Contents » Spatial data analysis » 5. Spatial distribution models

5. Spatial distribution models

This page shows how you can use the Random Forest algorithm to do regression or supervised classification with spatial data. This could be applied to classify remote sensing data into different land cover classes. But here our objective is to predict the entire range of an the hominid species Imaginus magnapedum (also known under the vernacular names of “bigfoot” and “sasquatch”). This species is so hard to find (at least by scientists) that its very existence is commonly denied! For more information about this controversy, see the article by Lozier, Aniello and Hickerson: Predicting the distribution of Sasquatch in western North America: anything goes with ecological niche modelling.

We want to find out

a. What the complete range of the species might be.
b. how good (general) our model can be by predicting from the Western US to the Eastern sub-species.
c. predict where in Mexico the creature is likely to occur.
d. How climate change might affect its distribution.

In this context, this type of analysis is often referred to as ‘species distribution modeling’ or ‘ecological niche modeling’. Here is a more in-depth discussion of this technique.

Data

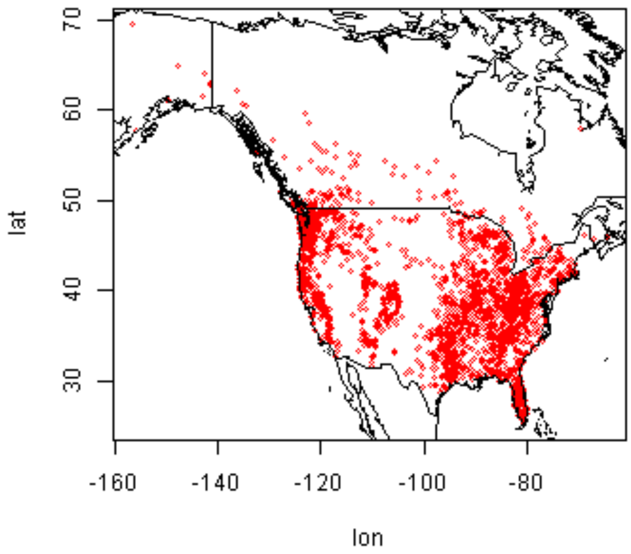
Observations

Here is the data for the reported observations, extracted from http://www.bfro.net/ in May 2012.

```
bf <- read.csv("data/bigfoot.csv")
dim(bf)
## [1] 3092    3
head(bf)
##         lon      lat Class
## 1 -142.9000 61.50000     A
## 2 -132.7982 55.18720     A
## 3 -132.8202 55.20350     A
## 4 -141.5667 62.93750     A
## 5 -149.7853 61.05950     A
## 6 -141.3165 62.77335     A
```

Plot the locations

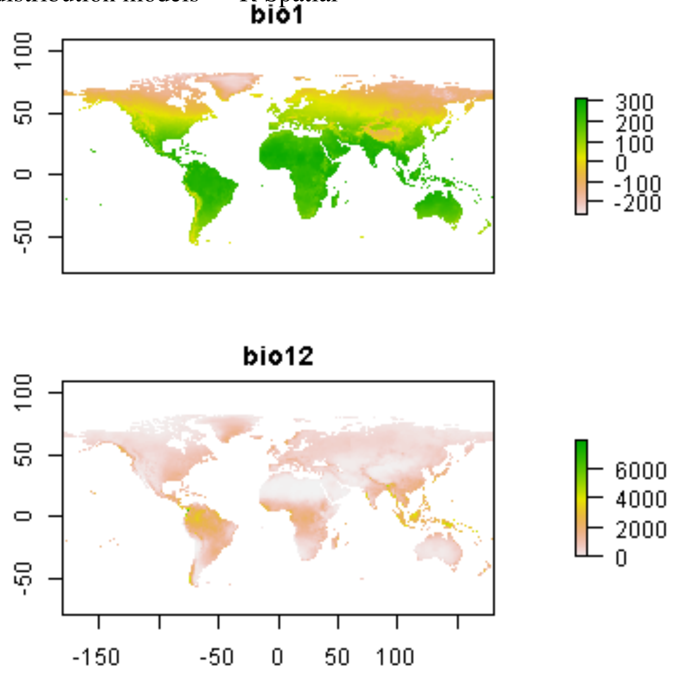
```
plot(bf[,1:2], cex=0.5, col='red')
library(maptools)
## Checking rgeos availability: TRUE
data(wrld_simpl)
plot(wrld_simpl, add=TRUE)
```



Predictors

Supervised classification often uses predictor data obtained from satellite remote sensing. But here, as is common in species distribution modeling, we use climate data. Specifically, we use 'bioclimatic variables', see: http://www.worldclim.org/bioclim

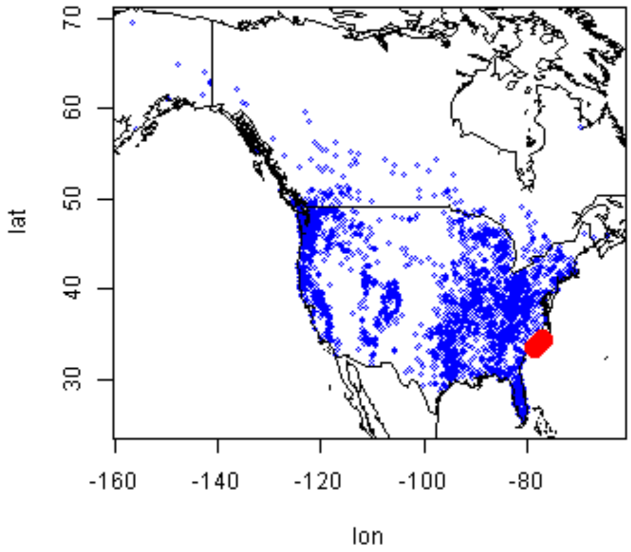
```
library(raster)
wc <- getData('worldclim', res=10, var='bio')
plot(wc[[c(1, 12)]], nr=2)
```



Now extract climate data for the locations of our observations. That is, get data about the climate that the species likes, apparently.

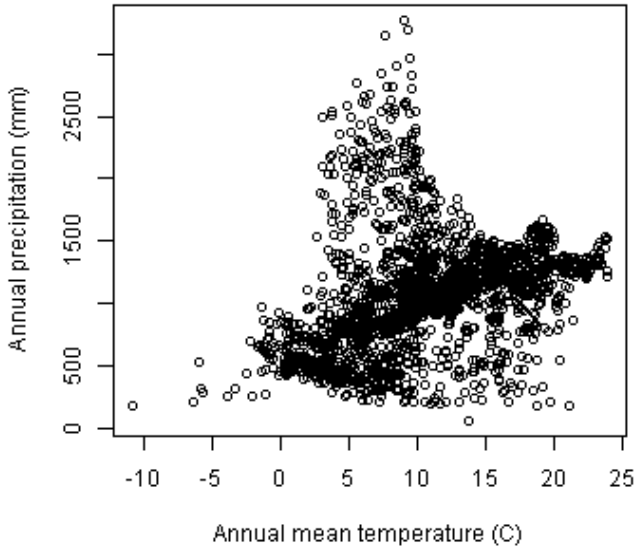
```
bfc <- extract(wc, bf[,1:2])
head(bfc)
##      bio1 bio2 bio3  bio4 bio5 bio6 bio7 bio8 bio9 bio10 bio11 bio12 bio13
## [1,]  -14  102   27  9672  174 -197  371   51  -11   108  -137   973   119
## [2,]   62   55   31  4136  157  -17  174   43   98   118    15  2602   385
## [3,]   62   55   31  4136  157  -17  174   43   98   118    15  2602   385
## [4,]  -57  125   23 15138  206 -332  538  127 -129   127  -256   282    67
## [5,]   10   80   25  8308  174 -140  314   66    5   119   -91   532    81
## [6,]  -59  128   23 14923  204 -334  538  122 -130   122  -255   322    75
##      bio14 bio15 bio16 bio17 bio18 bio19
## [1,]    43    30   332   156   290   210
## [2,]   128    33   953   407   556   721
## [3,]   128    33   953   407   556   721
## [4,]     6    81   163    22   163    27
## [5,]    22    41   215    72   159   117
## [6,]     8    79   183    28   183    32

# Any missing values?
i <- which(is.na(bfc[,1]))
i
## [1]  862 2667
plot(bf[,1:2], cex=0.5, col='blue')
plot(wrld_simpl, add=TRUE)
points(bf[i, ], pch=20, cex=3, col='red')
```



Here is a plot that illustrates a component of the ecological niche of our species of interest.

```
plot(bfc[ ,'bio1'] / 10, bfc[, 'bio12'], xlab='Annual mean temperature (C)',
        ylab='Annual precipitation (mm)')
```



Background data

Normally, we would build a model that would compare the values of the predictor variables as the locations where something was observed, with those at the locations where it was not. But we do not have data from a systematic survey that determined presence and absence. We have presence-only data. (determining absence would be very hard to do. It is here now, it is gone tomorrow).

The common trick to deal with this is to not model presence vs absence, but presence vs 'random expectation'. This random expectation (also referred to as background, or random-absence data) is what you would get if the species had no preference for any of the predictor variables (or other, perhaps correlated, variables that are not in the model).

There is not much point in taking absence data from very far away (tropical Africa or Antarctica). Typically they are taken from more or less the entire study area for which we have presences data.

```
library(dismo)
# extent of all points
e <- extent(SpatialPoints(bf[, 1:2]))
e
## class       : Extent
## xmin        : -156.75
## xmax        : -64.4627
## ymin        : 25.141
## ymax        : 69.5

# 5000 random samples (excluding NA cells) from extent e
set.seed(0)
bg <- sampleRandom(wc, 5000, ext=e)
dim(bg)
## [1] 5000   19
head(bg)
##      bio1 bio2 bio3  bio4 bio5 bio6 bio7 bio8 bio9 bio10 bio11 bio12 bio13
## [1,]  157  126   60  2935  262   55  207  124  191   197   122   379    88
## [2,]  -54  105   28  9244  142 -223  365   57  -62    68  -165   639    79
## [3,]  -57  104   20 14227  198 -317  515  106 -227   118  -247   473    71
## [4,]    1  119   24 12335  231 -251  482  138  -91   150  -168   844   104
## [5,]  208  169   44  7641  404   28  376  304  239   307   114   198    31
## [6,]  -89  111   23 12931  160 -316  476   78 -174    78  -248   476    76
##      bio14 bio15 bio16 bio17 bio18 bio19
## [1,]     0   100   225     2     4   222
## [2,]    28    30   226   101   219   138
## [3,]    17    46   197    55   194    59
## [4,]    34    33   301   128   291   137
## [5,]     2    50    73    11    52    62
## [6,]    25    40   193    79   193    82
```

Combine presence and background

```
d <- rbind(cbind(pa=1, bfc), cbind(pa=0, bg))
d <- data.frame(d)
dim(d)
## [1] 8092   20
```

Fit a model

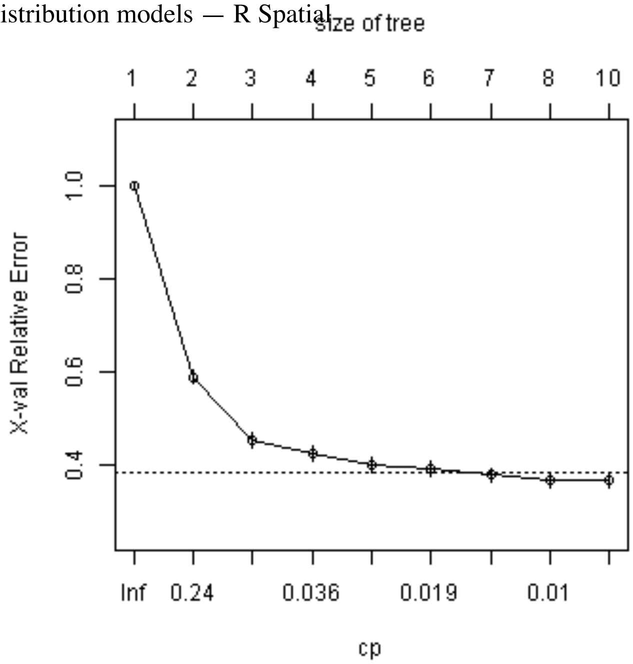
Now we have the data to fit a model. But I am going to split the data into East and West. Let's say I believe these are actually are different, albeit related, sub-species (The sasquatch is clearly darker and less hairy). I am principally interested in the western species.

```
de <- d[bf[,1] > -102, ]
dw <- d[bf[,1] <= -102, ]
```

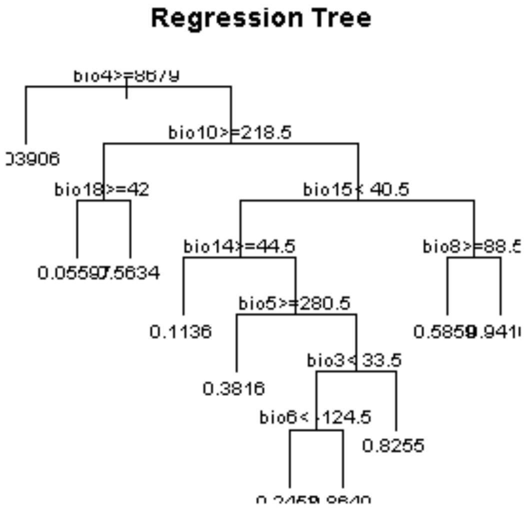
CART

Let’s first look at a CART.

```
library(rpart)
cart <- rpart(pa~., data=dw)
printcp(cart)
##
## Regression tree:
## rpart(formula = pa ~ ., data = dw)
##
## Variables actually used in tree construction:
## [1] bio10 bio14 bio15 bio18 bio3  bio4  bio5  bio6  bio8
##
## Root node error: 762.45/3246 = 0.23489
##
## n= 3246
##
##         CP nsplit rel error  xerror     xstd
## 1 0.410197      0   1.00000 1.00048 0.008909
## 2 0.137588      1   0.58980 0.59041 0.014191
## 3 0.044259      2   0.45222 0.45474 0.016586
## 4 0.029121      3   0.40796 0.42701 0.016572
## 5 0.018954      4   0.37884 0.40239 0.016586
## 6 0.018324      5   0.35988 0.39294 0.016337
## 7 0.010113      6   0.34156 0.37926 0.015777
## 8 0.010008      7   0.33144 0.36821 0.016014
## 9 0.010000      9   0.31143 0.36821 0.016014
plotcp(cart)
```



```
plot(cart, uniform=TRUE, main="Regression Tree")
text(cart, cex=.8)
```



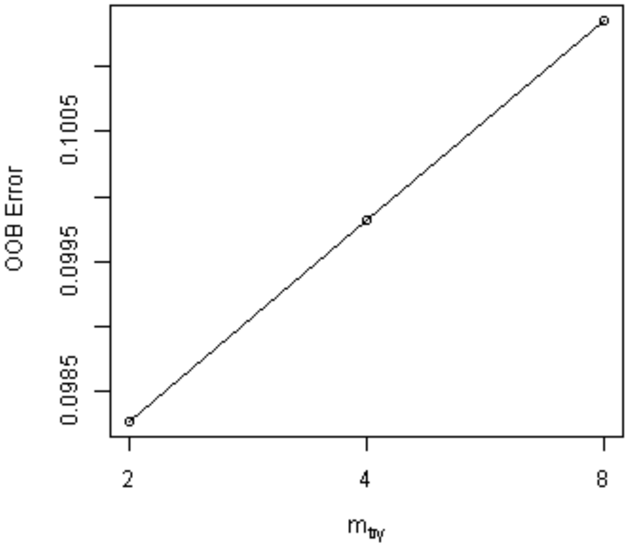
```
# text(cart, use.n=TRUE, all=TRUE, cex=.8)
```

Question 1: Describe the conditions under which you have the highest probability of finding our beloved species?

Random Forest

CART gives us a nice result to look at that can be easily interpreted (as you just illustrated with your answer to Question 1). But the approach suffers from high variance. Random Forest does not have that problem. Above, with CART, we use regression, let's do both regression and classification here. First classification

```
library(randomForest)
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
# create a factor to indicated that we want classification
fpa <- as.factor(dw[, 'pa'])
# first tune the randomForest
trf <- tuneRF(dw[, 2:ncol(dw)], fpa)
## mtry = 4  OOB error = 9.98%
## Searching left ...
## mtry = 2     OOB error = 9.83%
## 0.0154321 0.05
## Searching right ...
## mtry = 8     OOB error = 10.14%
## -0.0154321 0.05
```

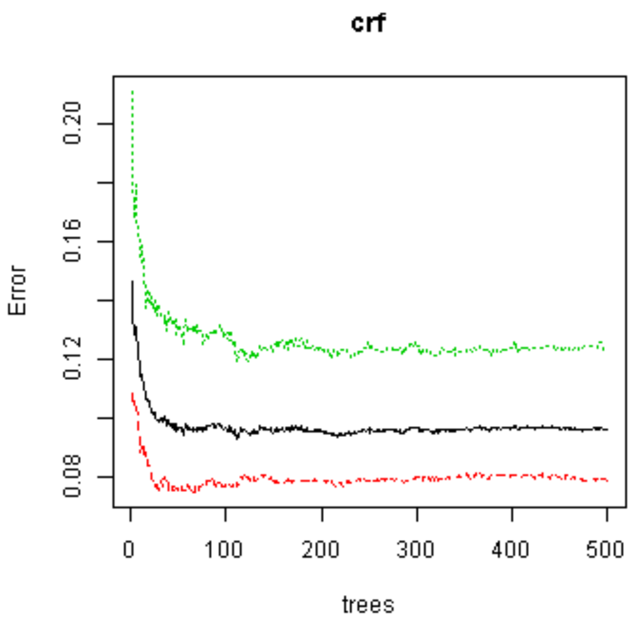


```
trf
##       mtry   OOBError
## 2.OOB    2 0.09827480
## 4.OOB    4 0.09981516
## 8.OOB    8 0.10135551
mt <- trf[which.min(trf[,2]), 1]
mt
## [1] 2
```

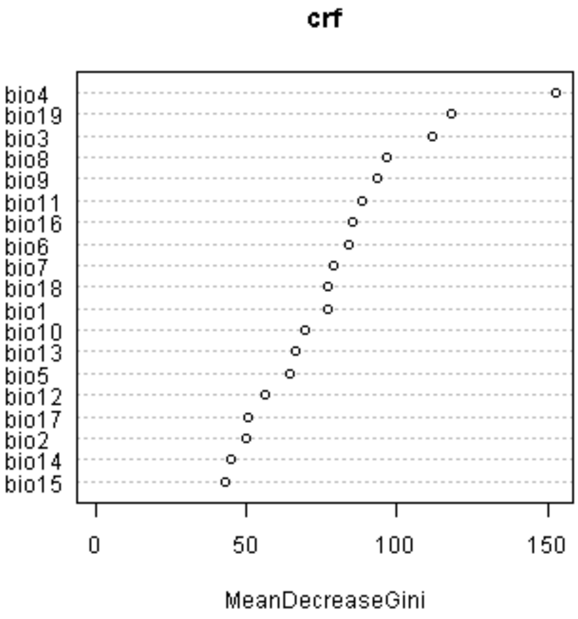
Question 2: What did tuneRF help us find? What does the values of mt represent? Could you refine this number?

Not fit the RandomForest model

```
crf <- randomForest(dw[, 2:ncol(dw)], fpa, mtry=mt)
crf
##
## Call:
##  randomForest(x = dw[, 2:ncol(dw)], y = fpa, mtry = mt)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 9.61%
## Confusion matrix:
##      0    1 class.error
## 0 1862  160  0.07912957
## 1  152 1072  0.12418301
plot(crf)
```

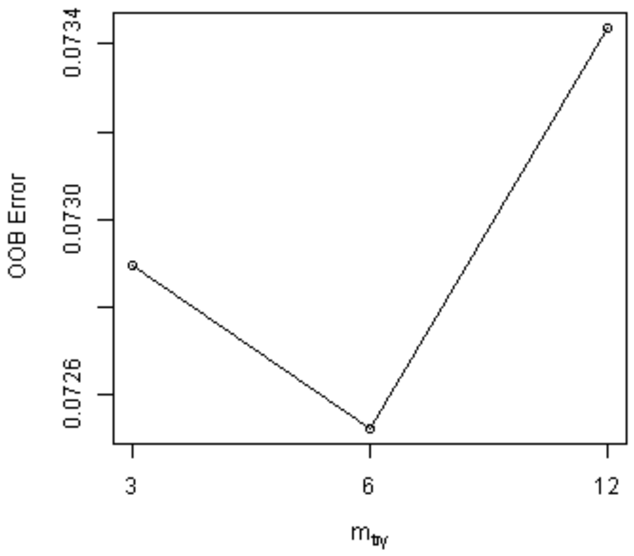


```
importance(crf)
##       MeanDecreaseGini
## bio1          77.02931
## bio2          49.84843
## bio3         111.61682
## bio4         152.29184
## bio5          64.67861
## bio6          84.04859
## bio7          78.60878
## bio8          96.28920
## bio9          93.28132
## bio10         69.35838
## bio11         88.06191
## bio12         56.16043
## bio13         66.38178
## bio14         45.16282
## bio15         42.93564
## bio16         85.07335
## bio17         50.51386
## bio18         77.05089
## bio19        118.24752
varImpPlot(crf)
```



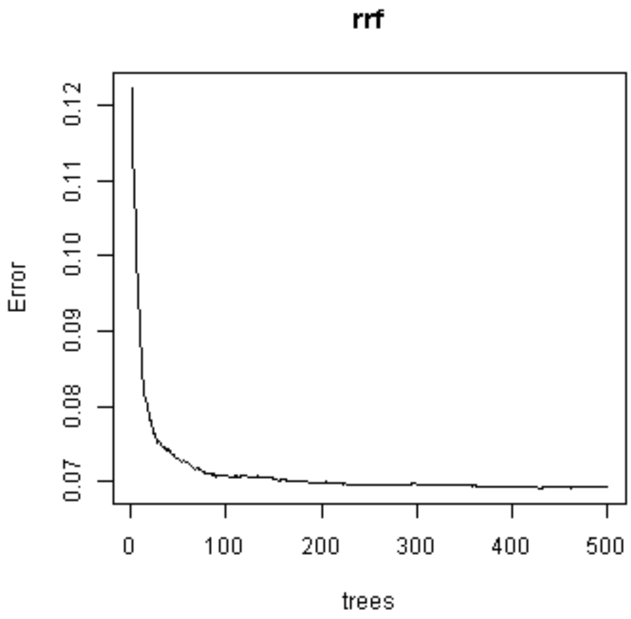
Now regression

```
library(randomForest)
trf <- tuneRF(dw[, 2:ncol(dw)], dw[, 'pa'])
## Warning in randomForest.default(x, y, mtry = mtryStart, ntree = ntreeTry, :
## The response has five or fewer unique values. Are you sure you want to do
## regression?
## mtry = 6  OOB error = 0.07252541
## Searching left ...
## Warning in randomForest.default(x, y, mtry = mtryCur, ntree = ntreeTry, :
## The response has five or fewer unique values. Are you sure you want to do
## regression?
## mtry = 3     OOB error = 0.07289613
## -0.005111565 0.05
## Searching right ...
## Warning in randomForest.default(x, y, mtry = mtryCur, ntree = ntreeTry, :
## The response has five or fewer unique values. Are you sure you want to do
## regression?
## mtry = 12    OOB error = 0.07343689
## -0.01256779 0.05
```

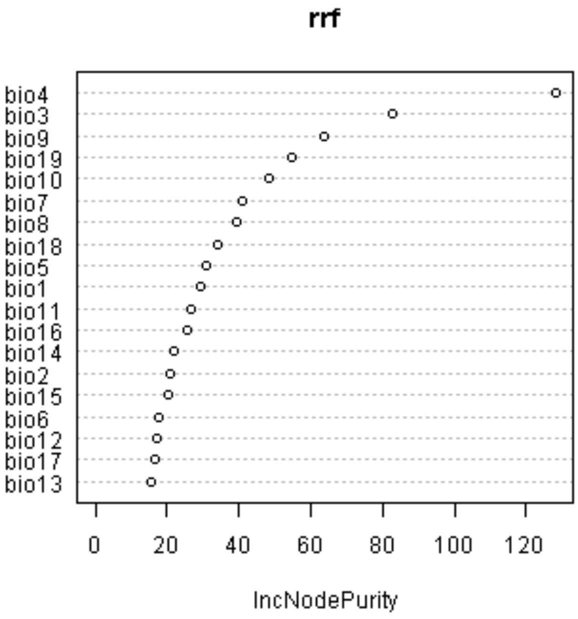


```
trf
##    mtry   OOBError
## 3     3 0.07289613
## 6     6 0.07252541
## 12   12 0.07343689
mt <- trf[which.min(trf[,2]), 1]
mt
## [1] 6

rrf <- randomForest(dw[, 2:ncol(d)], dw[, 'pa'], mtry=mt)
## Warning in randomForest.default(dw[, 2:ncol(d)], dw[, "pa"], mtry = mt):
## The response has five or fewer unique values. Are you sure you want to do
## regression?
rrf
##
## Call:
##  randomForest(x = dw[, 2:ncol(d)], y = dw[, "pa"], mtry = mt)
##                 Type of random forest: regression
##                       Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 0.06921956
##                     % Var explained: 70.53
plot(rrf)
```

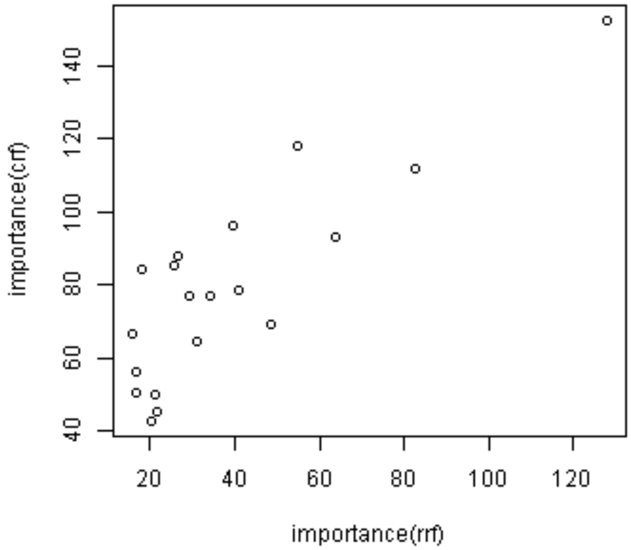


```
importance(rrf)
##       IncNodePurity
## bio1       29.12754
## bio2       21.05640
## bio3       82.51696
## bio4      128.04208
## bio5       30.85719
## bio6       17.88161
## bio7       40.69776
## bio8       39.51599
## bio9       63.80623
## bio10      48.59990
## bio11      26.71826
## bio12      16.89843
## bio13      15.58551
## bio14      21.69890
## bio15      20.43420
## bio16      25.79795
## bio17      16.56788
## bio18      34.24914
## bio19      54.89421
varImpPlot(rrf)
```



Question 3: Please compare/contrast the two approaches. What does the plot below tells us in this regard?

```
plot(importance(rrf), importance(crf))
```



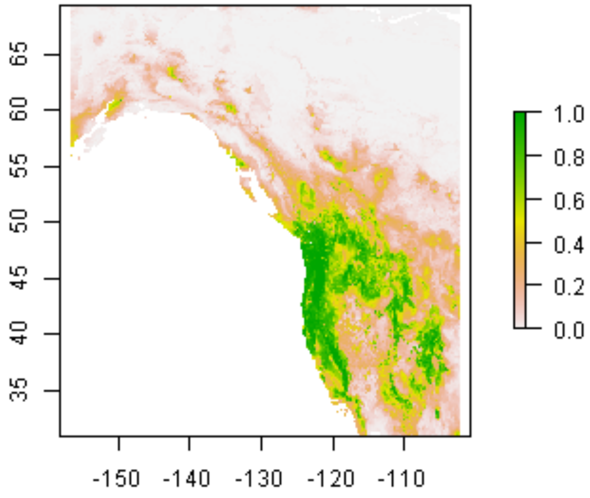
Predict

We can use the model to make predictions to any other place for which we have values for the predictor variables. Our climate data is global so we could find suitable places for bigfoot in Australia. At first I only want to predict to our study region, which I define as follows:

```
# Extent of the western points
ew <- extent(SpatialPoints(bf[bf[,1] <= -102, 1:2]))
ew
## class       : Extent
## xmin        : -156.75
## xmax        : -102.3881
## ymin        : 30.77722
## ymax        : 69.5
```

Regression

```
rp <- predict(wc, rrf, ext=ew)
plot(rp)
```

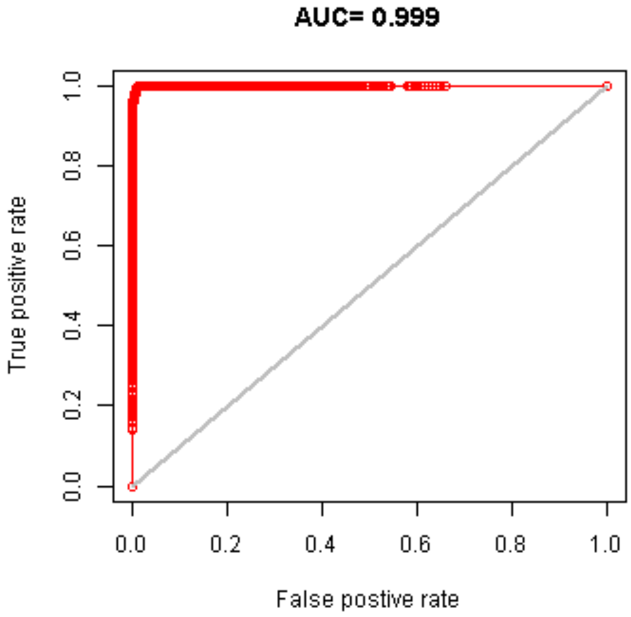


Note that the regression predictions are well-behaved, in the sense that they are between 0 and 1. However, they are continuous within that range, and if you wanted presence/absence, you would need a threshold. To get the optimal threshold, you would normally have a hold out data set, but here I used the training data for simplicity.

```
eva <- evaluate(dw[dw$pa==1, ], dw[dw$pa==0, ], rrf)
eva
## class          : ModelEvaluation
## n presences    : 1224
## n absences     : 2022
## AUC            : 0.9994917
## cor            : 0.9663393
## max TPR+TNR at : 0.4864567
```

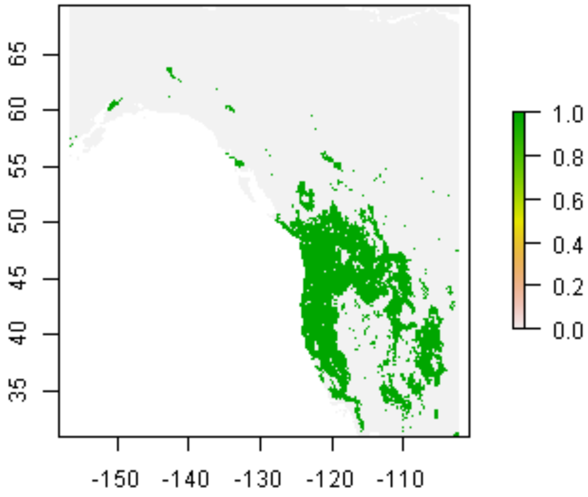
We can make a ROC plot

```
plot(eva, 'ROC')
```



Find a threshold and plot the preduction.

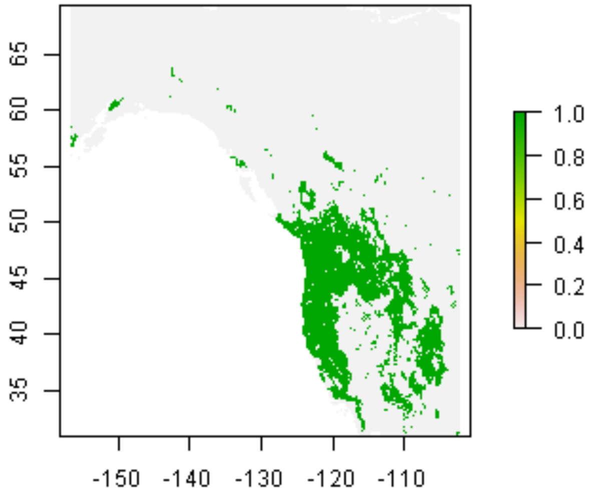
```
tr <- threshold(eva)
tr
##                 kappa spec_sens no_omission prevalence equal_sens_spec
## thresholds 0.4864567 0.4864567   0.4221905   0.377047         0.53297
##            sensitivity
## thresholds   0.7226077
plot(rp > tr[1, 'spec_sens'])
```



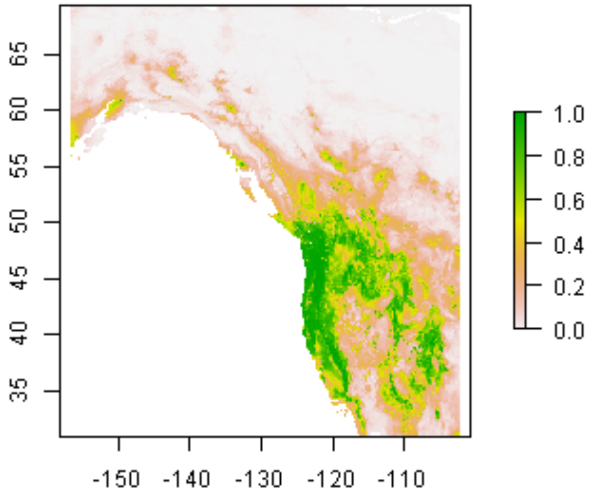
Classification

Now using the classification Random Forest model

```
rc <- predict(wc, crf, ext=ew)
plot(rc)
```



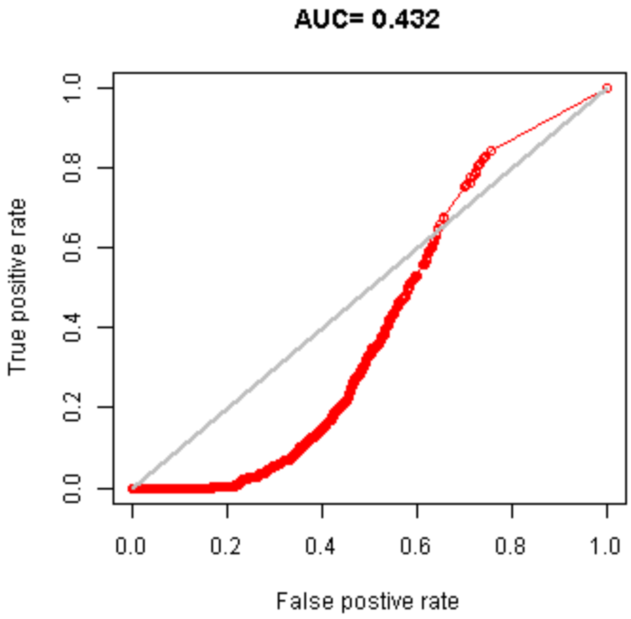
```
# you can also get probabilities
rc2 <- predict(wc, crf, ext=ew, type='prob', index=2)
plot(rc2)
```



Extrapolation

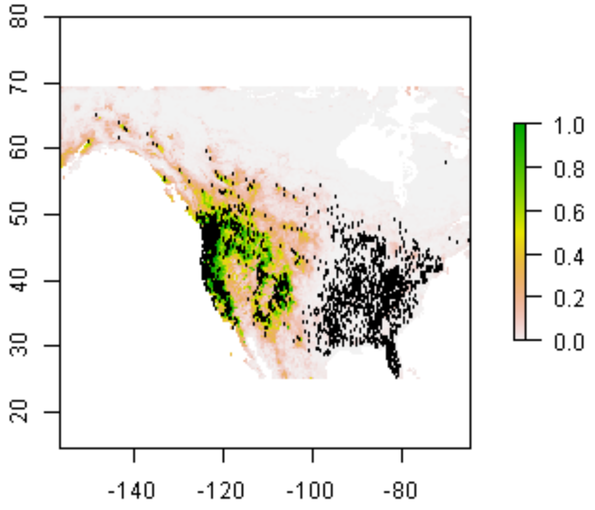
Now, let's see if our model is general enough to predict the distribution of the Eastern species.

```
de <- na.omit(de)
eva2 <- evaluate(de[de$pa==1, ], de[de$pa==0, ], rrf)
eva2
## class          : ModelEvaluation
## n presences    : 1866
## n absences     : 2978
## AUC            : 0.4315988
## cor            : -0.2595929
## max TPR+TNR at : 3e-04
plot(eva2, 'ROC')
```



Question 4: What does the AUC value / ROC plot suggest?

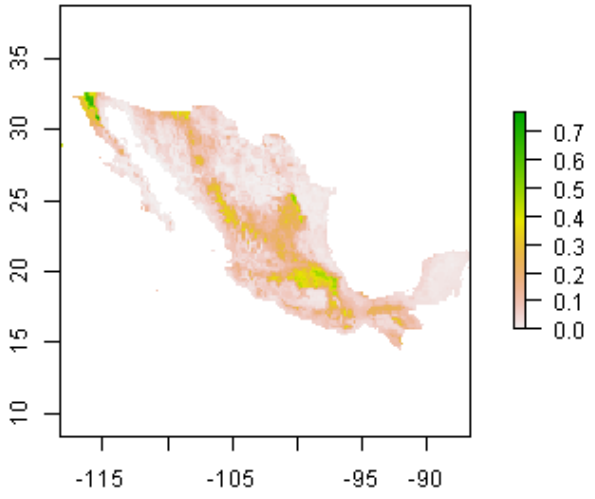
```
eus <- extent(SpatialPoints(bf[, 1:2]))
eus
## class       : Extent
## xmin        : -156.75
## xmax        : -64.4627
## ymin        : 25.141
## ymax        : 69.5
rcusa <- predict(wc, rrf, ext=eus)
plot(rcusa)
points(bf[,1:2], cex=.25)
```



Question 5: Why would extrapolation be so poorly?

An important question in the biogeography of the western species is why it does not occur in Mexico. Or if it does (or did before it was extirpated), where would that be?

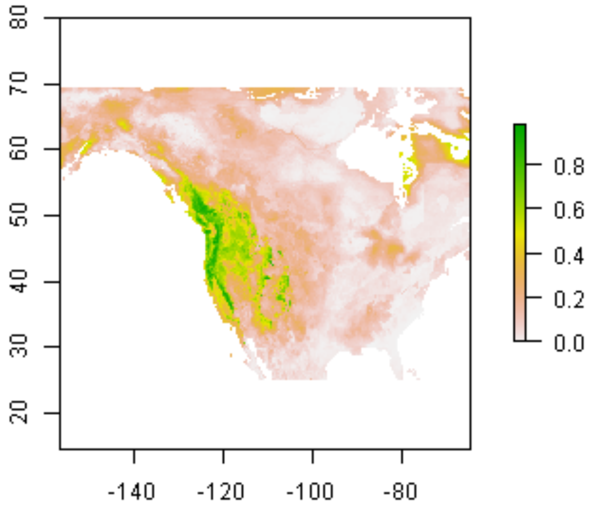
```
mex <- getData('GADM', country='MEX', level=1)
pm <- predict(wc, rrf, ext=mex)
pm <- mask(pm, mex)
plot(pm)
```



Question 6: Where in Mexico are you most likely to encounter western bigfoot?

We can also estimate range shifts due to climate change

```
fut <- getData('CMIP5', res=10, var='bio', rcp=85, model='AC', year=70)
names(fut)
##  [1] "ac85bi701"  "ac85bi702"  "ac85bi703"  "ac85bi704"  "ac85bi705"
##  [6] "ac85bi706"  "ac85bi707"  "ac85bi708"  "ac85bi709"  "ac85bi7010"
## [11] "ac85bi7011" "ac85bi7012" "ac85bi7013" "ac85bi7014" "ac85bi7015"
## [16] "ac85bi7016" "ac85bi7017" "ac85bi7018" "ac85bi7019"
names(wc)
##  [1] "bio1"  "bio2"  "bio3"  "bio4"  "bio5"  "bio6"  "bio7"  "bio8"
##  [9] "bio9"  "bio10" "bio11" "bio12" "bio13" "bio14" "bio15" "bio16"
## [17] "bio17" "bio18" "bio19"
names(fut) <- names(wc)
futusa <- predict(fut, rrf, ext=eus, progress='window')
## Loading required namespace: tcltk
plot(futusa)
```



Question 7: Make a map to show where conditions are improving for western bigfoot, and where they are not. Is the species headed toward extinction?

Further reading

More on Species distribution modeling with R; and on the use of boosted regression trees in the same context.