

RENAN PARMIGIANI

Mestrando Em ecologia no Instituto de Biociências, no Laboratório de Ecologia de Floresta Tropical (LabTrop)



A minha pesquisa consiste em analisar qual as características dos indivíduos de uma comunidade vegetal sob um gradiente de estresse. Contato/; renanparmigiani@hotmail.com

Meus Exercícios

[Exercício 1](#)

[Exercício 4](#)

[Exercício 5](#)

[Exercício 7.2](#)

[Exercício 7b](#)

[Exercício 8](#)

[Exercício 9](#)

Proposta de Trabalho Final

Plano A

Função arborizar: seleção de árvore mais apropriada para plantio em vias urbanas.

A arborização urbana é uma ação que demanda muito planejamento, começando na escolha de espécie com características que não causem danos a instalações urbanas presentes ao redor. Para um plantio apropriado há guias que especificam quais espécies usar segundo uma série de parâmetros, um deles é o Manual Técnico de Arborização Urbana (disponível no link https://www.sosma.org.br/wp-content/uploads/2015/03/MANUAL-ARBORIZACAO_22-01-15_.pdf), o qual possui a Chave Arborizar (com um funcionamento similar a uma chave de identificação de táxons) com o intuito de avaliar a melhor espécie a ser plantada levando em consideração quais as estruturas urbanas presentes ao redor do local de plantio e as características da árvore (porte, arquitetura, sistema radicular) A minha proposta é elaborar uma função que acompanhe esse manual de modo a automatizar a escolha da espécie mais apropriada para o plantio em uma via urbana. A entrada será um data frame com cada linha representando um lugar onde se deseja plantar uma árvore e as colunas com as seguintes especificações do local: largura da calçada (em metros), presença/ausência de fiação elétrica, fiação é compacta ou convencional, a distância em relação aos seguintes equipamentos e mobiliários urbanos (postes, hidrantes, esquinas, instalações subterrâneas,

mobiliário urbano, galerias, caixa de inspeção, acesso de pedestres). A saída da função será a espécie mais apropriada para ser plantada em cada lugar, e a quantidade de mudas necessárias de cada grupo (segundo a divisão das tabelas 5, 6 e 7 do manual, pg 30-37).

Oi, Renan!

Sei que vc está emocionalmente apegado à proposta A (e eu também estou, confesso), mas estou achando que ela vai te dar muito trabalho braçal (pra transferir os dados da chave e do anexo pra dentro da sua função) e te permitir explorar pouco as funções que vocês aprenderam ao longo do curso. Prevejo você usando muitos loops (mais precisamente 27), e alguma coisa de indexação, mas nada mais. Vamos tentar deixar a proposta B redonda? Proponho sentarmos um dia e fazermos a proposta A mesmo assim, pra gente enviar pros autores do documento da prefeitura! Você fica mais feliz? Ou não supera e tá afim de argumentar pra fazer a proposta A agora? To aberta a isso, hehe!

—[Luísa Novara](#)

Plano B

Função les: análise da correlação de atributos funcionais

O espectro da economia foliar é uma relação que há entre diversos atributos funcionais das plantas (massa por área, taxa de assimilação fotossintética, % de nitrogênio, % de fósforo, entre outros), sendo que muitos desses parâmetros têm seus valores correlacionados. A minha função tem como objetivo avaliar a variação da correlação entre esses atributos para diferentes comunidades. A entrada será um data.frame com as seguintes colunas: identificação do indivíduo, espécie, local, demais colunas com os atributos funcionais. A resposta da função será: se há diferença para algum atributo entre as comunidades, se há diferença em algum atributo para alguma espécie entre os diferentes lugares, qual a correlação entre os atributos para cada comunidade e se essa correlação varia entre os lugares.

Renan, como eu disse acima, acho que se vc desenvolver essa proposta B vc vai aprender mais! Mas ainda falta esclarecer algumas coisas.

- Como vc irá avaliar se há diferenças nos atributos entre diferentes localidades (tanto o geral pra comunidade quanto por espécie)?
- Como vc irá avaliar se há diferença nas correlações entre as diferentes localidades?

- Sugiro que vc pense em uma saída gráfica, além do data frame com os dados. Gráficos bons nos ajudam a entender nossos dados!
- Ainda tenho dúvidas em relação à “teoria” do espectro da economia foliar. Tentei baixar o artigo do Wright (2004) da Nature, mas não consegui (mesmo com o VPN!). Vc pode me passar? Quero entender direito essa ideia pra conseguir te ajudar melhor. Estou com uma dúvida: o que vc faz/conclui a partir das diferenças nas correlações entre diferentes comunidades (como vê, minha dúvida vem do fato de eu não conhecer a LES)? Não que vc precise concluir algo dentro da função (hehe), mas explicar melhor o contexto dela ajuda a entender sua importância.

Se tiver qualquer dúvida, escreva aqui, que respondo novamente! Por favor, quando vc atualizar qualquer coisa aqui, me mande um e-mail, pra agilizarmos nossa comunicação.

Beijo

—[Luísa Novara](#)

Plano B reformulado

Lendo as suas considerações resolvi melhorar a proposta B, segue a minha proposta reformulada. Busquei mudar algumas coisas para que ela fizesse mais sentido com a economia do espectro foliar.

O espectro da economia foliar é uma relação que há entre diversos atributos funcionais das plantas (massa por área, taxa de assimilação fotossintética, % de nitrogênio, % de fósforo, entre outros), sendo que muitos desses parâmetros têm seus valores correlacionados. O objetivo da minha função é construir o espectro dos atributos funcionais de uma comunidade que varie de 0 a 1 a partir dos valores extremos dos dados de entrada. Os valores próximos a zero representam indivíduos que possuem uma estratégia de rápido retorno energético pelo seu investimento (estratégia r), e os valores próximos a um representam indivíduos que possuem uma estratégia de retorno energético a longo prazo (estratégia K). Ou seja, a minha função irá transformar as regressões significativas em um índice representando o espectro dos atributos.

O índice será calculado atribuindo os valores de 0 e 1, ou 1 a 0 caso aquele atributo esteja relacionado negativamente com um atributo diretamente relacionado a K, para os extremos de cada atributo, o índice total do indivíduo será calculado através da média de cada atributo.

A entrada será um data.frame com as seguintes colunas: identificação do indivíduo, características dos indivíduos que não serão calculados como atributos funcionais (espécie, local, grupo funcional etc) demais colunas com os atributos funcionais. A função terá as seguintes saídas: gráficos mostrando a correlação par a par das características, junto com os coeficientes para cada regressão; o data.frame de entrada com mais uma coluna indicando o valor do índice calculado; uma

representação gráfica unidimensional do índice dando destaque para um grupo de indivíduo escolhido pelo usuário (espécie, local, grupo funcional).

Os argumentos da função serão:

x: o data.frame de entrada;

caracteristica: um vetor numérico indicando quais colunas das características dos indivíduos que não serão calculados como atributos funcionais;

atributo: um vetor numérico indicando quais colunas são referentes aos atributos funcionais que serão calculados;

K: nome da coluna do data.frame que possui os valores que serão positivamente relacionados ao índice;

r: nome da coluna do data.frame que possui os valores que serão negativamente relacionados ao índice (só é necessário colocar um dos argumentos r ou K);

destaque: um vetor com o nome da coluna que possui algum fator de interesse a ser destacado na saída gráfica do índice e o(s) nome(s) do(s) fator(es).

Oi Renan, Legal, acho o plano B mais promissor também. Eu só não sei se entendi bem alguns pontos. Nas saídas gráficas a função compara os valores entre espécies ou grupos funcionais, dependendo do que o usuário pede? Não entendi como isto fica para a correlação par a par (alias que correlações são essas?) para o coeficiente de regressão. Se você vai fazer correlações, por que calcular depois o coeficiente de regressão? Não seria o coeficiente de correlação? — [Sara Mortara](#)

Oi, Sara! (: Eu e o Renan conversamos na sexta-feira (29) sobre o plano B reformulado dele por hangouts! Mas pedi para ele postar as coisas aqui, e responder as perguntas que vc fez. Ele logo deve fazer isso! Obrigada! —[Luísa Novara](#)

Olá Sara, a minha idéia é que num primeiro momento a função avalie, dentre os atributos colocados pelo usuário, quais atributos estão significativamente correlacionados (de fato o valor que será apresentado será o coeficiente de correlação e não o coeficiente de regressão), e se essa correlação é positiva ou negativa. Estava pensando em fazer

esses testes de correlação de uma das duas formas (ou talvez de alguma outra que ainda venha a pensar):

1: entre todos os atributos comparando-os par a par para ver quais estão significativamente correlacionados.

2: como haverá um atributo chave (que foi colocado pelo usuário como a referência da estratégia K ou r), pensei em avaliar a relação de todos os atributos em relação a esse, e levar em consideração só os atributos que tiveram uma correlação significativa.

E essa seria uma das janelas gráficas mostrando essas correlações para todos os indivíduos

Uma vez sabendo quais os atributos significativos os usarei na construção do meu índice por indivíduo. Quando eu tiver o valor do índice para cada indivíduo a função irá ver qual o destaque que o usuário quer dar e irá representar em uma janela gráfica da seguinte maneira: um gráfico com um eixo que representa o índice de 0 a 1, nesse eixo estará a “nuvem” de pontos representando cada indivíduo, dependendo de qual característica (lugar, espécie, modo de vida etc) esses pontos irão ficar em diferentes cores de modo que o usuário possa ver em qual intervalo do espectro está determinado grupo de interesse.

Bem como você pode ver ainda estou em dúvida em relação a quais comandos usarei, vou tentar fazer um esquema do funcionamento da minha função para me dar mais uma noção de como será o meu script.

—[Renan Parmigiani](#)

Oi, Renan! Você chegou a alguma conclusão sobre para quais atributos calcular a correlação? Talvez faça mais sentido comparar os atributos com aquele que o usuário indica ter certeza estar relacionado a estratégias r e K, para evitar colocar no cálculo do índice atributos que não tenham a ver com estratégia de vida propriamente.

Acho que fazer um esquema do funcionamento da função é importante! Para me ajudar a perceber se está de fato tudo ok (e te ajuda mt tb!) e onde podemos melhorar. Beijo!

—[Luísa Novara](#)

Olá monitores queridos, fiz um esquema de como será a minha função em um fluxograma, me desculpem a falta de formalismo e rigor na construção do fluxograma mas algumas coisas não sabia exatamente como representar, mas acredito que elas estão com um certo grau de intuitismo, além do mais fiz algumas notas explicando o que significa/o que estou querendo fazer no próprio fluxograma. Abraços

[arquivo em PDF](#) [arquivo em ponto poderoso\(PowerPoint](#)

—*Renan Parmigiani*

Código da Função

```
#####  
####FUNÇÃO####  
#####  
  
les<- function (x, atr,K , r, destaque)                                #Criando a função  
{  
  ##Fazendo uma análise exploratória nos argumentos  
  if(missing(K) & missing(r))                                          #Analisando se o  
usuário colocou o arg r ou K  
  {  
    stop("Necessário informar o argumento K ou r")                    #Retornando  
uma msg caso o arg r ou K esteja faltando  
  }  
  if(missing(x))                                                        #Analisando se o usuário colocou  
o arg x  
  {  
    stop("Necessário informar o data.frame de entrada no argumento  
x")#Retornando uma msg caso o arg x esteja faltando  
  }  
  if(missing(atr))                                                      #Analisando se o usuário colocou  
o arg atr  
  {  
    stop("Necessário informar quais são os atributos a serem analisados no  
argumento atr") ##Retornando uma msg caso o argumento atr esteja faltando  
  }  
  ##Criando objeto que será trabalhado dentro da função  
  atr.df<- data.frame(x[atr])                                           #Transpondo os  
atributos do data.frame x para o data.frame atr.df  
  ##Analisando qual será o tamanho da janela gráfica que será necessária para  
plotar as relações entre os atributos  
  tam <- length(atr)                                                    #Guardando a
```

```

quantidade de atributos que o usuário irá analisar
  lado<- sqrt(tam)                                #Tirando a raiz
da quantidade de atributos para montar a janela gráfica
  hor <- ceiling(lado)                            #A quantidade
de gráficos na horizontal sempre será o valor da raiz arredondado para cima
  ver <- round(lado)                              #A quantidade
de gráficos na vertical sempre será o valor da raiz arredondada segundo a
regra de arredondamento
  x11()                                           #Abrindo janela
gráfica
  par(mfrow = c(hor,ver), bty = "l")            #Essa relação
vai ajustar os gráficos da forma mais próximo do ótimo para que eles não
fiquem pequenos e ao mesmo tempo não sobrem "janelas vazias"
##Vendo quais atributos estão correlacionados com o K, e conversão dos
valores dos atributos para o índice
  if(!missing(K))                                #Vendo se o
usuário usou o arg K para indicar alguma estratégia
  {
    indice= data.frame(rep(NA,length(atr.df[,1]))) #Criando um
data.frame para guardar os valores do índice do ciclo
    for (i in 1:tam)                             #Iniciando o
ciclo que irá fazer os teste entre os atr's e o atr K
    {
      mod <- lm(K~atr.df[,i])                    #Criando o
modelo
      suppressWarnings(teste <- anova(mod))       #Testando se o
modelo é significativo, coloquei o comando suppressWarnings, pois um dos
modelos irá analisar a relação entre a mesma coluna do data.frame de
entrada, e nessa situação há a saída de uma warning message que não é
necessária
      nome.y <- colnames(atr.df)[i]              #Criando o nome que
será atribuído ao eixo y no plot do modelo
      titulo<- c("Valor do p" , round(teste$`Pr(>F)` , 3)) #Criando o
nome que será atribuído ao título no plot do modelo que é o valor de p no
teste anova
      plot(atr.df[,i]~K, ylab= as.graphicsAnnot(nome.y), main =
as.graphicsAnnot(titulo)) #Plotando a relação entre os atr's
      if(teste$`Pr(>F)`[1]<0.05)                 #Vendo se o modelo
é significativo
      {
        if(mod$coefficients[2]>0)                #Vendo se a relação
entre os atributos é positiva
        {
          indice[i]= (atr.df[i]-min(atr.df[i]))/(max(atr.df[i])-
min(atr.df[i])) #Calculando o índice para os atributos positivamente
relacionados ao atr. K
        }
        else                                     #Caso a relação entre os
atributos seja negativa em relação a K
        {
          indice[i]= abs((atr.df[i]-min(atr.df[i]))/(max(atr.df[i])-

```

```

min(atr.df[i]))-1) #Calculando o indice para os atributos negativamente
relacionados ao atr. K
    }
  }
  else                                     #Caso a relação entre os atr não
seja significativa
  {
    indice[i]<- rep(NA, length(atr.df[i]))      #Atribuindo
NA's ao data.frame pois a relação não é siginificativa
  }
}
}
##Vendo quais atributos estão correlacionados com o r, e conversão dos
valores dos atributos para o índice, essa parte é muito parecida com a do K,
só invertendo alguns sinais
  else                                     #Caso o usuário tenha usado o arg. r
para indicar alguma estratégia
  {
    indice= data.frame(rep(NA,length(atr.df[,1])))      #Criando um
data.frame para guardar os valores do índice do ciclo
    for (i in 1:tam)                                  #Iniciando o ciclo
que irá fazer os teste entre os atr's e o atr r
    {
      mod <- lm(r~atr.df[,i])                          #Criando o
modelo
      suppressWarnings(teste <- anova(mod))              #Testando se o
modelo é significativo, coloquei o comando suppressWarnings, pois um dos
modelos irá analisar a relação entre a mesma coluna do data.frame de
entrada, e nessa situação há a saída de uma warning message que não é
necessária
      nome.y <- colnames(atr.df)[i]                     #Criando o nome que
será atribuido ao eixo y no plot do modelo
      titulo<- c("Valor do p" , round(teste$`Pr(>F)` , 3))      #Criando o
nome que será atribuido ao título no plot do modelo que é o valor de p no
teste anova
      plot(atr.df[,i]~r, ylab = as.graphicsAnnot(nome.y), main =
as.graphicsAnnot(titulo)) #Plotando a relação entre os atr's
      if(teste$`Pr(>F)`[1]<0.05)                          #Vendo se o modelo
é significativo
      {
        if(mod$coefficients[2]<0)                          #Vendo se a relação
entre os atributos é negativa
        {
          indice[i]= (atr.df[i]-min(atr.df[i]))/(max(atr.df[i])-
min(atr.df[i])) ##Calculando o indice para os atributos negativamente
relacionados ao atr. r
        }
        else                                               #Caso a relação entre os
atributos seja positiva em relação a r
        {

```



```

        indice[i]= abs((atr.df[i]-min(atr.df[i]))/(max(atr.df[i])-
min(atr.df[i]))-1) #Calculando o indice para os atributos positivamente
relacionados ao atr. r
    }
}
else                                     #Caso a relação entre os atributos
não seja significativa
{
    indice[i]<- rep(NA, length(atr.df[i]))          #Atribuindo
NA's ao data.frame pois a relação não é significativa
}
}
}
##Ajuste do data.frame indice
indice <- indice[,colSums(is.na(indice))<nrow(indice)]      #Excluindo as
colunas com NA's (que não foram significativa)
if(class(indice) == "numeric")                #Vendo se houve correlação
com pelo menos mais um atributo para a construção do índice
{
    stop("0 atributo referente a estratégia não se relaciona com nenhum
outro, insira outro atributo como referência") #Retornando ao usuário o
aviso de falta de correlação
}
indice$media<- apply(indice, 1, mean)          #Tirando a média do
índice dos atr's
x$indice <- indice$media                      #atribuindo o valor da
média a mais uma coluna do data.frame de entrada
if(missing(destaque))
#Analisando se o usuário quis destacar os valores do índice para categorias
dos indivíduos do data.frame
{
    return(x)                                #Retornando apenas o valor do
índice
}
##Construção do boxplot com os valores do índice por categoria
as.factor(destaque)                          #Convertendo a coluna de
destaque em fatores
x11()                                         #Abrindo uma janela gráfica para
expor ao usuário o valor do índice em relação a característica de destaque
par(mfrow = c(1,1), bty = "l")              #Alterando os parâmetros
gráficos
boxplot(x$indice~destaque, ylab = "Indice de r a K") #Abrindo janela
gráfica
return(x)                                    #Retornando ao usuário o data.frame
de entrada com a coluna a mais com o valor do índice
}

```

Página de Ajuda

les package: nenhum R Documentation

Cálculo de um índice de estratégia de vida para indivíduos utilizando seus atributos funcionais. O índice varia de 0 (para r estrategistas) a 1 (para K estrategistas).

Description:

A função `les` atribui um valor de índice através dos atributos funcionais que indicará se esse indivíduo tende para uma estratégia r ou K . A construção desse índice é em cima dos próprios atributos funcionais que o usuário coloca. Sendo possível avaliar em que região do índice está determinado grupo de indivíduo através do gráfico `boxplot` que a função retorna

Usage:

```
les<- (x, atr,K , r, destaque)
```

Arguments:

`x` data.frame. com os dados dos indivíduos
`atr` vetor numérico contendo as colunas com os atributos que serão usados na construção do índice
`K` coluna do data.frame `x` que será usado para indicar qual atributo cresce positivamente relacionada a uma estratégia K .
`r` coluna do data.frame `x` que será usado para indicar qual atributo cresce positivamente relacionada a uma estratégia r .
`destaque` coluna do data.frame `x` que contém alguma variável que separe os indivíduos em categorias para especificar em que posição do índice cada grupo está localizado na saída gráfica

Details:

Segundo Wright et al. (2004) existe uma série de atributos foliares que são correlacionadas, de modo que num extremo são encontradas plantas com estratégias de rápido retorno energético por investimento (r estrategistas) e no outro de retorno mais a longo prazo (K estrategistas). A função `les` constrói esse espectro através dos seguintes passos:

1. Análise se os argumentos essenciais estão presentes;
2. Construção de modelos lineares comparando cada coluna dos atributos com o atributo de referência da estratégia K ou r .
3. Teste anova de cada modelo para avaliar a sua significância.
4. Plot da relação entre os atributos e o atributo de referência da estratégia;
5. Análise da correlação positiva ou negativa entre os atributos;
6. Cálculo do índice de cada atributo, através de uma escala em que o maior valor é atribuído o valor 1 e para o menor 0, sendo os intermediários os valores intermediários através de um cálculo de proporção.
6. Cálculo do índice por indivíduo, através da média de cada atributo.

7. Construção do boxplot com os valores do índice por categorias da coluna destaque.

Value:

A função `les` possui os seguintes retornos para o usuário:

Um conjunto de gráficos com a relação entre os atributos e o atributo de referência da estratégia, com cada gráfico tendo valor de p do teste anova de cada modelo.

Um coluna a mais no mesmo `data.frame` de entrada com o valor do índice para cada indivíduo

Um gráfico boxplot com os valores do índice por categorias da coluna destaque.

Warning:

A função é interrompida se o usuário não colocar os argumentos obrigatório (`x`, `atr`, `r` ou `K`), ou caso o atributo não tenha correlação significativa, segundo o teste anova, com nenhum outro atributo.

Author(s):

Renan Parmigiani
renanparmigiani@hotmail.com

References:

Wright I. J., et al. The worldwide leaf economics spectrum. *Nature* 428, 821-827 (2004).

Examples:

```
sp<-rep(c("sp1","sp2","sp3","sp4","sp5"),each = 6)
lugar<- c((rep("A", each = 10)), "B", "B", rep("A", each = 4), rep("B", each =
12), "A", "A")
dap<- rnorm(30, 24, 8.3)
LMA<- sort(runif(30, 10, 1000))
Amass<- sort(runif(30, 9, 400), decreasing = TRUE)
Nmass<- sort(runif(30, 1,10),decreasing = TRUE)
alt<- rnorm(30, 14, 4.3)
LL<- sort(runif(30, 1, 100))

dado= data.frame (sp,lugar, dap, LMA, Amass,Nmass, alt,LL)

les(dado, K = dado$LL, atr = c(3:8), destaque = dado$lugar)
les(dado, r = dado$Nmass, atr = c(3:8), destaque = dado$sp)
les(dado, K = dado$dap, atr = c(3:8), destaque = dado$lugar)
```

Arquivo da Função

[Função les](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:renan.parmigiani:start 

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