

Disciplina IBE 875 - Modelagem de Distribuição de Espécies

Professores: Rodrigo Tardin, Maria Lucia Lorini, Mariana Mira Vasconcelos

Roteiro – Script 2 – Processamento e exploração visual de variáveis preditoras

Neste exercício, iremos importar as variáveis preditoras (também chamadas de camadas climáticas) e realizar um pré-processamento necessário para evitar potenciais vieses que podem levar a erros no processo de modelagem.

Funções usadas

- crop
- mask
- plot
- pairs
- vifcor
- vifstep

Agora que você lembrou/aprendeu o básico sobre o R Studio, vamos iniciar a etapa de modelagem em si, que é dividida em várias etapas. Para a disciplina seguiremos o protocolo ODMAP (Overview, Data, Model, Assessment, Prediction) elaborado por Zurell et al (2020)

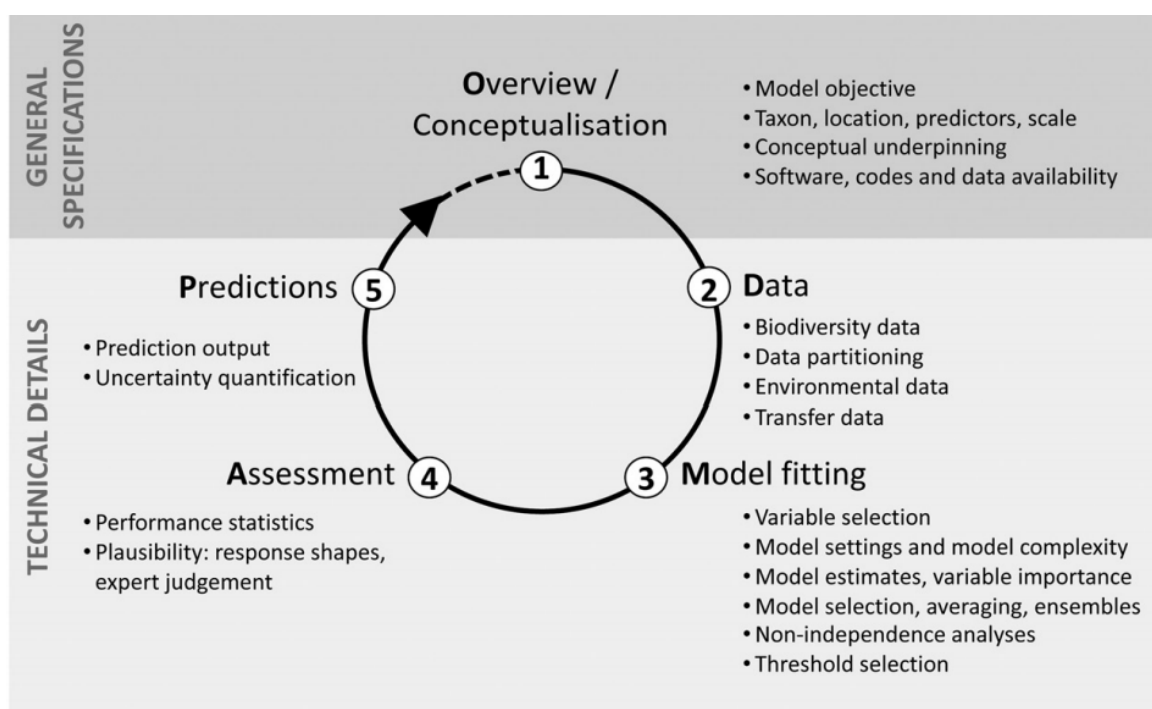


Figura 1. Retirado de Zurell et al (2020) representando os cinco passos no ciclo de modelagem de distribuição.

Nessa disciplina iremos modelar a distribuição da Araponga (*Procnias nudicollis*), uma ave passeriforme da família Cotingidae que se encontra ameaçada de extinção.

Antes de iniciarmos a obtenção e processamento das variáveis preditoras é importante que você carregue os pacotes que foram instalados na aula anterior, que envolve a função `'library'`. Não esqueça também de definir seu diretório de trabalho que deve ser a pasta inicial da disciplina de acordo com o que vc definiu no script 1.

```
setwd("C:/Users/rhtar/OneDrive/R/SDM_PPGE_PPGBio/") #Mude para o
endereço da pasta da disciplina no seu computador

getwd()
```

Para obter as variáveis preditoras (também chamadas de camadas aqui) em uma modelagem desse tipo, podemos fazer de duas formas: 1) de forma remota, indicando que o R se conecte com as bases existentes ou 2) fazendo o download das camadas online fora do R e depois carregando as camadas usando a função `raster`. Para evitar problemas que envolvem conexão à internet, vamos usar as camadas que já foram disponibilizadas na pasta do Google Drive da disciplina (pasta 'Camadas'). Como instruído na primeira aula, essas camadas já precisam estar baixadas e incluídas na pasta que você está trabalhando no R.

Para quem não conhece as bases de dados e as camadas disponíveis que podem ser acessadas pelo R, podemos usar funções específicas para isso incluídas no pacote `sdmpredictors`:

- Obtendo informações sobre os conjuntos de dados pre-existent

```
##### Listando e carregando os datasets pre-existent #####
```

```
list_datasets()
```

Aqui você irá ter acesso a todas as bases de dados existentes que podem ser usadas diretamente pelo R. Só que com essa função você não conseguirá ver todas as camadas que cada base de dados possui. Para isso, usaremos outra função

- Listando as camadas do WorldClim

```
list_layers("WorldClim")
```

Com essa função, você conseguirá ver todas as camadas que o WorldClim possui e seus detalhes como código da camada, nome, descrição, resolução espacial, unidade de medida, dentre outras.

Como já estamos disponibilizando as camadas, vamos carregá-las pela própria pasta da disciplina. Lembre-se que o endereço dentro do parênteses deve refletir o caminho até os arquivos seguindo a mesma configuração das pastas que você fez o download do Google Drive. Esse é o caminho relativo dos arquivos (e não o caminho completo) pois começa com a partir do presente diretório de trabalho (representado por um ponto). Depois do . vem o caminho relativo até os arquivos (/Camadas/Presente/WC_alt_lonlat.tif).

- Obtendo camadas do Worldclim a partir da pasta da disciplina

```
#Camadas topográficas
alt=raster("./Camadas/Presente/WC_alt_lonlat.tif")
#Altitude

#Camadas climáticas
bio1=raster("./Camadas/Presente/WC_bio1_lonlat.tif")
#Temperatura Anual Média

bio3=raster("./Camadas/Presente/WC_bio3_lonlat.tif")
#Isotermalidade

bio4=raster("./Camadas/Presente/WC_bio4_lonlat.tif")
#Sazonalidade da temperatura

bio7=raster("./Camadas/Presente/WC_bio7_lonlat.tif")
#Variação Anual da Temperatura

bio12=raster("./Camadas/Presente/WC_bio12_lonlat.tif")
#Precipitação Anual

bio15=raster("./Camadas/Presente/WC_bio15_lonlat.tif")
#Sazonalidade da precipitação
```

Agora você carregou para o R todas as variáveis preditoras que vamos usar. Como você deve ter percebido, você não consegue visualizar essas camadas ainda. Antes de gerarmos os primeiros mapas, vamos obter alguns detalhes sobre essas camadas. As funções abaixo vão permitir que você visualize os limites de longitude/latitude mínimas e máximas (`bbox`), o número de colunas (`ncol`), número de linhas (`nrow`) e resolução em graus decimais (`res`). Vamos juntar todos esses comandos numa mesma linha separados por ;

- Obtendo os limites e detalhes das camadas climáticas

```
bbox(alt); ncol(alt); nrow(alt); res(alt)
bbox(bio1); ncol(bio1); nrow(bio1) ; res(bio1)
bbox(bio3); ncol(bio3); nrow(bio3) ; res(bio3)
bbox(bio4); ncol(bio4); nrow(bio4) ; res(bio4)
bbox(bio7); ncol(bio7); nrow(bio7) ; res(bio7)
bbox(bio12); ncol(bio12); nrow(bio12) ; res(bio12)
bbox(bio15); ncol(bio15); nrow(bio15) ; res(bio15)
compareRaster(alt, bio1, bio3, bio4, bio7, bio12, bio15)
```

Com esses códigos você consegue perceber que a camada envolve a superfície terrestre do planeta como um todo, todas tem o mesmo número de linhas e colunas, assim como a mesma resolução espacial. Essas informações são importantes para quando precisarmos unir todas as camadas em um único objeto.

Agora que já sabemos mais sobre as camadas, vamos projetá-las no espaço geográfico sob o formato de mapas para melhor visualização usando a função `'plot()'`.

- Visualizando as camadas climáticas

```
plot(alt, main="Altitude")
par(mfrow=c(3,2),mar=c(3,3,2,0))
plot(bio1, col=topo.colors(255), main="Temperatura Média Anual
(bio1)")
plot(bio3, col=topo.colors(255), main="Isotermalidade (bio3)")
plot(bio4, col=topo.colors(255), main="Sazonalidade da temperatura
(bio4)")
plot(bio7, col=topo.colors(255), main="Variação Anual de Temperatura
(bio7)")
plot(bio12, col=topo.colors(255), main="Precipitação Anual (bio12)")
```

```
plot(bio15, col=topo.colors(255), main="Sazonalidade da precipitação  
(bio15)")
```

Aqui nós inserimos os argumentos `xlim` e `ylim` para limitarmos a visualização da camada apenas para a América do Sul.

Como a Araponga não ocorre em toda a América do Sul, vamos explorar o uso de duas funções importantes `crop` e `mask`. A função `crop` recorta a camada dentro de uma extensão especificada, em geral, sob a forma de um retângulo. Ao usar a função `mask` em conjunto com a função `crop` conseguimos criar uma máscara para recortar a camada de forma mais ajustada ao polígono que se deseja, criando um novo objeto. Com essas funções já estamos de fato fazendo um processamento das camadas e não apenas mudando os limites para a visualização. Primeiro, vamos recortar cada camada para o Brasil.

- Carregando o mapa do Brasil

```
Brasil=getData('GADM', country= 'Brazil', level=1, path = "./Camadas")
```

- Cortando cada camada de uma vez para o Brasil e 'perfumando' minimamente os mapas

```
altBrasil=mask(crop(alt,Brasil),Brasil)
```

```
plot(altBrasil, main = "Altitude Brazil",xlab = "Longitude", ylab =  
"Latitude")
```

```
bio1Brasil=mask(crop(bio1,Brasil),Brasil)
```

```
plot(bio1Brasil, main = "bio1 Brazil",xlab = "Longitude", ylab =  
"Latitude")
```

```
bio3Brasil=mask(crop(bio3,Brasil),Brasil)
```

```
plot(bio3Brasil, main = "bio3 Brazil",xlab = "Longitude", ylab =  
"Latitude")
```

```
bio4Brasil=mask(crop(bio4,Brasil),Brasil)
```

```
plot(bio4Brasil, main = "bio4 Brazil",xlab = "Longitude", ylab =  
"Latitude")
```

```
bio7Brasil=mask(crop(bio7,Brasil),Brasil)
```

```
plot(bio7Brasil, main = "bio7 Brazil",xlab = "Longitude", ylab =  
"Latitude")
```

```
bio12Brasil=mask(crop(bio12,Brasil),Brasil)
```

```
plot(bio12Brasil, main = "bio12 Brazil",xlab = "Longitude", ylab =  
"Latitude")
```

```
bio15Brasil=mask(crop(bio15,Brasil),Brasil)
```

```
plot(bio15Brasil, main = "bio15 Brazil",xlab = "Longitude", ylab =  
"Latitude")
```

Explore os mapas com calma, avaliando como as variáveis preditoras se distribuem ao longo do território brasileiro, onde conseguimos visualizar os maiores e menores valores para cada uma das variáveis.

Agora vamos comparar as novas camadas recortadas e ajustadas para o Brasil, com as camadas originais que tem extensão mundial. Tomemos a camada 'bio1' como exemplo:

- Comparando as dimensões das camadas Brasil e Mundo

```
bbox(bio1); ncol(bio1); nrow(bio1) ; res(bio1)
```

```
bbox(bio1Brasil); ncol(bio1Brasil); nrow(bio1Brasil) ; res(bio1Brasil)
```

Você conseguiu perceber a diferença entre as camadas, em relação aos limites, número de linhas e colunas e resolução?

Agora vamos repetir o mesmo procedimento para o polígono de distribuição da Araponga. Esse polígono foi obtido no site da União Internacional para a Conservação da Natureza (*IUCN* em inglês - <https://www.iucnredlist.org/>) e representa a área de extensão de ocorrência da espécie, definida como “a área contida dentro do menor limite imaginário contínuo que possa ser traçado para englobar todos os pontos conhecidos, inferidos ou projetados da presença atual de uma espécie, excluindo os casos de errantes e visitantes”.

Assim como para o Brasil, primeiro temos que carregar o polígono de ocorrência da Araponga para o *workspace* que estamos trabalhando. Por se tratar de dados vetoriais (*shapefile*) e não matriciais (*raster*) vamos usar uma função diferente. Lembre-se que o arquivo com polígono da distribuição da Araponga também foi disponibilizado na pasta do Drive.

- Vamos unir as variáveis em um único 'stack' (pilha de camadas, uma em cima da outra)

```
biostack = stack(altBrasil, bio1Brasil, bio3Brasil, bio4Brasil,
bio7Brasil, bio12Brasil, bio15Brasil)

plot(biostack)
```

- Carregando o polígono da Araponga

```
procniaspolygon=readOGR("../Procnias/procnias_polygon.shp")
```

Não se esqueça de indicar o formato (.shp) sempre depois do nome do arquivo. Muitas vezes os erros são gerados pelo esquecimento desse pequeno, mas importante detalhe.

Vamos visualizar esse polígono da Araponga dentro do mapa do Brasil.

- Plotando o polígono da distribuição da Araponga no mapa do Brasil

```
plot(altBrasil, main="Altitude Brasil")
plot(procniaspolygon, add=T)
```

Agora é só usar as funções `crop` e `mask` mais uma vez, conforme fizemos acima, para recortar as camadas para o polígono da Araponga. Atente-se que o primeiro argumento dessas funções usadas em conjunto sempre será a camada que você quer cortar seguido pela área que você está usando para cortar a camada, como explicado abaixo. Ao mesmo tempo vamos plotar os mapas de cada camada cortados. Explore bem os mapas.

- Cortando cada camada de uma vez para o polígono da Araponga e 'perfumando' minimamente os mapas

```
altprocnicas=mask(crop(alt,procnicaspolygon),procnicaspolygon)
plot(altprocnicas, main = "Altitude Procnias",xlab = "Longitude",
ylab = "Latitude")
bio1procnicas=mask(crop(bio1,procnicaspolygon),procnicaspolygon)
plot(bio1procnicas, main = "bio1 Procnias",xlab = "Longitude",
ylab = "Latitude")
bio3procnicas=mask(crop(bio3,procnicaspolygon),procnicaspolygon)
plot(bio3procnicas, main = "bio3 Procnias",xlab = "Longitude",
ylab = "Latitude")
bio4procnicas=mask(crop(bio4,procnicaspolygon),procnicaspolygon)
plot(bio4procnicas, main = "bio4 Procnias",xlab = "Longitude",
ylab = "Latitude")
bio7procnicas=mask(crop(bio7,procnicaspolygon),procnicaspolygon)
plot(bio7procnicas, main = "bio7 Procnias",xlab = "Longitude",
ylab = "Latitude")
bio12procnicas=mask(crop(bio12,procnicaspolygon),procnicaspolygon)
plot(bio12procnicas, main = "bio12 Procnias",xlab = "Longitude",
ylab = "Latitude")
bio15procnicas=mask(crop(bio15,procnicaspolygon),procnicaspolygon)
plot(bio15procnicas, main = "bio15 Procnias",xlab = "Longitude",
ylab = "Latitude")
```

Agora precisamos unir todas as variáveis unidas em único objeto (*stack*), já que esse formato é necessário para a modelagem de distribuição que iremos realizar em outras aulas.

- Unindo as variáveis em um unico 'stack'

```
biostack1=mask(crop(biostack,procniaspolygon),procniaspolygon)
biostack1
summary(biostack1)
```

Vamos explorar visualmente os mapas das variáveis unidas em único ‘stack’

```
plot(biostack1)
```

Agora temos um único objeto com todas as camadas armazenadas, cortadas na extensão do polígono de ocorrência da Araponga. Vamos explorar um pouco mais os detalhes do objeto, apenas lendo o nome dele. Aqui você consegue obter várias informações como dimensões, resolução, extensão, o sistema de coordenadas (crs), e as camadas que estão armazenadas dentro desse stack.

```
biostack1
```

Antes de darmos por definitivo que essas variáveis serão usadas na modelagem precisamos analisar se elas não têm problemas de correlação ou multicolinearidade. Tais problemas podem afetar seriamente a relação da espécie com as variáveis e precisa ser ajustada, caso exista. Aqui usaremos dois testes distintos: um teste exclusivo para correlação (`pairs`) e um teste de Fator de Variação da Inflação (*VIF*, na sigla em inglês) (`vifcor` e `vifstep`). Nesses testes estamos indicando os valores de cortes, dentre os quais, acima deles, essas variáveis são consideradas muito correlacionadas ou colineares e precisam ser descartadas.

```
#Checando existência de correlação e multicolinearidade entre
variáveis
```

```
#Correlação entre as camadas usando a função 'pairs'
pairs(biostack1)
```

Você verá os resultados desse teste na tela dos plots (abaixo à direita), dentro do RStudio, com os gráficos e a matriz de correlação entre cada par de variáveis. Vamos usar também o pacote `corrplot` para visualizar as correlações entre as variáveis, quanto mais escuro a cor maior o valor de correlação positiva (em azul) e negativa (em vermelho). A `bio4` é positivamente correlacionada com a `bio7` e negativamente com a `bio3` e as correlações são bem forte (cores mais intensas).

```
set.seed(1963) #seed number para sempre gerar os mesmos pontos
abaixo
```



```

backgr <- randomPoints(biostack1, 10000)
absclim <- data.frame(extract(biostack1, backgr))
absclim.std <- data.frame(scale(absclim)) # Scale variables
library(corrplot)
M <- cor(absclim.std)
corrplot.mixed(M, upper = "ellipse", lower = "number", number.cex
= 0.8, tl.cex = 0.8)

# Checagem de multicolinearidade entre as camadas - remoção
stepwise com threshold = 0.7

vifcor(biostack, th=0.7)
#th = valor de corte para correlação

vifstep(biostack, th = 10)
#th = valor de corte para VIF

```

Você verá os resultados desse teste na tela do Console, dentro do RStudio, onde os resultados de correlação máxima e mínima, assim como os valores de VIF estarão disponíveis. Na primeira função (`vifcor`), o valor de corte será focado na correlação e não na multicolinearidade, logo você verá variáveis que foram retidas com o $VIF > 10$, porém serão excluídas aquelas com o coeficiente de correlação > 0.7 . Na segunda função (`vifstep`).

Com esses resultados podemos ver que as variáveis Bio 1 e Bio 4 possuem valores de correlação e colinearidades altos, portanto, serão descartadas do nosso conjunto de variáveis preditoras para modelar a distribuição da Araponga.

- Unindo as variáveis em um unico 'stack' após as análises de correlação e multicolinearidade

```

biostack2 = stack(altprocnias, bio3procnias,
bio7procnias, bio12procnias, bio15procnias)

biostack2

```

Agora apenas as variáveis que não estão correlacionadas nem apresentam tendências de multicolinearidade estão unidas dentro de um novo objeto do tipo RasterStack.

- Plotando o novo 'stack' com as camadas climáticas não correlacionadas e não-colinares dentro do polígono da Araponga

```
plot(biostack2, main=c("Altitude", "Isotermalidade (bio3)",  
"Variação Anual de Temperatura (bio7)", "Precipitação Anual  
(bio12)", "Sazonalidade da precipitação (bio15)"))
```

Agora você já tem o conjunto de suas variáveis preditoras recortadas para a área de calibração do modelo (polígono da Araponga), constando apenas as variáveis não-correlacionadas. Esse é o primeiro passo para a modelagem nas etapas seguintes.

Vamos então salvar todos esses objetos como um RData que podemos carregar durante os próximos passos. Dessa forma, não precisamos gerar todos esses objetos novamente. Especialmente o biostack2 que iremos usar nos próximos passos.

```
save.image(file="script2.RData")
```

#Fim do Script 2