

Silvania Goularte Correia



Mestranda em Ecologia, Instituto de Biociências, USP.

O título da minha dissertação é: “Estoque de carbono na biomassa subterrânea em Cerrado nativo e sob usos silvipastoris”, com orientação da Profª Dra. Vania Regina Pivello.

Meus exercícios

Linque para a página com meus exercícios resolvidos: [exec](#)

Proposta de trabalho final

Principal

FUNÇÃO: IVI

Proposto por Curtis e McIntoshi (1951), o Índice de Valor de Importância (IVI) expressa, numericamente, a importância de determinada espécie dentro de uma comunidade arbórea. Para o seu cálculo, utiliza-se a fórmula:

$$IVI = DR + FR + DoR$$

Onde,

DR = Densidade Relativa, obtido pela divisão do número de indivíduos da espécie em questão pelo número total de indivíduos de todas as espécies arbóreas da comunidade.

FR = Frequência Relativa. A frequência de uma espécie é dada pelo número de parcelas em que a espécie ocorre, dividido pelo número total de parcelas. A frequência relativa é obtida pela divisão da frequência da espécie em questão pela soma das frequências de todas as espécies da comunidade.

DoR = Dominância Relativa, obtida pela divisão entre a área basal da espécie em questão pela soma da área basal de todas as espécies da comunidade.

DR, FR e DoR são expressos em porcentagem, portanto, a soma do IVI de todas as espécies do inventário de uma comunidade é igual a 300.

INPUT: Data frame contendo os dados obtidos em campo (Número da parcela, nome das espécies arbóreas presentes, nº de árvores de cada espécie dentro da parcela e DAP de cada árvore).

OUTPUT: Data frame contendo o nome de cada espécie do inventário, sua DR, FR e DoR (em %), IVI e IVI relativo (em %).

Referências

Curtis, J. T.; McIntoshi, R. P. An Upland Forest Continuum in the Prairie-Forest Border Region of Wisconsin. *Ecology*, 32(1): 476-496, 1951.

Poggiani, F.; Oliveira, R. E.; Cunha, G. C. Práticas de Ecologia Florestal. Disponível em: <<http://www.ipef.br/publicacoes/docflorestais/cap16.pdf>>. Acesso em: 19 mar. 2015.

Plano B

FUNÇÃO: IMC

O índice de massa corporal (IMC) é adotado pela Organização Mundial de Saúde (OMS) para o diagnóstico do sobrepeso e da obesidade. O IMC é calculado pela fórmula:

$$\text{IMC} = M/H^2$$

Onde,

M = Massa do indivíduo, em quilos. H = Altura do indivíduo, em metros.

A interpretação do IMC varia de acordo com o sexo e a idade do indivíduo (tabelas para cada idade e sexo podem ser vistas [aqui](#) e [aqui](#)).

INPUT: Data frame contendo identificação dos indivíduos (ID ou nome), altura, peso (em uma ou mais ocasiões), sexo e idade.

OUTPUT: Data frame contendo a identificação dos indivíduos, peso atual, classificação de acordo com a interpretação do IMC (abaixo, ideal, sobrepeso, obeso) e qual seria o peso ideal de cada indivíduo. O usuário também poderá optar por gerar um gráfico com a evolução do peso dos indivíduos ao longo do tempo.

Oi, Silvania!

As suas propostas ainda estão um pouco simples para o trabalho. Seria interessante, talvez, você pesquisar outros trabalhos que critiquem ou refinem esse índice de importância que você menciona e, então, comparar os resultados do índice original e do índice melhorado, por exemplo.

Para a proposta do IMC, pode valer a mesma coisa: seria interessante você procurar trabalhos que discutam as limitações desse índice e que procurem aprimorá-lo. A partir disso, você poderia comparar os modelos, por exemplo.

Essas minhas sugestões são para você ir um pouco além do nível dos exercícios que foram propostos no curso: não

esperamos que você utilize conceitos além dos que foram apresentados, mas que você encontre meios de explorar esses conceitos de forma um pouco mais elaborada do que nos exercícios pedidos até agora.

— *Viviane Santos* 2015/03/24 15:05

Comentários Vitor Rios

Concordo com Viviane. Não se atenha às análises que vc vai fazer na sua dissertação. para primeira função, vc poderia, por exemplo, comparar IVIs da mesma espécie em comunidades diferentes, ou tentar comparar as comunidades através das diferenças nos IVIs de suas espécies. Para a segunda função, o calculo do IMC é simplista demais, e note que você não tem como calcular uma evolução no tempo se ela não estiver nos dados de entrada

Comentários Silvania

Obrigada pelos comentários! Infelizmente, ainda não consegui refazer as propostas devido a um problema de saúde (suspeita de dengue). Vou agregar as contribuições dadas às propostas e tentar fazer a função A.

Adequação das propostas

Principal

FUNÇÃO: “Importância Ecológica” (eco.imp)

Esta função permitirá ao usuário a comparação entre comunidades florestais quanto a importância ecológica das espécies que as compõem. Esta comparação poderá ser feita através do Índice de Valor de Importância - IVI, do Índice de Valor de Cobertura - IVC ou de ambos. Os índices expressam, numericamente, a importância de determinada espécie dentro de uma comunidade arbórea. No entanto, o IVC sofre menos influência do processo de amostragem, sem superestimar a densidade de indivíduos em uma área.

Além disso, a função gerará diagramas de frequência para cada comunidade florestal estudada. Estes diagramas são histogramas que exibem o número de espécies em cada classe de frequência absoluta (porcentagem de parcelas que apresentam determinada espécie), de acordo com a Figura 1. Com isso, o usuário poderá visualizar rapidamente se as comunidades estudadas apresentam homogeneidade ou heterogeneidade florística, e juntamente com os valores para IVI e IVC, poderá verificar quais espécies possuem maior expressão em cada comunidade. Comunidades com forte representatividade nas classes mais altas de frequência (IV ou V) apresentam alto grau de homogeneidade florística. Por outro lado, comunidades com maior representatividade nas classes I e II são consideradas muito heterogêneas.



Figura 1. Classes de frequência absoluta

Para o cálculo do IVI, utiliza-se a fórmula:

$$IVI = DR + FR + DoR$$

Onde,

DR = Densidade Relativa, obtido pela divisão do número de indivíduos da espécie em questão pelo número total de indivíduos de todas as espécies arbóreas da comunidade.

FR = Frequência Relativa. A frequência de uma espécie é dada pelo número de parcelas em que a espécie ocorre, dividido pelo número total de parcelas. A frequência relativa é obtida pela divisão da frequência da espécie em questão pela soma das frequências de todas as espécies da comunidade.

DoR = Dominância Relativa, obtida pela divisão entre a área basal da espécie em questão pela soma da área basal de todas as espécies da comunidade.

DR, FR e DoR são expressos em porcentagem, portanto, a soma do IVI de todas as espécies do inventário de uma comunidade é igual a 300.

Para o cálculo do IVC, a fórmula utilizada é:

$$IVC = DR + DoR$$

ou

$$IVC = IVI - FR$$

A soma do IVC de todas as espécies do inventário de uma comunidade é igual a 200.

INPUT: Data frame contendo os dados obtidos em campo (Identificação do local/comunidade do inventário, número da parcela, nome das espécies arbóreas presentes, nº de árvores de cada espécie dentro da parcela e DAP de cada árvore).

OUTPUT: Data frame contendo o nome de cada espécie do inventário, sua DR, FR e DoR (em %), IVI e IVI relativo (em %), IVC e IVC relativo (em %); diagrama de frequência para cada comunidade, contendo o número de espécies presente em cada classe de frequência absoluta.

Referências

Instituto Ambiental do Paraná. Parâmetros Fitossociológicos. Disponível em:
<http://www.iap.pr.gov.br/arquivos/File/Plano_de_Manejo/PE_Rio_Guarani/Anexos/anexo_2.pdf>.
Acesso em: 02 abr. 2015.

Poggiani, F.; Oliveira, R. E.; Cunha, G. C. Práticas de Ecologia Florestal. Disponível em:
<<http://www.ipef.br/publicacoes/docflorestais/cap16.pdf>>. Acesso em: 19 mar. 2015.

Silva, A. G.; Guesdes-Bruni, R. R. Uma abordagem quantitativa para importância de recursos florais em comunidades vegetais. *Natureza online*, 3(1): 3-6, 2005.

Trabalho final

Código da função

```
##### Funcao importancia ecologica #####

eco.imp = function(x, index = "both", diagram=TRUE, dim.diagram = NULL,
save.diagram = FALSE) #criacao da funcao e seus argumentos
{
  ##Validacao dos argumentos
  if (diagram != TRUE & diagram != FALSE) #verifica se o usuario inseriu
um valor logico para o argumento "diagram"
  {
    stop("Invalid diagram") #exibe mensagem de erro caso o usuario nao
tenha inserido um valor logico em "diagram"
  }
  if(diagram) #caso o usuario queira que diagramas sejam gerados, serao
realizadas as validacoes abaixo
  {
    if(is.null(dim.diagram)) #verifica se o usuario inseriu um valor em
"dim.diagram"
    {
      stop("dim.diagram is mandatory when diagram is TRUE") #exibe
mensagem de erro caso o usuario nao tenha inserido um valor em "dim.diagram"
    }
    if(length(dim.diagram) != 2 | class(dim.diagram) != "numeric")
#verifica se o usuario inseriu um vetor de duas posicoes da classe "numeric"
em "dim.diagram"
    {
      stop("Invalid dim.diagram") #exibe mensagem de erro caso o
usuario nao tenha inserido um vetor de duas posicoes da classe "numeric" em
"dim.diagram"
    }
    if(save.diagram != TRUE & save.diagram != FALSE) #verifica se o
usuario inseriu um valor logico para o argumento "save.diagram"
    {
      stop("Invalid save.diagram") #exibe mensagem de erro caso o
usuario nao tenha inserido um valor logico em "save.diagram"
    }
  }
  if (index != "both" & index != "ivi" & index != "ivc") #verifica se o
usuario inseriu um nome valido para "indice"
  {
    stop("Invalid Index") #exibe mensagem de erro caso o usuario nao
tenha inserido um nome valido em "indice"
  }

  ##Declaracao de variaveis
  n.species = table(x[,3], x[,1]) #cria tabela com o numero de individuos
```

```
de cada especie por local estudado
total = apply(n.species, 2, sum) #cria objeto obtendo o numero total de
individuos presentes em cada local estudado
dr = n.species #cria objeto que contera a densidade relativa de cada
especie por local estudado
for(i in 1:dim(x)[2]) #abre ciclo para calcular densidade relativa de
cada especie por local estudado
{
  dr[,i]=(n.species[,i]/total[i])*100 #calculo de densidade relativa
de cada especie por local estudado
}
n.parcel = x #atribui o data.frame contido em "x" para outro objeto
n.parcel[,1] = as.factor(n.parcel[,1]) #converte a coluna 1 de n.parcel
(que contem o nome dos locais estudados) em objeto da classe fator
n.site = length(unique(n.parcel[,1])) #cria um objeto contendo o numero
de locais estudados
uni = unique(n.parcel[,1]) #cria objeto contendo o nome de cada local
estudado
fr = c() #cria objeto que contera a frequencia relativa de cada especie
por local estudado
fa = c() #cria objeto que contera a frequencia absoluta de cada especie
por local estudado
sp.site = c() #cria objeto que contera o numero de especies presentes em
cada local estudado
area.b = x #atribui o data.frame contido em "x" para outro objeto
area.b$area_basal = pi*(area.b[,4]/2)^2 #cria uma coluna no data.frame
contendo a area basal de cada arvore amostrada
tabela.area = aggregate(area.b$area_basal, by=list(area.b[,1],
area.b[,3]), sum) #cria objeto contendo a area basal total de cada especie
por local estudado
dor = c() #cria objeto que contera a dominancia relativa de cada especie
por local estudado
niveis = levels(unique(x[,3])) #cria objeto contendo o nome de todas as
especies presentes no data.frame de "x"
resultados = new.env() #cria um environment que contera uma matriz para
cada local estudado, contendo DR, FR, DoR e os indices solicitados pelo
usuario das especies
##Calculo de DR, FR, FA e DoR
for(i in 1:n.site) #abre ciclo para o calculo de DR, FR e DoR em cada
local estudado
{
  ##calculo de frequencia relativa de especie por local estudado
  tabela.parcel = n.parcel[n.parcel[,1] == uni[i],] #cria objeto
contendo parcelas, especies e dbh de determinado local estudado
  tabela2 = table(tabela.parcel[,3], tabela.parcel[,2]) #cria objeto
contendo as parcelas de determinado local em que cada especie foi amostrada
  multip = dim(tabela2)[1]*dim(tabela2)[2] #cria objeto contendo o
numero de itens presentes na tabela2
  for(j in 1:multip) #abre ciclo para inserir "1" em cada parcela onde
cada espécie está presente
```

```

        {
            if(tabela2[j] != 0) #verifica se há espécie presente na parcela
            {
                tabela2[j] = 1 #insere "1" caso haja especie presente na
parcela
            }
        }
        tabela3 = apply(tabela2,1,sum) #cria objeto contendo o numero de
parcelas em que cada especie esta presente
        tabela4 = data.frame(tabela3, rep(uni[i], length(tabela3))) #cria um
data.frame contendo o numero de parcelas em que cada especie esta presente e
uma coluna identificando o local de origem destas especies
        parcelas = length(unique(tabela.parcel[,2])) #contabiliza o total de
parcelas presentes em determinado local
        tabela4[,1] = tabela4[,1]/parcelas #calcula a frequencia absoluta de
cada especie em determinado local
        tabela5 = tabela4[,1]*100 #tranforma a frequencia absoluta em
porcentagem e atribui para um novo objeto
        soma.freq = sum(tabela4[,1]) #soma as frequencias absolutas de todas
as especies
        tabela4[,1] = (tabela4[,1]/soma.freq)*100 #calcula a frequencia
relativa de cada especie em determinado local
        fr = c(fr, tabela4) #cria objeto contendo a FR calculada em todos os
locais
        fa = c(fa, tabela5) #cria objeto contendo a FA calculada em todos os
locais
        sp = length(unique(tabela.parcel[,3])) #contabiliza o numero de
especies presentes em determinado local
        sp.site = c(sp.site, sp) #objeto contendo o numero total de especies
presentes em cada um dos locais estudados
        tabela.areal = tabela.area[tabela.area$Group.1 == uni[i],] #cria
objeto contendo a area basal das especies de determinado local estudado
        tabela.areal$x = (tabela.areal$x/sum(tabela.areal$x))*100 #cria
coluna contendo a DoR das especies de determinado local estudado
        dor = c(dor, tabela.areal) #cria objeto contendo a DoR calculada em
todos os locais
        novo = rep(NA,length(unique(x[,3]))) #cria objeto que contera a DoR
das especies que estao presentes em determinado local e o valor "0" para as
que não estão presentes (para possibilitar a montagem da matriz final)
        n = 1 #cria contador
        faltantes = length(unique(x[,3]))-length(dor[i*3-1]$Group.2)
#calcula o numero de especies faltantes em cada local (considerando o numero
total de especies em todos os locais)
        dor[i*3-1]$Group.2 = as.vector(dor[i*3-1]$Group.2) #converte o fator
contendo as especies de cada local em um vetor
        dor[i*3-1]$Group.2 = c(dor[i*3-1]$Group.2, rep("null", faltantes))
#insere a palavra "null" em numero de vezes correspondente a quantidade de
especies faltantes
        for(j in 1:length(unique(x[,3]))) #abre ciclo para inserir "0" como
DoR das especies que nao estao presentes no local
        {

```

```
        if(niveis[j] != dor[i*3-1]$Group.2[n]) #caso a especie nao
esteja presente no local
        {
            novo[j] = 0 #insere um 0 no vetor caso a especie nao esteja
presente no local
        }
        else #caso a especie esteja presente no local
        {
            novo[j] = dor[i*3]$x[n] #insere a sua DoR correspondente no
vetor
            n = n+1 #soma 1 no contador para mudar a posicao de
comparacao do "if"
        }
    }
    dor[i*3]$x = novo #atribui o conteudo do vetor contendo zeros para
as especies faltantes do local estudado ao objeto contendo a DoR de todas as
especies em todos os locais
    local1 = matrix(c(dr[,i], fr[i*2-1]$tabela3, dor[i*3]$x), ncol = 3,
nrow = length(unique(x[,3])), dimnames = list(rownames(dr), c("DR (%)", "FR
(%)", "DoR (%)"))) #cria matriz contendo a DR, FR e DoR de determinado local
    ##Calculo dos indices IVI e/ou IVC
    if(index == "both" | index == "ivi") #verifica se o usuario
selecionou both ou ivi em "index"
    {
        local1.ivi = apply(local1, 1, sum) #calcula o ivi das especies
de determinado local
        local1.ivi.por = local1.ivi/3 #cria um objeto contendo o ivi
relativo de cada especie
    }
    if (index == "both" | index == "ivc") #verifica se o usuario
selecionou both ou ivc em "index"
    {
        local1.ivc = apply(local1, 1, sum)-local1[,2] #calcula o ivc das
especies de determinado local
        local1.ivc.por = local1.ivc/2 #cria um objeto contendo o ivc
relativo de cada especie
    }
    if (index == "both") #verifica se o usuario selecionou both para
"index"
    {
        local2 = matrix(c(dr[,i], fr[i*2-1]$tabela3, dor[i*3]$x,
local1.ivi, local1.ivi.por, local1.ivc, local1.ivc.por), ncol = 7, nrow =
length(unique(x[,3])), dimnames = list(rownames(dr), c("DR (%)", "FR (%)",
"DoR (%)", "IVI", "IVIR (%)", "IVC", "IVCR (%)))) #cria matriz contendo DR,
FR, DoR e ambos os indices para cada especie de determinado local
    }
    else if (index == "ivi") #caso o usuario queira somente o ivi
    {
        local2 = matrix(c(dr[,i], fr[i*2-1]$tabela3, dor[i*3]$x,
local1.ivi, local1.ivi.por), ncol =5, nrow = length(unique(x[,3])), dimnames
```

```

= list(rownames(dr), c("DR (%)", "FR (%)", "DoR (%)", "IVI", "IVIR (%)"))
#cria matriz contendo DR, FR, DoR e IVI das especies de determinado local
}
else #caso o usuario queira somente o ivc
{
  local2 = matrix(c(dr[,i], fr[i*2-1]$tabela3, dor[i*3]$x,
local1.ivc, local1.ivc.por), ncol = 5, nrow = length(unique(x[,3])),
dimnames = list(rownames(dr), c("DR (%)", "FR (%)", "DoR (%)", "IVC", "IVCR
(%)"))) #cria matriz contendo DR, FR, DoR e IVC das especies de determinado
local
}
assign(paste(uni[i]), local2, envir = resultados) #atribui a matriz
contendo todos os indices calculados no environment criado anteriormente
}
matrizes = as.list(resultados) #cria uma lista com as matrizes contidas
no environment
##Geracao dos diagramas de frequencia absoluta
if(diagram) #se o usuario optou por obter diagramas de frequencia
absoluta das especies em cada local
{
  if(save.diagram) #se o usuario optou por salvar os diagramas criados
no diretorio de trabalho
  {
    jpeg(filename = "frequency_diagram.jpg", width = 1000, height =
1000, units = "px", pointsize = 12, quality = 100, bg = "white", res = 144)
#abre dispositivo jpeg para salvar os diagramas
  }
  s = 1 #cria contador
  par(mfrow=c(dim.diagram[1],dim.diagram[2])) #permite construir uma
figura com mais de um diagrama (com numero de linhas e colunas fornecidos
pelo usuario)
  for(q in 0:(n.site-1)) #abre ciclo para contruir um diagrama de FA
para cada local
  {
    n.sp.site = fa[s:(s+(length(unique(x[,3])))-1)] #cria objeto
contendo a frequencia absoluta das especies em determinado local
    n.sp.site1 = n.sp.site[n.sp.site!=0] #retira as especies que
possuem FA igual a 0 (ou seja, nao estao presentes no local)
    hist(n.sp.site1, main = paste("Frequency Diagram - Location:" ,
uni[q+1]), col="black", border = "white", breaks = c(0, 20, 40, 60, 80,
100), xlab = "Absolute frequency (%)", ylab = "Number of species") #cria
histograma de frequencia absoluta das especies
    s = s+length(unique(x[,3])) #soma um valor ao contador
  }
  if(save.diagram) #caso o usuario tenha optado por salvar o diagrama
em um arquivo de imagem
  {
    dev.off() #fecha o dispositivo de imagem aberto
  }
}
return(matrizes) #retorna as matrizes de cada local com os indices

```

```
solicitados (resultado da funcao)
    }
##fim
```

Arquivo da função

[eco_imp.r](#)

Página de ajuda

eco.imp package:unknown R Documentation

Função para calcular a densidade relativa, a frequência relativa, a dominância relativa, o Índice de Valor de Importância e o Índice de Valor de Cobertura de espécies arbóreas.

Description:

Esta função permite ao usuário a comparação entre comunidades florestais quanto à importância ecológica das espécies que as compõem. Esta comparação poderá ser feita através do Índice de Valor de Importância - IVI, do Índice de Valor de Cobertura - IVC ou de ambos. Além disso, a função gerará diagramas de frequência para cada comunidade florestal estudada. Estes diagramas são histogramas que exibem o número de espécies em cada classe de frequência absoluta (porcentagem de parcelas que apresentam determinada espécie).

Usage:

```
eco.imp(x, index = "both", diagram = TRUE, dim.diagram = NULL, save.diagram = FALSE)
```

Arguments:

x Conjunto de dados de entrada. Deve ser um dataframe.

index Índice que deve ser calculado. Pode-se escolher "both", "ivi" ou "ivc".

diagram Define se deverão ser gerados diagramas de frequência absoluta para cada comunidade florestal estudada. Pode ser TRUE ou FALSE.

dim.diagram Obrigatório quando diagram = TRUE. Define em quantas linhas e colunas os diagramas gerados serão distribuídos. A entrada deve ser um vetor numérico contendo o número de linhas e o número de colunas, respectivamente.

`save.diagram` Caso TRUE, salva o diagrama gerado em um arquivo de imagem com formato jpeg no diretório de trabalho.

Details:

O dataframe de entrada no argumento `x` deve conter colunas com as informações coletadas em campo, na seguinte ordem:

1ª coluna: identificação dos locais/comunidades de coleta de dados.

2ª coluna: identificação das parcelas amostradas.

3ª coluna: identificação das espécies arbóreas e quantidade de indivíduos amostrados.

4ª coluna: Diâmetro a altura do peito de cada indivíduo arbóreo amostrado.

Caso `index = "ivi"`, será calculado o Índice de Importância Ecológica das espécies. Para `index = "ivc"`, será calculado o Índice de Valor de Cobertura das espécies. Caso `index = "both"`, ambos os índices serão calculados.

Value:

A função retorna uma lista contendo uma matriz para cada comunidade/local estudado.

Cada matriz conterá a densidade relativa, a frequência relativa, a dominância relativa, o Índice de Valor de Importância e/ou o Índice de Valor de Cobertura das espécies do local.

Author(s):

Silvania Goularte Correia (silvania.goularte@gmail.com)

References:

Curtis, J. T.; McIntoshi, R. P. An Upland Forest Continuum in the Prairie-Forest Border Region of Wisconsin. *Ecology*, 32(1): 476-496, 1951.

Instituto Ambiental do Paraná. Parâmetros Fitossociológicos. Disponível em: http://www.iap.pr.gov.br/arquivos/File/Plano_de_Manejo/PE_Rio_Guarani/Anexos/anexo_2.pdf. Acesso em: 02 abr. 2015.

Examples:

Faça download do arquivo `exemplo.csv`.

```
tabela = read.table("exemplo.csv", header=TRUE, sep=";")
eco.imp(tabela, "both", TRUE, c(2,3), TRUE) #Gera matrizes contendo os
índices solicitados e salva os diagramas de frequência absoluta solicitados.
eco.imp(tabela, "both", TRUE, c(2,3), FALSE) #Gera matrizes contendo os
índices solicitados e exibe na tela os diagramas de frequência absoluta
solicitados.
```

[exemplo.csv](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:silvaniagc:start



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