INSTITUTO
VENEZOLANO DE
INVESTIGACIONES
CIENTIFICAS

Material didáctico sobre

MODELOS DE NICHO Y DISTRIBUCIÓN DE ESPECIES

Sesión práctica sobre el algoritmo Bioclim en *R*

José Rafael Ferrer Paris

Laboratorío de Ecología Espacial, Centro de Estudios Botánicos y Agroforestales
Instituto Venezolano de Investigaciones Científicas
Maracaibo, estado Zulia

Centro de Ecología, IVIC
Altos de Pipe, 23 al 31 de octubre 2017

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para Educación Universitaria, Ciencia y Tecnología

1817 - 2017 ZAMORA UNIÓN CÍVICO MILITAR

Documento desarrollado por J. R. Ferrer Paris para el **Curso especial teórico - práctico presencial ECO-289: Modelos de Nicho y Distribución de Especies** del *Postgrado de Ecología* del Centro de Estudios Avanzados del Instituto Venezolano de Investigaciones Científicas (IVIC). Versión pública disponible en http://dx.doi.org/10.6084/m9.figshare.7891052. Disponible libremente según los términos de la licencia de "Creative Commons Reconocimiento 4.0 Internacional".

http://creativecommons.org/licenses/by-sa/4.0/

Este documento es generado por medio de las funciones de `Sweave` desde una sesión de *R* (R Development Core Team, 2011), por tanto todas las tablas y figuras se generan y actualizan automáticamente a partir de los datos suministrados. Para acceso al código fuente en *R* y los archivos de datos contacte al autor. Dentro de la sesión de *R* utilizamos los paquetes *pROC* (Robin *et al.*, 2011); *dismo* (Hijmans *et al.*, 2012); *raster* (Hijmans & van Etten, 2012); *sp* (Pebesma & Bivand, 2005); *gdata* (Warnes *et al.*, 2014).

1. Leer datos de distribución de especies

Para este ejemplo trabajamos con los datos de distribución de una especie de escarabajo coprófago en el Neotrópico. Usamos la *Base de datos del género Eurysternus (Escarabajos coprófagos) de America* (Camero & Lobo, 2012).[1]

Los datos provienen de una revisión taxonómica basada en ejemplares de más de 50 museos y colecciones entomológicas de América y Europa. Descargamos el archivo en formato `Excel` y lo abrimos con la función `read.xls` del paquete `gdata`:

```
> require(gdata)
> dir.spp <- "/home/jferrer/Dropbox/NeoMapas/data/Eurysternus"
> if (!exists("Eury")) {
+   Eury <- read.xls(sprintf("%s/Eurysternus Data Base mayo 2012.xls",dir.spp))
+ }
> head(Eury)
```

```
  Specimen_ID Stored_In                        Species_name   Country   Province
1 WSD00007284      CMNC Eurysternus aeneus G\xe9nier, 2009 ARGENTINA CORRIENTES
2 WSD00007285      CMNC Eurysternus aeneus G\xe9nier, 2009 ARGENTINA CORRIENTES
3 WSD00007286      CMNC Eurysternus aeneus G\xe9nier, 2009 ARGENTINA CORRIENTES
4 WSD00007287      CMNC Eurysternus aeneus G\xe9nier, 2009 ARGENTINA CORRIENTES
5 WSD00007288      CMNC Eurysternus aeneus G\xe9nier, 2009 ARGENTINA CORRIENTES
6 WSD00007289      CMNC Eurysternus aeneus G\xe9nier, 2009 ARGENTINA CORRIENTES
      Locality      Sex  Latitude Longitude Elevation   Collector Determined_by
1 Puerto Valle specimen -27.61667 -56.46833           Mart\xednez  F. G\xe9nier
2 Puerto Valle specimen -27.61667 -56.46833           Mart\xednez  F. G\xe9nier
3 Puerto Valle specimen -27.61667 -56.46833           Mart\xednez  F. G\xe9nier
```

[1] Disponible en http://www.biogeografia.org/biogeografia/index.php/recent-projects/145-base-de-datos-de-eurysternus-de-america, consultado en octubre 2017

```
4 Puerto Valle specimen -27.61667 -56.46833          Mart\xednez  F. G\xe9nier
5 Puerto Valle specimen -27.61667 -56.46833          Mart\xednez  F. G\xe9nier
6 Puerto Valle specimen -27.61667 -56.46833          Mart\xednez  F. G\xe9nier
```

Existen columnas con información de País, estado y localidad, y las coordenadas, pero no se indica incertidumbre en las coordenadas ni el origen de la georreferencia. Aparentemente la precisión de la georeferencia varía entre un minuto y un segundo, lo cual puede introducir errores entre 30 metros y 3 km de desplazamiento lineal, y que representa hasta 28,3 km^2 de incertidumbre en la georeferencia.[2]

Usamos el paquete `sp` para convertir los datos en un objeto espacial. Para ello determinamos las columnas que contienen la información de las coordenadas geográficas con la función `coordinates`. Las columnas de coordenadas no pueden contener valores nulos o vacíos, por tanto aplicamos un filtro con las funciones `subset` y `is.na`. Adicionalmente especificamos la información de la proyección geográfica en el formato `proj4` usando la función `proj4string`:

```
> require(sp)
> datos.spp <- subset(Eury,!is.na(Longitude) & Longitude <0)
> coordinates(datos.spp) <- c("Longitude","Latitude")
> proj4string(datos.spp) <- "+datum=WGS84 +proj=longlat"
```

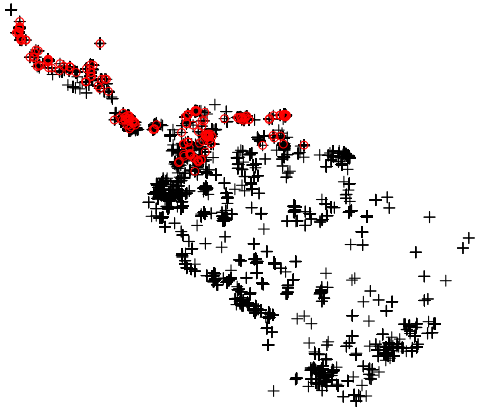
Extraemos los datos de presencia (`pres`) de la especie *Eurysternus mexicanus* Harold, 1869, y removemos los casos con coordenadas repetidas.

```
> pres <- subset(datos.spp,Species_name %in% "Eurysternus mexicanus Harold, 1869")
> pres <- subset(pres,!duplicated(coordinates(pres)))
```

[2] Medimos el área de incertidumbre como $\pi * r^2$, en donde r es el error de desplazamiento

Visualizamos la distribución de los puntos de distribución del género (cruces) y la ubicación de los datos de presencia de la especie seleccionada (círculos rojos).

```
> plot(datos.spp)
> points(pres,col=2)
```



José R. Ferrer Paris
Centro de Estudios Botánicos y Agroforestales

2. Leer datos de coberturas espaciales

Primero definimos la carpeta en la cual tenemos los datos de variables ambientales a utilizar. En este ejemplo usamos los datos descargados de WorldClim (Hijmans *et al.*, 2005) en una resolución relativamente gruesa (5 minutos de grados geográficos[3]).

```
> dir.mapas <- "/opt/gisdata/clima/WC1.4/05.0m/"
```

Verificamos que estén todas las capas en el formato adecuado (usamos aquí la versión en formato BIL):

```
> dir(dir.mapas,".bil$")

 [1] "bio10.bil" "bio11.bil" "bio12.bil" "bio13.bil" "bio14.bil" "bio15.bil"
 [7] "bio16.bil" "bio17.bil" "bio18.bil" "bio19.bil" "bio1.bil"  "bio2.bil"
[13] "bio3.bil"  "bio4.bil"  "bio5.bil"  "bio6.bil"  "bio7.bil"  "bio8.bil"
[19] "bio9.bil"
```

Para leer las capas en la sesión de *R* usamos la función `stack` del paquete `raster`

```
> require(raster)
> vars <- stack(dir(dir.mapas,".bil$",full.name=T))
```

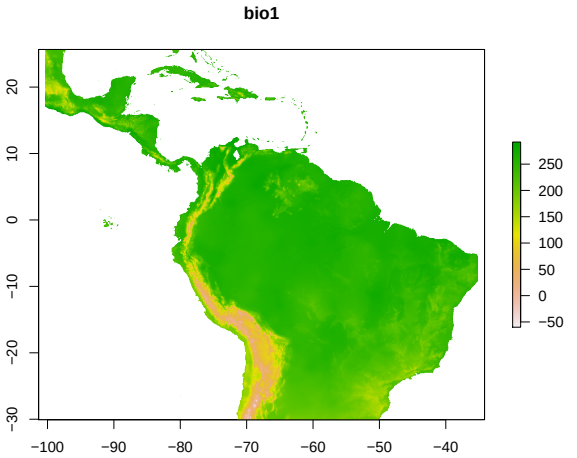
[3]Es recomendable usar una resolución similar o menor (más gruesa) que la incertidumbre de la georeferenciación (en este caso 29 km^2). Una resolución de 5 min implica que las celdas tienen un área de alrededor de 80 u 85 km^2 en regiones tropicales, mientras que la resolución de 2.5 min implica celdas de entre 18 y 21 km^2.

Cortamos la capa a la región del Neotrópico según los datos de distribución del género. Utilizamos la función `crop`:

```
> vars.NT <- crop(vars,datos.spp)
```

Veamos el resultado para una variable (bio1):

```
> plot(vars.NT,"bio1")
```



3. Ajustar modelo de Envoltorios Bioclimáticos o "Bioclim"

Bioclim define "envoltorios climáticos" que representan los límites o umbrales de la distribución de una especie (Booth *et al.*, 2014).

Ajustamos un modelo de *Bioclim* con la función del mismo nombre del paquete `dismo`. Empezamos con un modelo sencillo con dos variables que representan la temperatura promedio y la precipitación total anual. Guardamos el resultado en un objeto llamado `mdl01`.

```
> require(dismo)
> mdl01 <- bioclim(stack(vars.NT,layers=c("bio1","bio12")),
+                  pres)
```

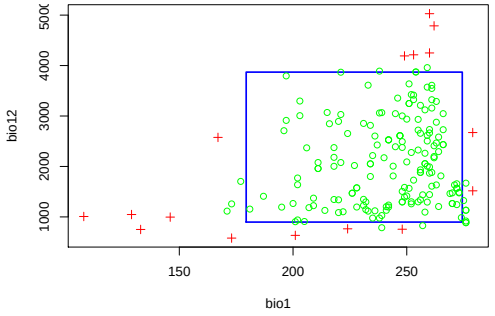
La descripción del resultado no es muy informativa

```
> mdl01

class    : Bioclim

variables: bio1 bio12


presence points: 191
   bio1 bio12
1   234  2052
2   248  1506
3   234  2052
4   253  1729
5   250  1415
6   248  3002
7   251  3637
8   202  1771
```

```
9    203   3297
10   221   3030
  (... ...  ...)
```

La función para visualizar el resultado muestra las combinaciones de valores de ambas variables en las localidades de presencia de la especie y muestra un rectángulo que representa el envoltorio climático con un porcentaje de los datos *para cada variable* (el valor por omisión es 90 %). Además muestra con círculos de color verde los datos que se encuentran dentro del envoltorio calculado *para todas las variables*, y con cruces rojas las localidades que se encuentran fuera de este envoltorio.[4]

```
> par(mar=c(5,4,0,0))
> plot(md101)
```



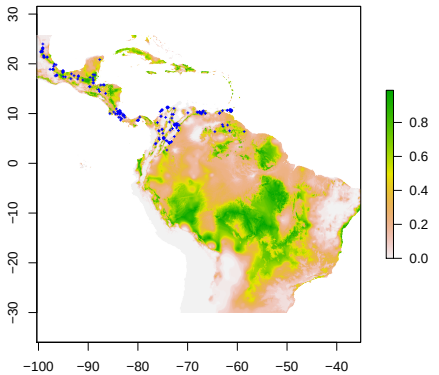
[4]En el caso de dos variables se observa una ligera discrepancia, posiblemente porque internamente se aplican dos umbrales diferentes o por alguna característica no documentada de la función.

Para obtener un mapa con la predicción del modelo (la expresión geográfica del envoltorio) utilizamos la función `predict` y combinamos el objeto con el modelo ajustado y las capas ambientales:

```
> prd01 <- predict(mdl01,vars.NT)
```

Visualizamos el resultado junto con los datos de presencia (cruces azules):

```
> plot(prd01)
> points(pres,col=4,pch=3,cex=.25)
```

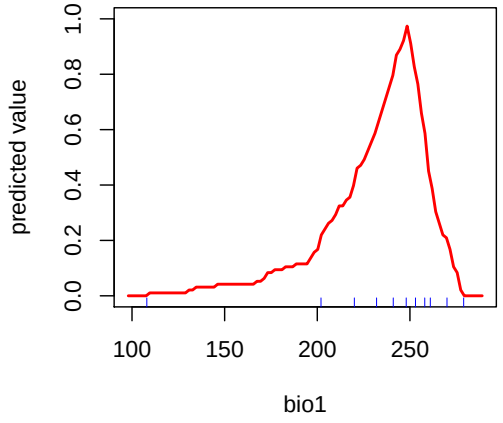


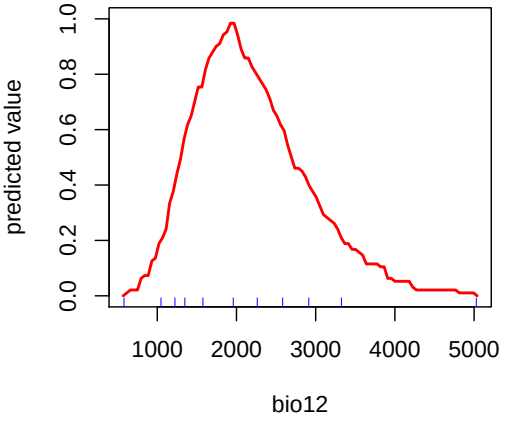
En esta implementación el resultado se expresa en un indice entre cero y uno que se basa en la ley de Liebig de factores limitantes (Hijmans *et al.*, 2012). En la documentación de la función (`?bioclim`) se describen los pasos para calcularlo. Se puede interpretar como una distancia en la escala de las variables ambientales transformadas a percentiles, y representa el doble de la mínima distancia de un punto al valor más extremo (mínimo o máximo) de cualquiera de las variables utilizadas. Por tanto un valor

cercano a uno se encuentra aproximadamente en el valor promedio de todas las variables (percentil 50), y un valor de 0.05 significa que la localidad se encuentra en la cola de valores extremos de al menos una variable. En otras palabras, **un valor de uno representa condiciones similares al promedio de las condiciones observadas en las localidades de presencia**, lo cual es interpretado como condiciones idóneas para la especie.

La función `response` muestra una curva de respuesta para cada variable, es decir el resultado del índice que se obtendría para cada variable si las demás variables se mantuvieran en un valor del percentil 50.

```
> response(md101)
```



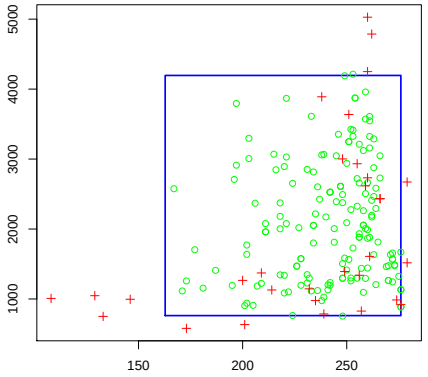


En este método, entre más variables coloquemos menor será el valor máximo del índice, pues más difícil será conseguir lugares con condiciones promedios de todos las variables. Probemos un modelo con todas las variables bioclimáticas:

```
> mdl02 <- bioclim(vars.NT,
+                  pres)
```

En los gráficos sólo podemos visualizar dos variables a la vez, `a` es la primera capa a comparar (eje x), `b` es la segunda capa (eje y), `p` es la proporción de datos a incluir en el envoltorio.

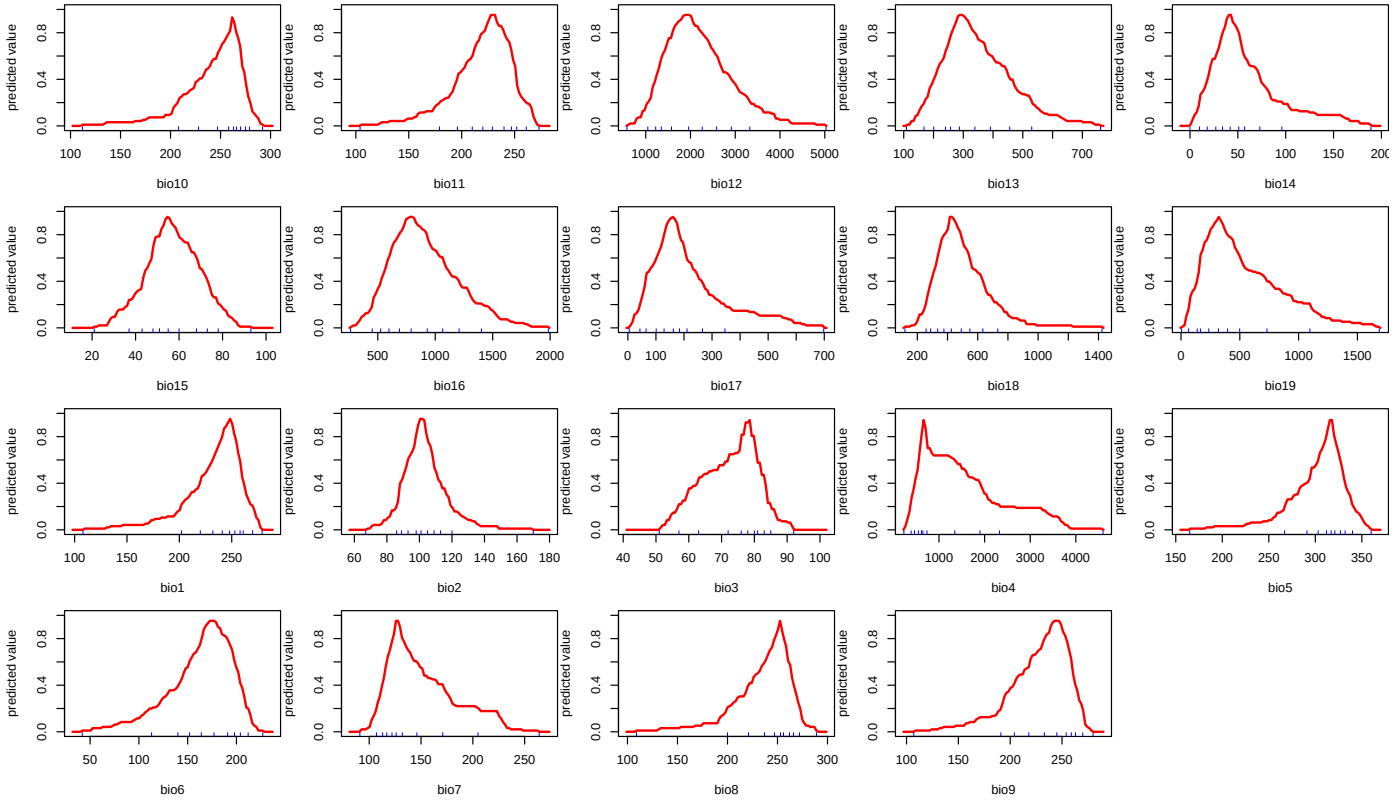
```
> plot(mdl02,a="bio1",b="bio12",p=.95)
```



Las cruces rojas dentro del recuadro representan localidades excluidas por alguna de las otras variables no visualizadas en el gráfico.

Podemos visualizar la respuesta a todas las variables:

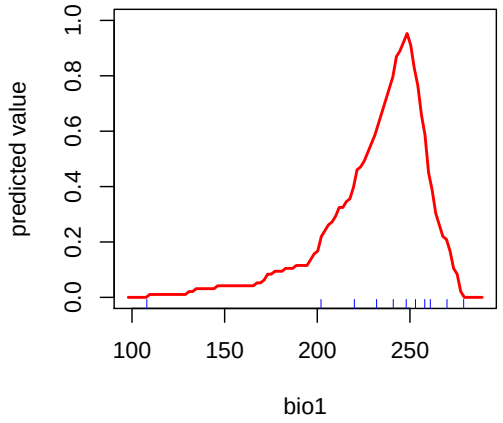
```
> par(mar=c(5,4,0,0))
> response(md102)
```

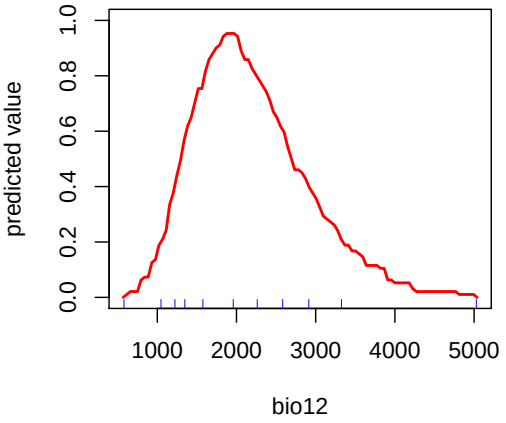


José R. Ferrer Paris
Centro de Estudios Botánicos y Agroforestales

Si nos enfocamos en las mismas variables que usamos en `mdl01` vemos que la respuesta es exactamente igual:

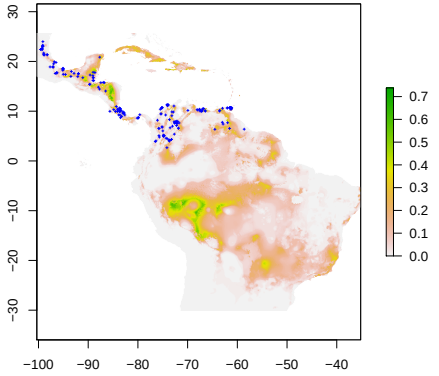
```
> response(mdl02,c("bio1","bio12"))
```





La respuesta a cada variable es igual, pero el resultado de combinar tantas variables cambia la predicción o expresión geográfica del resultado:

```
> plot(predict(mdl02,vars.NT))
> points(pres,col=4,pch=3,cex=.25)
```



4. Predicción y proyección a otras regiones

Es posible proyectar el modelo a una región diferente (o una época diferente) utilizando la función `predict` combinada con un conjunto de capas equivalentes a las utilizadas en el modelo.

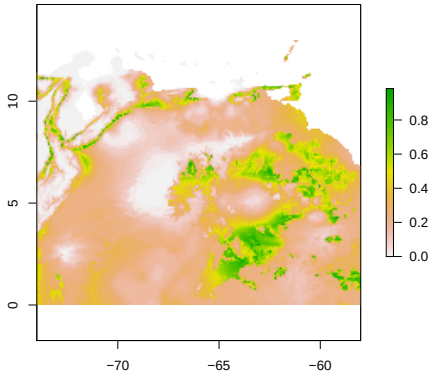
Primero hacemos un recorte de las capas al área de Venezuela (entre 58 y 74° de longitud oeste y entre 0 y 13° de latitud norte):

```
> vars.vz <- crop(vars,c(-74,-58,0,13))
```

Luego usamos el objeto con el modelo ajustado y las nuevas capas recortadas para la proyección:

```
> prd01.vz <- predict(mdl01,vars.vz)
```

```
> plot(prd01.vz)
```



5. Evaluación y comparación de modelos

Para evaluar modelos necesitamos tener localidades de evaluación con información sobre la "presencia" y "ausencia" de la especie.[5] Podemos aplicar diferentes estrategias para la evaluación, dependiendo del tipo de datos que tengamos y de los supuestos que hagamos sobre las ausencias.

5.1. Evaluación con datos externos o independientes

En el mejor de los casos contamos con dos conjuntos de datos independientes para calibrar y evaluar nuestro modelo. En este caso contamos con los datos de distribución continental que hemos descrito anteriormente, y tenemos datos del muestreo de *NeoMapas* de escarabajos coprófagos en Venezuela entre 2005 y 2010 (Ferrer-Paris *et al.*, 2013).

Primero ajustamos varios modelos con los datos de distribución neotropical:

```
> mdl01 <- bioclim(stack(vars.NT,layers=c("bio1","bio12")),
+                  pres)
> mdl02 <- bioclim(stack(vars.NT,layers=c("bio1","bio5","bio7","bio12","bio14")),
+                  pres)
> mdl03 <- bioclim(vars.NT,
+                  pres)
```

Luego, usamos los datos de NeoMapas para la evaluación. Los datos de *Eurysternus mexicanus* se encuentran en un archivo en formato `csv` que podemos leer con la función `read.csv`:

```
> dir.spp <- "/home/jferrer/Dropbox/NeoMapas/data/Eurysternus"
> datos.NM <- read.csv(sprintf("%s/Eurysternus_mexicanus.csv",dir.spp))
```

[5] En la práctica se trata de localidades de detección y localidades sin detecciones, pero donde no podemos determinar si las ausencias son reales.

```
> str(datos.NM)

'data.frame':        2697 obs. of  4 variables:
 $ X                   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ longitud            : num  -67.8 -67.8 -67.8 -67.8 -67.8 ...
 $ latitud             : num  5.06 5.06 5.06 5.06 5.06 ...
 $ Eurysternus.mexicanus: int  0 0 0 0 0 0 0 0 0 0 ...
```

El archivo contiene las coordenadas de las localidades de muestreo de escarabajos de NeoMapas y una columna que indica la detección (uno) o no detección (cero) de *Eurysternus mexicanus*.

Transformamos los datos a un objeto geográfico, tal y como hicimos con los datos de distribución neotropical:

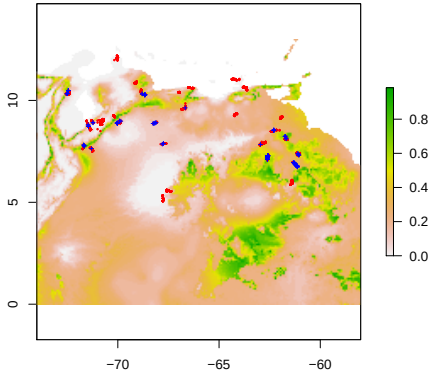
```
> coordinates(datos.NM) <- c("longitud","latitud")
> proj4string(datos.NM) <- "+datum=WGS84 +proj=longlat"
```

Ahora dividimos los datos en presencias (detecciones) y ausencias (no detecciones):

```
> pres.NM <- subset(datos.NM,Eurysternus.mexicanus==1)
> aus.NM <- subset(datos.NM,Eurysternus.mexicanus==0)
```

Visualizamos estos datos sobre la predicción del primer modelo en el área de Venezuela:

```
> prd01.vz <- predict(mdl01,vars.vz)
> plot(prd01.vz)
> points(aus.NM,col=2,pch=19,cex=.25)
> points(pres.NM,col=4,pch=3,cex=.45)
```



Para evaluar los modelos usamos la función `evaluate` con el objeto del modelo ajustado y los parámetros `p` para los datos de presencia del conjunto de evaluación, `a` para los datos de ausencia (aquí usamos localidades aleatorias o "pseudo-ausencias", `x` para las capas de variables ambientales:

```
> e01 <- evaluate(mdl01, p=pres.NM, a=aus.NM, x=vars.NT)
> e02 <- evaluate(mdl02, p=pres.NM, a=aus.NM, x=vars.NT)
> e03 <- evaluate(mdl03, p=pres.NM, a=aus.NM, x=vars.NT)
```

Consultamos el resultado de la evaluación del primer modelo:

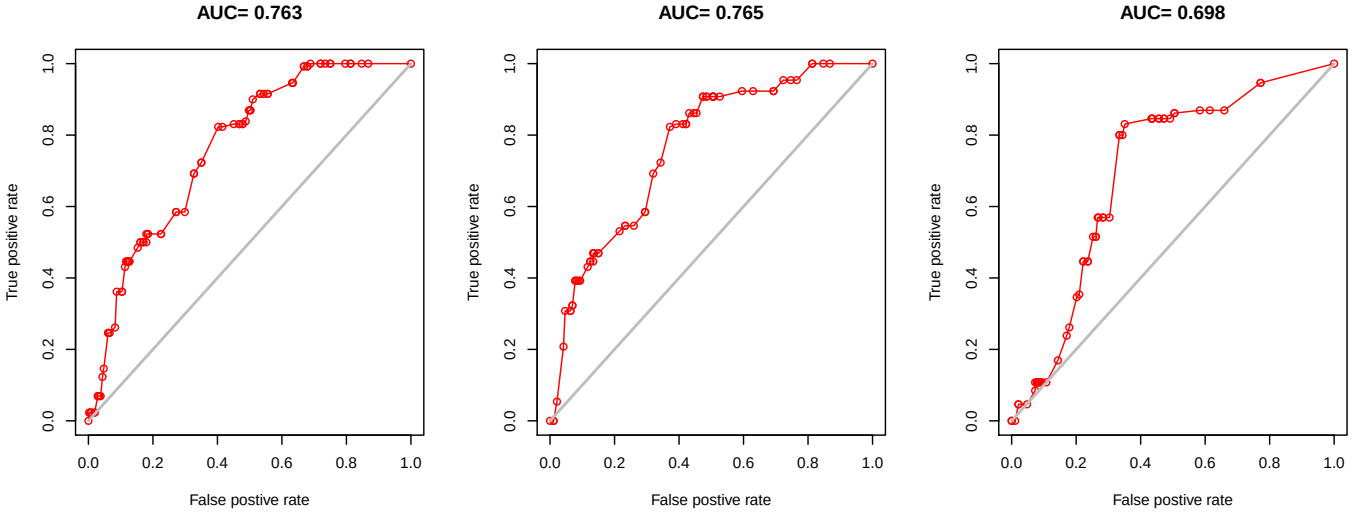
```
> e01

class           : ModelEvaluation
n presences     : 130
n absences      : 2567
AUC             : 0.7626577
cor             : 0.2037456
max TPR+TNR at  : 0.2407377
```

Y lo comparamos con el segundo modelo:

```
> e02

class           : ModelEvaluation
n presences     : 130
n absences      : 2567
AUC             : 0.7653726
cor             : 0.2258978
max TPR+TNR at  : 0.2407377
```

En este caso, y según el criterio de área bajo la curva (AUC) los modelos con menos variables explica mejor los datos obtenidos durante los muestreos de NeoMapas.

```
> layout(matrix(1:3,ncol=3))
> plot(e01, 'ROC')
> plot(e02, 'ROC')
> plot(e03, 'ROC')
```



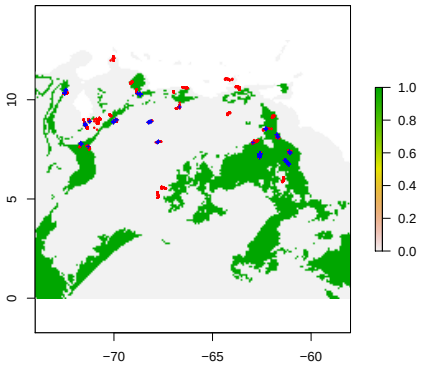
José R. Ferrer Paris
Centro de Estudios Botánicos y Agroforestales

Podemos calcular el punto de corte (umbral) óptimo según diferentes criterios con la función `threshold`:

```
> threshold(e02)

              kappa spec_sens no_omission prevalence equal_sens_spec
thresholds 0.565345 0.2407377  0.03131361 0.05225602       0.2878581
           sensitivity
thresholds   0.1517325
```

Usamos el criterio de maximizar la suma de tasas de verdaderos positivos y verdaderos negativos para escoger nuestro umbral y visualizar el resultado en el mapa:

```
> prd02.vz <- predict(mdl02,vars.vz)
> plot(prd02.vz > threshold(e02,"spec_sens"))
> points(aus.NM,col=2,pch=19,cex=.25)
> points(pres.NM,col=4,pch=3,cex=.45)
```



5.2. Partición de datos de presencia + puntos de muestreo del género

En muchos casos no contamos con datos externos para realizar la evaluación, pero conocemos los sitios en los que se han realizado muestreos de especies similares (probablemente con métodos de muestreo similares), y podemos suponer que estos sitios representan localidades de "ausencia" de la especie.

El primer paso consiste en crear una variable para separar los datos de presencia en dos conjuntos

```
> ss <- sample(c("CLB","EVL"),length(pres),replace=T,prob=c(0.75,0.25))
> calibracion <- subset(pres,ss %in% "CLB")
> evaluacion <- subset(pres,ss %in% "EVL")
```

Ajustamos varios modelos con los datos de calibración:

```
> mdl01 <- bioclim(stack(vars.NT,layers=c("bio1","bio12")),
+                  calibracion)
> mdl02 <- bioclim(stack(vars.NT,layers=c("bio1","bio5","bio7","bio12","bio14")),
+                  calibracion)
> mdl03 <- bioclim(vars.NT,
+                  calibracion)
```

Luego consideramos los datos de "ausencia" como aquellas localidades de colecta de escarabajos donde no se ha registrado la especie. Removemos los duplicados y las localidades que coinciden con los datos de presencia.

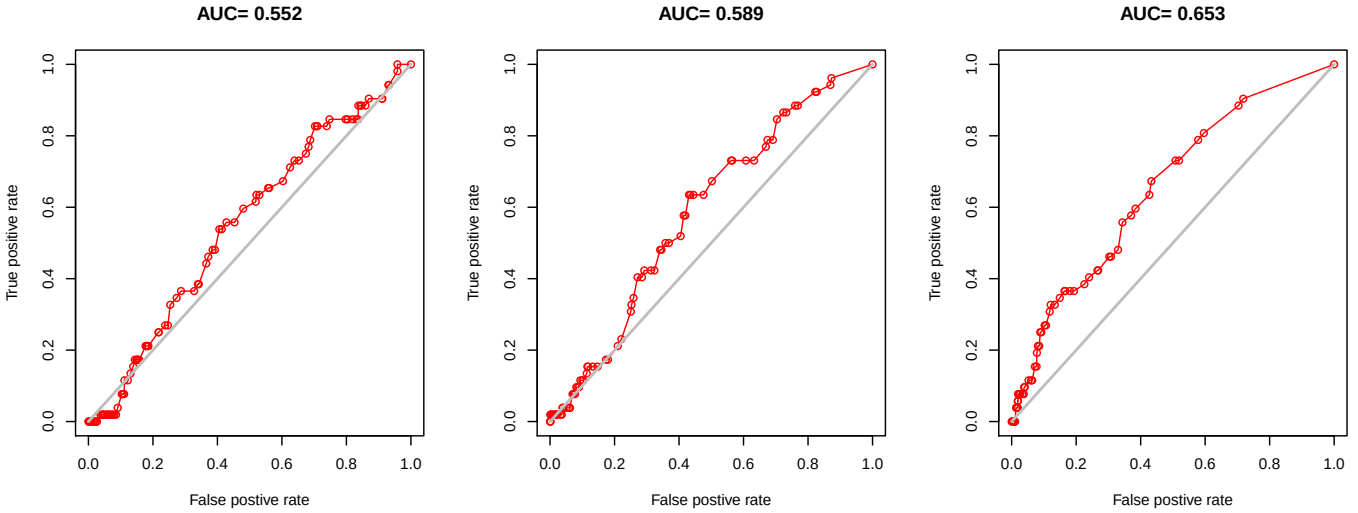
```
> aus <- subset(datos.spp,!Species_name %in% "Eurysternus mexicanus Harold, 1869")
> aus <- subset(aus,!duplicated(coordinates(aus)))
> aus <- subset(aus,is.na(over(aus,pres)$Specimen_ID))
```

En la función `evaluate` colocamos los datos de evaluación en el parámetro `p` y los datos de ausencias en el parámetro `a`:

```
> e01 <- evaluate(mdl01, p=evaluacion, a=aus, x=vars.NT)
> e02 <- evaluate(mdl02, p=evaluacion, a=aus, x=vars.NT)
> e03 <- evaluate(mdl03, p=evaluacion, a=aus, x=vars.NT)
```

En este caso, el desempeño del modelos con más variables es ligeramente superior a los otros aunque el valor de AUC es relativamente bajo. Debido a que la selección de los conjuntos de calibración y evaluación es aleatoria, estos valores pueden variar cada vez que se repita el procedimiento.

```
> layout(matrix(1:3,ncol=3))
> plot(e01, 'ROC')
> plot(e02, 'ROC')
> plot(e03, 'ROC')
```



5.3. Partición de datos de presencia + puntos aleatorios

Frecuentemente no contamos con datos independientes para la evaluación y los datos de colecta de otras especies no están disponibles o no son representativos de las ausencias de la especie de interés. En este caso, una opción para realizar la evaluación es combinar un subconjunto de datos de presencia con una muestra aleatorea de puntos que representarian "pseudo-ausencias" de la especie. Suponemos que estas "pseudo-ausencias" representan la variedad de condiciones ambientales disponibles en toda el área de estudio.

En este caso, el primer paso es igual al caso anterior:

```
> ss <- sample(c("CLB","EVL"),length(pres),replace=T,prob=c(0.75,0.25))
> calibracion <- subset(pres,ss %in% "CLB")
> evaluacion <- subset(pres,ss %in% "EVL")
> mdl01 <- bioclim(stack(vars.NT,layers=c("bio1","bio12")),
+                  calibracion)
> mdl02 <- bioclim(stack(vars.NT,layers=c("bio1","bio5","bio7","bio12","bio14")),
+                  calibracion)
> mdl03 <- bioclim(vars.NT,
+                  calibracion)
```

Luego creamos datos aleatorios a lo largo de las coberturas espaciales que representan condiciones disponibles para la especie:

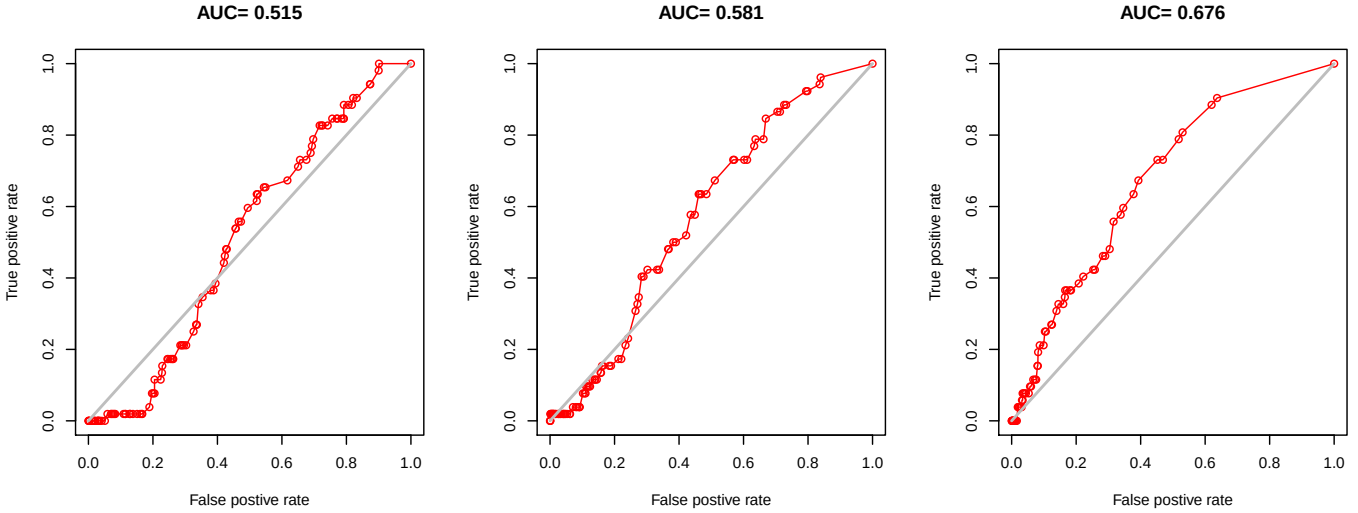
```
> pseudo.aus <- randomPoints(vars.NT, 1000)
```

En la función `evaluate` colocamos los datos de evaluación en el parámetro `p` y los datos de "pseudo-ausencias" en el parámetro `a`:

```
> e01 <- evaluate(mdl01, p=evaluacion, a=pseudo.aus, x=vars.NT)
> e02 <- evaluate(mdl02, p=evaluacion, a=pseudo.aus, x=vars.NT)
> e03 <- evaluate(mdl03, p=evaluacion, a=pseudo.aus, x=vars.NT)
```

En este caso, los resultados son similares al caso anterior, se favorece el modelo más complejo, pero el poder de discriminación es muy bajo.

```
> layout(matrix(1:3,ncol=3))
> plot(e01, 'ROC')
> plot(e02, 'ROC')
> plot(e03, 'ROC')
```



6. Para usuarios avanzados

6.1. Pruebas estadísticas para evaluación de modelos

La primera pregunta que nos podemos hacer es si un modelo tiene un ajuste significativo, o sea, si el modelo explica los datos observados mejor que una clasificación aleatoria. Hemos visto que en algunos de los ejemplos anteriores los valores de AUC fueron muy bajos.

Repasemos el caso del modelo ajustado con un subconjunto de datos de calibración y evaluado con con otro subconjunto de datos de prueba más los datos de muestreo de otras especies del género:

```
> e01 <- evaluate(mdl01, p=evaluacion, a=aus, x=vars.NT)
> e02 <- evaluate(mdl02, p=evaluacion, a=aus, x=vars.NT)
> e03 <- evaluate(mdl03, p=evaluacion, a=aus, x=vars.NT)
```

Una medida alternativa de desempeño predictivo es la correlación entre el valor de modelo y los valores de presencia y ausencia.

```
> data.frame(modelo=c("mdl01","mdl02","mdl03"),
+            AUC=signif(c(e01@auc,e02@auc,e03@auc),3),
+            correlación=signif(c(e01@cor,e02@cor,e03@cor),2),
+            p=signif(c(e01@pcor,e02@pcor,e03@pcor),2))

  modelo   AUC correlación       p
1  mdl01 0.552       0.022 0.49000
2  mdl02 0.589       0.058 0.06500
3  mdl03 0.653       0.120 0.00021
```

En este caso vemos que el modelo intermedio tiene un valor de AUC muy bajo y no tienen una correlación significativa con los datos de evaluación.[6]

Existen otras formas de probar si una curva de AUC es significativamente diferente de una clasificación aleatoria. Para ello usamos funciones del paquete `pROC` (Robin *et al.*, 2011). Para usar este paquete tenemos que crear dos vectores, uno con los datos observados (ceros y unos) y otro con las predicciones en las localidades conocidas.

```
> require(pROC)
> observado <- c(rep(1,length(evaluacion)),
+                rep(0,length(aus)))
> prd02 <- predict(mdl02,vars.NT)
> predicho <- c(extract(prd02,evaluacion),
+               extract(prd02,aus))
```

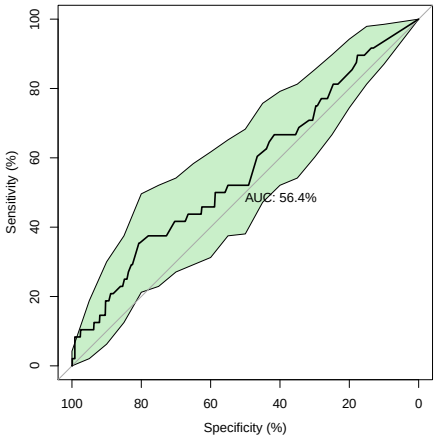
Usamos la función `roc` para construir una curva ROC ("Receiver Operating Characteristic") con intervalos de confianza.

```
> if (!exists("roc02.aus"))
+     roc02.aus <- roc(response=observado,predictor=predicho,percent=T,ci=T,of="se",sp=seq(0,100,5))
```

El paquete ofrece varias opciones para la visualización de la curva resultante. Si los intervalos de confianza se solapan con la diagonal del gráfico se puede decir que la curva no es significativa diferente de una clasificación aleatoria, o sea, el área bajo la curva no es significativamente diferente de 0,5.

```
> plot(roc02.aus, print.auc=T, auc.polygon=F, ci.type="shape",ci.col=rgb(.3,.8,.3,.3))
```

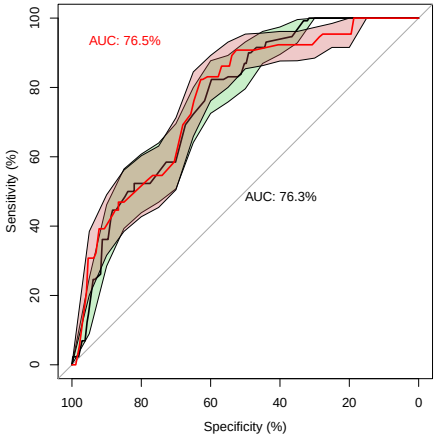
[6] Paradójicamente, este es el modelo que tiene mejor desempeño para predecir datos independientes. Estos resultados ponen en evidencia el riesgo de realizar evaluaciones con datos que no son independientes, y que pueden llevar a descartar modelos sencillos que realmente pueden tener un buen desempeño.



También podemos usar estas funciones para determinar si dos modelos son similares. En el ejemplo con evaluación externa de los datos vimos que el modelo de dos variables tenia una curva parecida a la del modelo con cinco variables, ahora podemos ver si las ligeras diferencias son significativas o no:

```
> observado <- c(rep(1,length(pres.NM)),
+                rep(0,length(aus.NM)))
> predicho <- c(extract(prd01.vz,pres.NM),
+               extract(prd01.vz,aus.NM))
> predicho2 <- c(extract(prd02.vz,pres.NM),
+               extract(prd02.vz,aus.NM))
> if (!exists("roc01.NM")) {
+     roc01.NM <- roc(response=observado,predictor=predicho,percent=T,ci=T,of="se",sp=seq(0,100,5))
+     roc02.NM <- roc(response=observado,predictor=predicho2,percent=T,ci=T,of="se",sp=seq(0,100,5))
+ }
> plot(roc01.NM, print.auc=T, auc.polygon=F, ci.type="shape",
+      ci.col=rgb(.3,.8,.3,.3))
> plot(roc02.NM, print.auc=T, auc.polygon=F, ci.type="shape",
```

```
+       ci.col=rgb(.8,.3,.3,.3), add=T, col=2,
+       print.auc.x=95, print.auc.y=95)
```



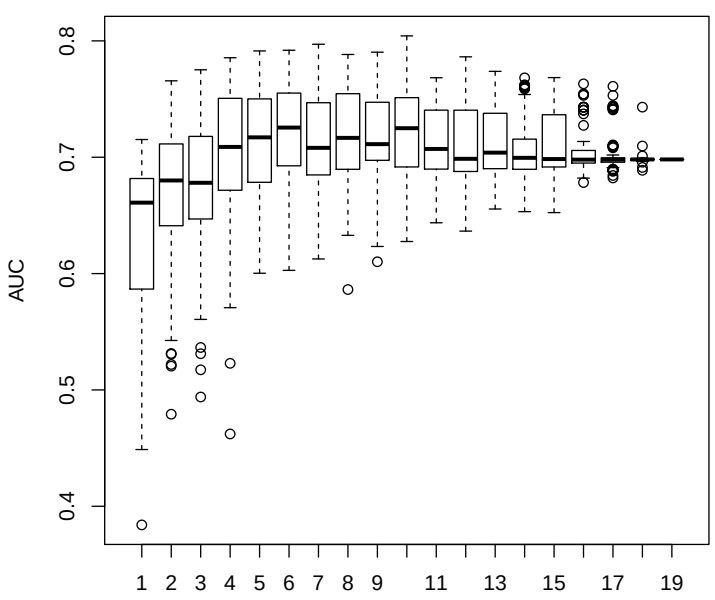
También se puede realizar una prueba formal, conocida como prueba de DeLong, para estimar si la diferencia entre las curvas es significativa:

```
> roc.test(roc01.NM,roc02.NM)

        DeLong's test for two correlated ROC curves

data:  roc01.NM and roc02.NM
Z = -0.40691, p-value = 0.6841
alternative hypothesis: true difference in AUC is not equal to 0
sample estimates:
AUC of roc1 AUC of roc2
   76.26577    76.53726
```

6.2. Selección de variables para *Bioclim*

Una pregunta frecuente es cómo mejora el desempeño de los modelos de *Bioclim* con el número de variables empleadas. Para ello podemos crear una rutina que seleccione todas las posibles combinaciones de variables, ajuste todos los modelos con los datos de distribución neotropical, evalúe con los datos de NeoMapas y guarde el valor de AUC, para luego comparar el cambio en AUC con el número de variables empleadas:

```
> if (!exists("rrs")){
+     rrs <- data.frame()
+     for (k in 1:19) {
+         combinaciones <- combn(1:19,k)
+         if (ncol(combinaciones)>100) {
+             combinaciones <- combinaciones[,sample(1:ncol(combinaciones),100)]
+         }
+         for (j in 1:ncol(combinaciones)) {
+             nms <- names(vars.NT)[combinaciones[,j]]
+             mdl00 <- bioclim(stack(vars.NT,layers=nms),
+                              pres)
+             e00 <- evaluate(mdl00, p=pres.NM, a=aus.NM, x=vars.vz)
+             rrs <- rbind(rrs,cbind(data.frame(k,AUC= e00@auc), bio=t(names(vars.NT) %in% nms)))
+         }
+     }
+ }
> boxplot(AUC~k,rrs,ylab="AUC",xlab="Número de variables")
```



Para esta especie el desempeño promedio es mejor entre cinco y nueve variables.

Utilizamos una regresión lineal múltiple del valor de AUC con variables que indique la inclusión o exclusión de las variables bioclimáticas en cada modelo para determinar cuales variables mejoran el desempeño predictivo del modelo.

```
> lm00 <- lm(AUC~bio.1+bio.2+bio.3+bio.4+bio.5+bio.6+bio.7+bio.8+bio.9+bio.10+bio.11+bio.12+bio.13+bio.14
> summary(lm00)
```

```
Call:
lm(formula = AUC ~ bio.1 + bio.2 + bio.3 + bio.4 + bio.5 + bio.6 +
    bio.7 + bio.8 + bio.9 + bio.10 + bio.11 + bio.12 + bio.13 +
    bio.14 + bio.15 + bio.16 + bio.17 + bio.18 + bio.19, data = rrs)

Residuals:
      Min        1Q    Median        3Q       Max
-0.308822 -0.011921  0.002449  0.017053  0.059383

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.6932040  0.0015910 435.716  < 2e-16 ***
bio.1TRUE    0.0111718  0.0015839   7.053 2.58e-12 ***
bio.2TRUE   -0.0033084  0.0015754  -2.100 0.035877 *
bio.3TRUE    0.0122758  0.0016123   7.614 4.49e-14 ***
bio.4TRUE    0.0213978  0.0015527  13.781  < 2e-16 ***
bio.5TRUE    0.0031329  0.0015510   2.020 0.043552 *
bio.6TRUE   -0.0250923  0.0015455 -16.235  < 2e-16 ***
bio.7TRUE    0.0169069  0.0015679  10.783  < 2e-16 ***
bio.8TRUE    0.0029755  0.0015549   1.914 0.055847 .
bio.9TRUE   -0.0576003  0.0015664 -36.772  < 2e-16 ***
bio.10TRUE   0.0015519  0.0015660   0.991 0.321852
bio.11TRUE   0.0083017  0.0015456   5.371 8.95e-08 ***
bio.12TRUE   0.0056116  0.0015404   3.643 0.000278 ***
bio.13TRUE  -0.0003732  0.0015519  -0.240 0.809990
bio.14TRUE   0.0008334  0.0015462   0.539 0.589957
bio.15TRUE   0.0115400  0.0015480   7.455 1.46e-13 ***
bio.16TRUE   0.0058447  0.0015590   3.749 0.000184 ***
bio.17TRUE   0.0024329  0.0015723   1.547 0.121991
bio.18TRUE   0.0065130  0.0015500   4.202 2.79e-05 ***
bio.19TRUE   0.0008224  0.0015622   0.526 0.598628
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02863 on 1619 degrees of freedom
Multiple R-squared:  0.5725,        Adjusted R-squared:  0.5675
F-statistic: 114.1 on 19 and 1619 DF,  p-value: < 2.2e-16
```

En este caso encontramos 10 variables con un coeficiente significativo y positivo.

Referencias

Booth, T., Nix, H., Busby, J. & Hutchinson, M., 2014. *Bioclim: the first species distribution modelling package, its early applications and relevance to most current maxent studies*. Diversity and Distributions, 20:1–9.

Camero, E. & Lobo, J. M., 2012. *The distribution of the species of* Eurysternus *Dalman, 1824 (Coleoptera: Scarabaeidae) in America: potential distributions and the locations of areas to be surveyed*. Tropical Conservation Science, 5:225–244.

Ferrer-Paris, J. R., Rodríguez, J. P., Good, T. C., Sánchez-Mercado, A. Y., Rodríguez-Clark, K. M., Rodríguez, G. A. & Solis, A., 2013. *Systematic, large-scale national biodiversity surveys: Neomaps as a model for tropical regions*. Diversity and Distributions, 19:215–231. URL http://onlinelibrary.wiley.com/doi/10.1111/ddi.12012/abstract.

Hijmans, R., Cameron, S., Parra, J., Jones, P. & Jarvis, A., 2005. *Very high resolution interpolated climate surfaces for global land areas*. International Journal of Climatology, 25:1965–1978.

Hijmans, R. J. & van Etten, J., 2012. *raster: Geographic analysis and modeling with raster data*. URL http://CRAN.R-project.org/package=raster. R package version 2.0-08.

Hijmans, R. J., Phillips, S., Leathwick, J. & Elith, J., 2012. *dismo: Species distribution modeling*. URL http://CRAN.R-project.org/package=dismo. R package version 0.7-17.

Pebesma, E. & Bivand, R., 2005. *Classes and methods for spatial data in r*. R News, 5. URL http://cran.r-project.org/doc/Rnews/.

R Development Core Team, 2011. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/. ISBN 3-900051-07-0.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C. & Müller, M., 2011. *pROC: an open-source package for R and S+ to analyze and compare ROC curves*. BMC Bioinformatics, 12:77.

Warnes, G. R., Bolker, B., Gorjanc, G., Grothendieck, G., Korosec, A., Lumley, T., MacQueen, D., Magnusson, A., Rogers, J. & others, 2014. *gdata: Various R programming tools for data manipulation*. URL http://CRAN.R-project.org/package=gdata. R package version 2.13.3.