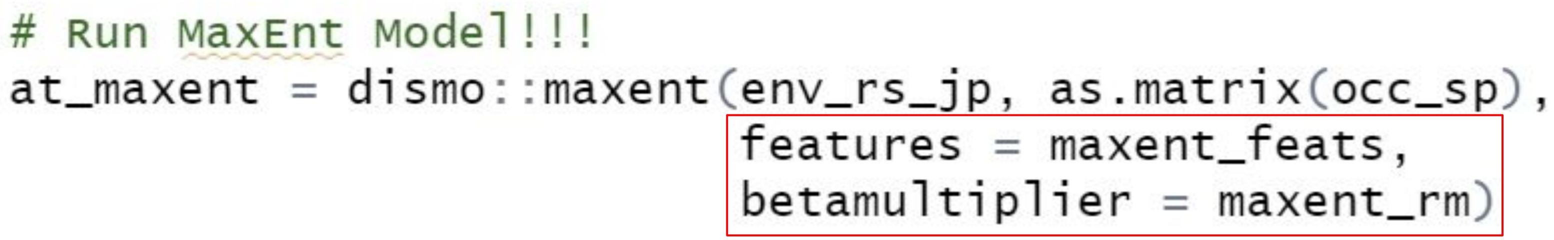


```
# Subset the ENMeval results to get the best model
enmeval_bestm = subset(enmeval_df, delta.AICc == 0)

# Extract the best model values from enmeval_bestm
maxent_feats = as.character(enmeval_bestm$fc)
maxent_rm = as.character(enmeval_bestm$rm)
```

1) Subset for the model with the lowest delta AICc
2) Extract the best model’s feature combination (in this case L)
3) Extract the best model’s beta multiplier (in this case 1)

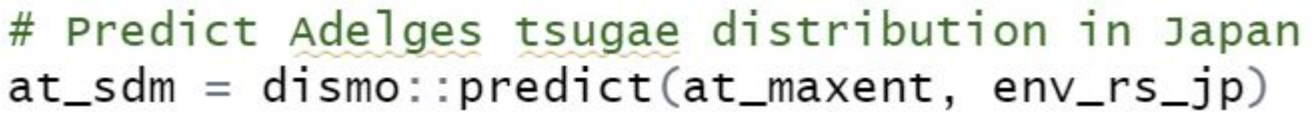
5. Run MaxEnt with our layers and best settings!



```
# Run MaxEnt Model!!!
at_maxent = dismo::maxent(env_rs_jp, as.matrix(occ_sp),
                          features = maxent_feats,
                          betamultiplier = maxent_rm)
```

- We can use the extracted best maxent features and beta multiplier values as the inputs for the MaxEnt model :)

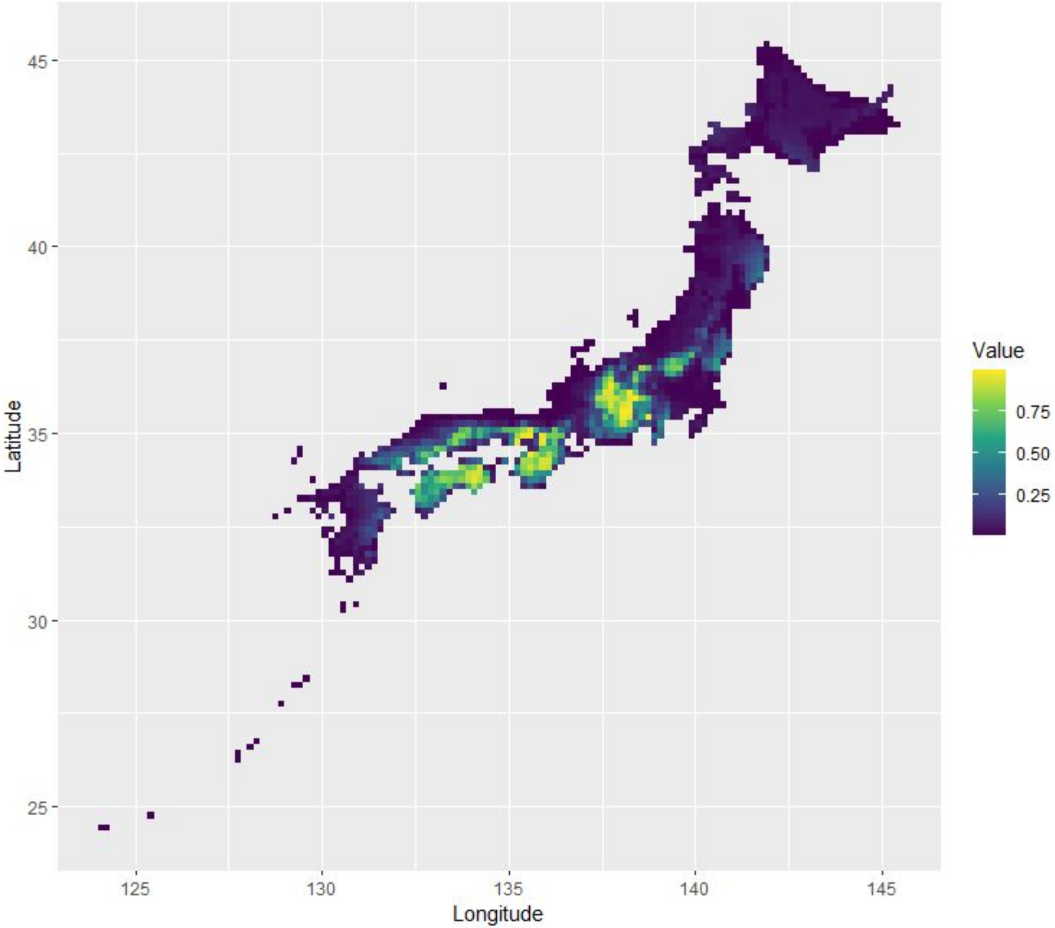
6. Predict an output map



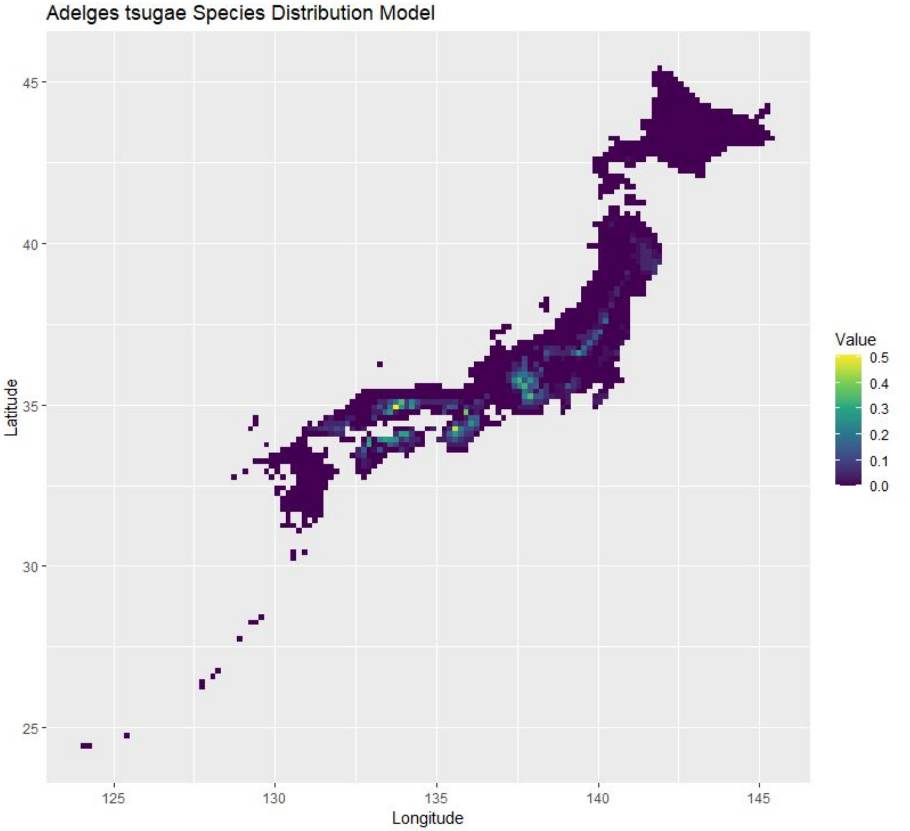
```
# Predict Adelges tsugae distribution in Japan
at_sdm = dismo::predict(at_maxent, env_rs_jp)
```

- Same method as before!
- We just swap out at_bioclim for at_maxent

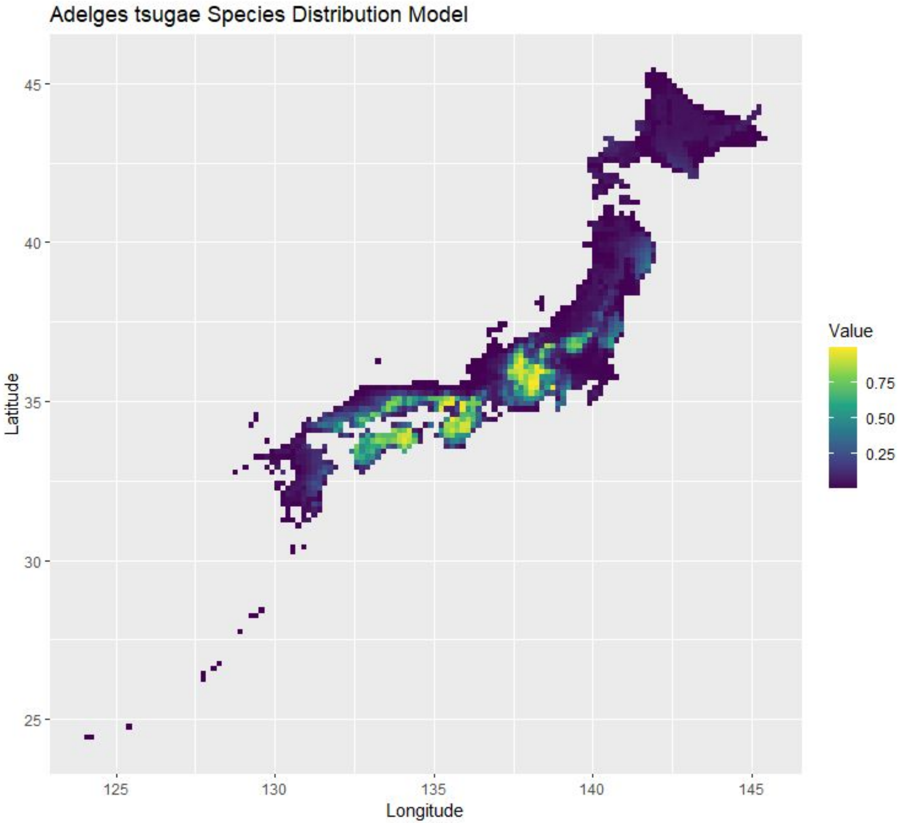
Adelges tsugae Species Distribution Model



Bioclim Model



MaxEnt Model



4. More Species Distribution Models

Domain Models

- The Domain algorithm (Carpenter et al. 1993) that has been extensively used for species distribution modeling.
- The Domain algorithm computes the Gower distance between environmental variables at any location and those at any of the known locations of occurrence ('training sites'). For each variable the minimum distance between a site and any of the training points is taken. To integrate over environmental variables, the maximum distance to any of the variables is used. This distance is subtracted from one, and (in this R implementation) values below zero are truncated so that the scores are between 0 (low) and 1 (high).
- Packages
    - dismo contains domain() function

Random Forests (RF)

- Methodology: An ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- Use: Effective for handling large datasets with high dimensionality and complex, non-linear relationships without overfitting.
- Packages
    - randomForest for model, and dismo for predicting the model

Support Vector Machines (SVM)

- Methodology: SVMs are a set of supervised learning methods used for classification, regression, and outliers detection. For SDMs, SVMs identify the optimal separating hyperplane between presence and absence (or pseudo-absence) data in a high-dimensional space.
- Use: Effective for datasets where the number of dimensions exceeds the number of samples, and is versatile in the choice of the kernel function.

Artificial Neural Networks (ANN)

- Methodology: ANNs are computing systems vaguely inspired by the biological neural networks. They learn to perform tasks by considering examples, generally without being programmed with task-specific rules.
- Use: Suitable for modeling complex patterns and interactions in large datasets, though they require substantial data for training to avoid overfitting.

Boosted Regression Trees (BRT)

- BRT combines the strengths of two algorithms:
  - regression trees: which are capable of handling complex data structures
  - Boosting: a method of sequentially improving a model's accuracy by focusing on areas of poor performance.
- Use: Effective for improving model accuracy and handling various types of data, including presence-absence, count, and continuous data.