Fernanda Alves

Sou aluna de mestrado do programa de Zoologia da USP. O título da minha dissertação é “Densidade populacional, organização social e seleção de habitat pelo mutum-do-sudeste *Crax blumenbachii* . Pretendo fazer minhas análises de seleção de habitat no R e por isso escolhi a disciplina.

exec

Minha Proposta

Plano A

As listas de espécies ameaçadas são importantes ferramentas para estudos voltados à conservação. No entanto, uma mesma espécie pode estar classificada em diferentes categorias de ameaça dependendo da lista e é importante que essas diferentes categorias sejam conhecidas. Quero gerar uma função que facilite o uso dessas listas trazendo de maneira compilada informações de diversas listas. A função receberá como argumentos uma lista que contenha espécies e suas respectivas Ordens e Famílias, *n* listas com categorias de ameaça (Ex: IUCN, IBAMA, listas estaduais) e argumentos para filtragem por família, espécie e/ou categoria de ameaça. Como retorno será dado uma lista compilada dos dados obtidos das *n* listas, permitindo a comparação das categorias de ameaça em diferentes listas no formato “Ordem, Família, Espécie, Categoria de ameaça n_1, Categoria de ameaça n_2, ...”, com a opção de exportar a lista gerada para um arquivo .csv. Além disso, será gerado um gráfico de barras por lista fornecida com a proporção de espécies ameaçadas por Ordem e Família. Para construir a função vou utilizar como base a lista de aves ameaçadas da IUCN e a lista Nacional de espécies ameaçadas.

Plano B

Construir uma função que receba as dimensões de cômodos de uma casa, valores de materiais (argamassa, revestimento) e de mão de obra para calcular a quantidade de materias e custos para cada ambiente e custo total em uma construção.

Comentários

Esta função me pareceu muito interessante porém, da forma que está apresentada ainda está muito específica. Veja que em sua função entra-se com uma lista com espécies, famílias e ordens e obtem-se outra lista com os dados sintetizados conforme o desejo do usuário. Diante disso, tenho algumas sugestões: generalize sua função para outros grupos taxonômicos e inclua um módulo para adição de uma nova lista de ameaça no banco de dados interno de sua função.

Seu plano B é extremamente trivial, envolvendo apenas cálculos algébricos muitos simples. Por exemplo, o custo total é simplesmente a soma dos valores de materiais multiplicado pela soma da área dos comodos da caso. Dada a adequabilidade do seu plano A, sugiro que você se foque nele.

Ale

Fernanda: como o Marcelo sugeriu, foco no plano A. O principal é ter a base de dados incluída na função, ou seja as listas de espécies ameaçadas de diferentes classificações. — *Alexandre Adalardo de Oliveira 2013/03/24 21:38*

Fernanda

Ale e Marcelo: Acabei de postar minha função (18/04/2013) e lendo os comentários novamente acho que houve um engano. Eu não tenho uma base de dados dentro da minha função. Quando disse que usaria como base a lista de aves ameaçadas da IUCN, eu quis dizer que usaria ela como exemplo e não como base de dados interna, pois a idéia da função é justamente ser genérica como o Marcelo sugeriu.

Função

```
compila.ameacadas<-function(x, Listas, nome.listas=NULL, filtro.ordem=NULL,
filtro.familia=NULL, filtro.especie=NULL, filtro.ca=NULL, nome.csv=NULL,
grafico=TRUE)
{
  # Lista principal com Ordem, Família e Espécies.
  Lista.retorno<-na.omit(x)
  # define o nome das colunas da Lista.retorno
  names(Lista.retorno)<-c("Ordem","Familia","Especie")

  # Verifica se nome.listas não foi fornecido.
  if(is.null(nome.listas))
  {
    nome.listas<-rep("",length(Listas))
    cont<-1
    for(Lista in Listas)
    {
      nome.listas[cont]<-paste("Lista",cont)
      cont=cont+1
    }
  }
  # Aplicando filtros
  # Filtro Ordem
  if(is.null(filtro.ordem)==FALSE)
  {
    Lista.retorno<-Lista.retorno[Lista.retorno$Ordem %in% filtro.ordem,]
  }
  # Filtro Família
  if(is.null(filtro.familia)==FALSE)
  {
    Lista.retorno<-Lista.retorno[Lista.retorno$Familia==filtro.familia,]
```

```
  }

  # Filtro Espécie
  if(is.null(filtro.especie)==FALSE)
  {
    Lista.retorno<-Lista.retorno[Lista.retorno$Especie %in% filtro.especie,]
  }

  # Inicia a criação de colunas na Lista.retorno, baseado nas listas que o
usuário forneceu para comparar
  Indice.coluna<-4 # Começa a partir da 4a coluna, pois a Lista.retorno deve
possuir Ordem, Família e Espécie, então a partir da 4a entram as colunas de
categoria de ameaça de cada lista fornecida

  # Faz um loop no objeto nome.listas que contém o nome de cada lista
fornecida e cria uma coluna com estes nomes no objeto Lista.retorno
  for(nome in nome.listas)
  {
    Lista.retorno$aux<-""
    names(Lista.retorno)[Indice.coluna]<-nome
    Indice.coluna<-Indice.coluna+1
  }

  # Inicia a procura de categoria de ameaça em cada uma das listas
fornecidas
  Indice.coluna<-4 # Volta o índice para a 4a coluna

  # Faz um loop nas listas fornecidas
  for(Lista in Listas)
  {
    # Nomeia as colunas de cada lista fornecida
    names(Lista)<-c("Especie", "CA")
    # Faz loop em todas as espécies da Lista.retorno
    for(especie in Lista.retorno$Especie)
    {
      categoria.encontrada<-NULL
      # Procura na lista de ameaça a espécie de acordo com a Lista.retorno
      # Verifica se tem filtro por categoria.
      if(is.null(filtro.ca)==FALSE)
      {
        ## COM FILTRO POR CATEGORIA
        # Verifica se existe a espécie na lista de ameaça
        if(lenght(Lista$CA[Lista$Especie==especie & Lista$CA==filtro.ca])>0)
          try(categoria.encontrada<-
as.character(Lista$CA[Lista$Especie==especie & Lista$CA==filtro.ca]),
silent=T)
        else
          categoria.encontrada = NA
      }
      else
      {
```

```
        ## SEM FILTRO POR CATEGORIA
        # Verifica se existe a espécie na lista de ameaça
        if(length(Lista$CA[Lista$Especie==especie])>0)
          try(categoria.encontrada<-
as.character(Lista$CA[Lista$Especie==especie]), silent=T)
        else
          categoria.encontrada = NA
      }
      # Preenche a categoria encontrada na Lista.retorno
      try(Lista.retorno[Lista.retorno$Especie==especie, Indice.coluna]<-
categoria.encontrada, silent=T)
    }
    Indice.coluna<-Indice.coluna+1
  }
  # Exporta a lista.retorno para um arquivo csv
  if(is.null(nome.csv)==F)
    write.table(Lista.retorno, nome.csv, sep=";", row.names=F)

  # Gráficos
  if(grafico==TRUE)
  {
    for(nome in nome.listas)
    {
      #gerando gráfico por Ordem.
      tabela<-table(Lista.retorno[!is.na(Lista.retorno[nome]),1])
      freq.relativa<-tabela/sum(tabela)
      if(sum(tabela)>0)
      {
        x11()
        mtext(text = names(freq.relativa), side = 1, at =
barplot(freq.relativa, col=rainbow(length(freq.relativa)),
main=paste(nome,": Ordens"), axisnames = FALSE, ylab="Frequência"), line =
0, las=2,cex=0.6)
      }
      #gerando gráfico por Família.
      tabela<-table(Lista.retorno[!is.na(Lista.retorno[nome]),2])
      freq.relativa<-tabela/sum(tabela)
      if(sum(tabela)>0)
      {
        x11()
        mtext(text = names(freq.relativa), side = 1, at =
barplot(freq.relativa, col=rainbow(length(freq.relativa)),
main=paste(nome,": Famílias"), axisnames = FALSE, ylab="Frequência"), line =
0, las=2,cex=0.6)
      }
      #gerando gráfico por Categoria de Ameaça.
      tabela<-table(Lista.retorno[!is.na(Lista.retorno[nome]),nome])
      freq.relativa<-tabela/sum(tabela)
      if(sum(tabela)>0)
      {
```

```
        x11()
        barplot(freq.relativa, col=rainbow(length(freq.relativa)),
names.arg=names(freq.relativa), main=paste(nome,": Categoria de Ameaça"),
ylab="Frequência")
      }
    }
  }
  return(Lista.retorno)
}
```

Help

```
compila.ameacadas         package:nenhum       R Documentation


Compilação de listas de espécies ameaçadas

Description:

A partir de n listas de espécies ameaçadas (Lista Internacional, Lista
Nacional, Listas Estaduais)realiza a compilação para que seja obtida uma
lista única com as categorias de ameaças de cada uma das listas fornecidas e
então gera gráficos com as proporções de espécies ameaçadas por Ordem ou
Família.

Usage:

compila.ameacadas (x, Listas, nome.listas=NULL, filtro.ordem=NULL,
filtro.familia=NULL, filtro.especie=NULL, filtro.ca=NULL, nome.csv=NULL,
grafico=TRUE)

Arguments:

x                     lista principal que será usada como base, que deve
conter apenas 3 colunas: Ordem, Família e Espécie.

Listas                n listas com categorias de ameaça, que devem conter
apenas 2 colunas: Espécie e Categoria de Ameaça.

nome.listas           nome para cada lista fornecida, sendo na mesma ordem

filtro.ordem          um vetor opcional com as Ordens para filtragem

filtro.familia        um vetor opcional com as Famílias para filtragem

filtro.especie        um vetor opcional com as Espécies para filtragem

filtro.ca             um vetor opcional com as categorias de ameaça para
filtragem
```

```
nome.csv              nome para exportar a lista compilada no formato csv
(opcional)

grafico               indica se deve gerar gráficos ou não por padrão
grafico=TRUE


Details:

Usando as espécies da lista principal a função irá pesquisar em cada lista
fornecida a presença dessas espécies e suas respectivas categorias de
ameaça.

Se nome.listas não for fornecido, as colunas de categoria de ameaça da lista
retorno serão geradas com nomes genéricos.

O filtros podem ser usados de maneira combinada.


Value:

Retorna uma lista compilada com as categorias de ameaças das diferentes
listas utilizadas e gráficos com as proporções das diferentes Ordens e
Famílias. Essa lista pode ser importanda para um arquivo .cvs


Author(s):

Fernanda Alves

fernandabiologia@gmail.com

References:

Crawley, M. J. 2007. The R Book. John Wiley & Sons, LTDA.


See Also:

read.table, list


Examples:

## Uso sem filtros

teste<-compila.ameacadas(read.table("CBRO, 2009.csv", header=T, sep=";"),
Lista=list(read.table("IUCN, 2012.csv", header=T, sep=";",
na.strings=""),read.table("IBAMA, 2003.csv", header=T, sep=";",
na.strings="")))
```

```
## Uso com filtro por Ordem e nome das listas

teste<-compila.ameacadas(read.table("CBRO, 2009.csv", header=T, sep=";"),
Lista=list(read.table("IUCN, 2012.csv", header=T, sep=";",
na.strings=""),read.table("IBAMA, 2003.csv", header=T, sep=";",
na.strings="")), nome.listas=c("IUCN Red List","IBAMA"),
filtro.ordem=c("Galliformes","Passeriformes"))
```

Função

Help

CSVs para teste Lista principal Lista aves ameaçadas IBAMA Lista aves ameaçadas IUCN

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:fernandabiologia:start

Last update: **2020/08/12 06:04**