



# Polliana

Meu nome é Polliana Zocche de Souza e faço Doutorado em Ecologia, Departamento de Botânica, IB, na Unicamp.

Minha tese está relacionada a estrutura de comunidades herbáceo-arbustivas em dunas costeiras. Sou orientada pelo Prof. Dr. Flavio Antonio Mães dos Santos e Co-orientada pela Prof<sup>a</sup>. Dr<sup>a</sup>. Tânia Tarabini Castellani.

Meu interesse na disciplina está em fazer análises que extraiam mais informações dos dados coletados ao longo do Mestrado e Doutorado para publicações e análises para a própria tese.

Contato e-mail: farbby@gmail.com

## Meus exercícios

exec

## Propostas de Trabalho Final

## Plano A

Introdução: A ocorrência ou não de uma espécie (Y) em um determinado local pode estar condicionada à outra espécie estruturadora (X). Quando esta ocorrência é maior do que esperado ao acaso, podemos analisar se Y tem interações positivas com X. Quando esta ocorrência é menor que o esperado ao acaso, podemos supor que há interações negativas entre as espécies. Assim, esta co-ocorrência com a espécie estruturadora X pode estar ligada à forma de vida das espécies Y, que podem (1) ocorrer preferencialmente onde X está presente ou (2) não ocorrer na presença de X.

Entrada: um data.frame com dados de presença e ausência da espécie estruturadora X, outra com os nomes das espécies da comunidade observadas em locais onde Y tem presença e locais onde X está ausente, suas formas de vida e dados de presença e ausência destas espécies.



O que a função fará? As colunas de espécies e formas de vida serão transformadas em fatores. Coberturas das espécies associadas e da espécie Y serão dados numéricos. A função simulará a ocorrências destas das espécies Y ao acaso, ou seja, sem nenhuma interação interespecífica com a espécie estruturadora X (modelo nulo). A partir disso serão calculadas as frequências e probabilidades de ocorrência das espécies Y. Os dados de probabilidade de ocorrência da simulação serão comparados com as probabilidades de ocorrência da amostra por testes de co-ocorrência (c-score?). Estes resultados serão plotados em um gráfico com histogramas das probabilidades de

ocorrência simulada e observada. Uma tabela (semelhante a um summary também será gerado com os resultados das comparações estatísticas).

Output: gráficos e summary dos resultados estatísticos.

A ideia da função vem da estruturação de comunidades herbáceo-arbustivas de restingas que estudo desde o Mestrado. Nestas comunidades, uma espécie de arbusto clonal se expande por grandes áreas e poucas espécies conseguem se estabelecer onde o arbusto ocorre. Penso então na lógica de competição entre as espécies, onde a co-ocorrência com a espécie arbustiva é pequena. Mas a função abrange qualquer tipo de padrão de co-ocorrência.

## Plano B

Introdução: pensando na temática de estudos que avaliam efeitos alelopáticos de extratos contendo material vegetal (folha, casca, raízes) sobre o crescimento de plântulas. Observar o crescimento de plântulas submetidas a diferentes tipos de extratos e diferentes concentrações e comparar com o crescimento de plântulas do tratamento controle para avaliar o efeito dos extratos.

Entrada: tamanho total por plântula em diferentes tempos, nos tratamentos controle e extratos avaliados.

O que a função fará? Executa uma análise de variância de medidas repetidas e plota um gráfico com as curvas de crescimento em relação ao tempo.

Output: gráfico das curvas. Os resultados da anova podem ser chamados com summary.

## Plano C

Introdução: ao preparar um lanche devemos escolher meticulosamente o que vamos comer para depois não sofrer com a balança depois! Mesmo assim, podemos nos programar para gastar as calorias ingeridas nos exercitando, ao ar livre ou área fechada, e com diferentes graus de dificuldade.

Entrada: um dataset contendo um data.frame com os ingredientes do lanche, com o conteúdo calórico de cada ingrediente e outro data.frame contendo uma lista de atividades físicas, a quantidade de calorias que essas atividades queimam por minuto e o grau de dificuldade serão utilizados.

O que a função fará? o dataset contendo estes dados será carregado. A função terá um argumento onde são colocados os ingredientes do lanche. Em outro argumento será especificado se o exercício é ao ar livre ou área fechada. Um terceiro argumento especifica os níveis de dificuldade dos exercícios. A função somará as calorias dos ingredientes e calculará o tempo de atividade física equivalente à quantidade de calorias ingeridas.

Resultado (output): a lista de atividades, dentro dos parâmetros escolhidos, e o tempo necessário em cada uma delas para gastar a quantidade de calorias consumidas no lanche.

# Comentários

Oi Poliana, A primeira proposta esta bem apresentada e parece interessante. Vc pode usar o C score ou qualquer outro indice de co-ocorrendia como sua estatistica de interesse e fazer a simulacao de cenario nulo para inferir significancia dos valores. Defina brm o cenario nulo!. Note que as interacoes sao apenas inferidas a partir de uma associacao espacial, portanto o que tem é apenas co-ocorrencias, e se sao significativas...poderia ser por outro fator (mesma preferencia de habitat). Deixe que a inferencia sobre interacao seja feita pelo usuario! — [Alexandre Adalardo de Oliveira](#)  
2013/03/26 13:19

## Trabalho Final

## Página de Ajuda

co.occure

package:nenhum

R Documentation

Avalia coocorrência entre uma espécie alvo e outras espécies da comunidade.

Description:

Calcula o c-score da comunidade, utilizando checkboards par a par da espécie alvo de investigação e as outras espécies da comunidade; simula comunidades onde as espécies são distribuídas aleatoriamente (mantendo apenas a ocorrência da espécie alvo) e calcula o c-score da comunidade simulada. Retorna o p-valor e mostra graficamente os resultados.

Usage:

```
co.occure(x, nsim=100)
```

Arguments:

x    numérico    matriz ou data.frame contendo apenas dados numéricos de presença e ausência.

nsim    numérico    número de simulações realizadas para calcular o c-score médio das comunidade simuladas. O default deste parâmetro são 100 simulações.

Details:

A função é voltada à investigação de padrões de coocorrência em comunidades estruturadas pela presença de uma espécie. O c-score é um valor médio dos checkerboards entre a espécie alvo e cada espécie que ocorre pelo menos uma vez na comunidade, obtido de acordo com o cálculo de Stone e Roberts (1990).

O objeto deve conter espécies nas linhas e locais nas colunas. A espécie alvo, estruturadora da comunidade, deve estar na primeira linha. A presença de NAs será verificada e estes serão substituídos por 0, representando ausências. Qualquer valor diferente de zero no objeto de entrada será convertido em presença.

A comunidade simulada mantém apenas a ocorrência original da espécie alvo e rearranja aleatoriamente as ocorrências das outras espécies, sem armazenar nenhum padrão de ocorrência da comunidade observada.

O p-valor retorna a razão entre o número de vezes que o c-score da comunidade simulada foi maior que o c-score da comunidade observada pelo número total de simulações efetuadas (nsim).

Value:

Retorna uma lista com o c-score da comunidade observada, c-score médio das comunidades simuladas, p-valor e histograma da frequência dos c-scores simulados com linhas marcando o c-score observado e intervalo de confiança de 95% para os valores de c-score das comunidades simuladas.

Author(s):

Polliana Zocche de Souza (farbby@gmail.com)  
Estudante de Doutorado em Ecologia – Universidade Estadual de Campinas.

References:

Stone, L. & Roberts, A. 1990. The checkerboard score and species distributions. *Oecologia* 85: 74-79.  
Gotelli, N. J. & Graves, G. R. 1996. Null models in ecology. Smithsonian Institution Press, Washington & London.  
Ecological Society of America. Ecological Archives. Gotelli, N. J. Help online software EcoSim 5.0.  
<http://www.esapubs.org/archive/ecol/E081/022/EcoSim%20Help/CoOccur/CoOccurrence.htm>

See Also:

C.score() do pacote bipartite.

Examples:

```
#Data.frame presença de NAs
comu1=data.frame(t(matrix(data=c(rep(c(0,1), each=10),sample(c(0,1, NA),
200, replace=TRUE))),ncol=11)))
co.occure(comu1)

#Matriz
comu2=matrix(data=c(rep(c(0,1), each=5), sample(c(0,1), 100, replace=TRUE)),
ncol=10, byrow=TRUE)
co.occure(comu2)

# Comunidade com competição
comu3<- matrix(ncol=10, nrow=10, data=0,
dimnames=list(paste("sp",1:10,sep=""),
paste("site",1:10,sep="")))
comu3[1,sample(1:10,5)] <- 1
for(i in 1:50){
comu3[sample(2:10,1),sample((1:10)[as.numeric(comu3[1,])!=1],1)] <- 1 }
co.occure(comu3)
```

## Código da Função

```
# Função coocorrência de espécies
# Comunidade com a primeira linha sendo a espécie alvo
co.occure=function(x, nsim=100)
{ ##### Informes sobre a função
  cat("\n\t Esta função analisa co-ocorrência entre uma espécie alvo e o
  restante da comunidade
      \n\t A espécie alvo deve estar na primeira linha
      \n\t O objeto 'x' deve ser matriz ou data.frame formado por caracteres
numéricos\n\n")
  if(class(x)=="data.frame"|class(x)=="matrix") # Tipo de objeto
operado pela função
  { ##### Manipulação dos NAs
    if(sum(is.na(x))>1) # Se a soma de NAs
for > 1
    {
      nas=sum(is.na(x)) # Número de NAs que
serão trocados
      cat("\t", nas , " NA foram substituídos por 0\n") # Mostra na tela qts
NAs foram substituídos
      x[is.na(x)]=0 # Trocar NAs por zero
      x[x!=0]=1 # Tudo que não for 0
vira 1
      x1=x
    } else {
      x[x!=0]=1 # Tudo que não for 0
vira 1
      x1=x
    }
  }
```

```
} else {                                # Avisa que o objeto não é data.frame ou
matriz
  stop("O objeto não é um data.frame. Corrija esta condição") # Mensagem
de erro
}
##### C-score da Comunidade observada
#####
n.row=length(x1[,1])                    # Total de linhas (espécies)
s=c()                                    # Vetor que armazenará os resultados do
for
  for (i in 2:n.row)                    # S numero de locais em comum
  {                                      # Cálculo do c-score par a par com a sp
alvo (início)
    s[i]=sum((x1[1,]==1)&(x1[i,]==1)) # Soma dos locais onde as duas
espécies estão presentes
  }
  ri=sum(x1[1,]==1)                    # Soma das presenças para a sp i
  rj=c()                                # Vetor que armazenará as somas das
presenças das spp rj
  for(i in 2:n.row)
  {
    rj[i]=sum(x1[i,]==1)                # Soma das presenças por espécie para
todas espécies da comunidade
  }
  c.score = (ri - s)*(rj - s)           # C-score par a par com a espécie alvo
  c.score=round(mean(c.score[-1]),1)# Retirando o 1o valor (NA) e calculando
o C-score da comunidade
##### C-score da comunidade simulada
#####
c.score.nsim=c()
for(j in 1:nsim)
{
  sp.alvo=x1[1,]                        # Vetor da espécie alvo
  com.sim=data.frame(t(apply(x1[-1,], 1, sample))) # Comunidade sem a
espécie alvo
  n.row=length(com.sim[,1])             # Total de linhas (espécies)
  s=c()                                  # Vetor que armazenará os resultados do for de s
  for (i in 1:n.row)                    # Calcula o número de locais em comum
de ri com cada rj
  {
    s[i]=sum((sp.alvo==1)&(com.sim[i,]==1)) # Soma locais onde as duas spp
estão presentes
  }                                     # Calcula o número de locais em comum
de ri com cada rj
  ri=sum(sp.alvo==1)                    # Soma das presenças para a sp i
  rj=c()                                # Vetor que armazenará as somas das
presenças das spp rj
  for(i in 1:n.row)                    # Calcula o número de presenças de
cada rj
  {
```

```

      rj[i]=sum(com.sim[i,]==1)      # Soma das presenças por rj para todas
espécies da comunidade
    }                                # Calcula o número de presenças de
cada rj
    score.sim=(ri - s)*(rj - s)      # C-score par a par com a espécie alvo
    c.score.nsim[j]=mean(score.sim)  # Vetor que guarda os c-scores
simulados "nsim" vezes
  }
  c.score.super=c.score.nsim        # Vetor igual para calcular o p-
valor
  c.score.mean.sim=round(mean(c.score.nsim),1) # C-score médio simulado
arredondado para 1 casa
  ##### p-valor #####
  soma.c.score.super=sum(c.score.super<c.score)
  p.valor=soma.c.score.super/nsim
  p.valor
  cat("\n p(observed >= expected) = 1 \n p(observed <= expected) = 0\n\n") #
Mensagem explicando o p
  ##### Gráfico #####
  par(mar=c(5,4,3,2))
  hist=hist(c.score.nsim, main="", col="grey", xlab="C-score \n Comunidade
simulada",
            ylab="Frequência")      #Histograma de frequência dos c-scores das
comunidades simuladas
  abline(v=c.score, col="blue")     # C-score observado
  abline(v=(quantile(c.score.nsim, probs=c(0.025,0.975))), col="red")
#Intervalo confiança C-score da comunidade simulada
  legend("topright", legend=c(paste("Observado \n",round(c.score,1)), "I.C.
Simulado"), bty="n",
        pch=15, col=c("blue", "red"))
  ##### Resultados #####
  resultados=list(c.score,c.score.mean.sim, p.valor) # Resultados que
retornarão para o usuário
  names(resultados)=c("C-score da comunidade observada", "C-score da
comunidade simulada","p-valor")
  return(resultados)                # Comando que retorna
resultados
}

```

## Arquivo da Função

[co.occu.r](#)

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2013:alunos:trabalho\\_final:fabrbby:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:fabrbby:start) 

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