

Carolina Farhat

Mestranda no departamento de Zoologia, IB-USP



Projeto sobre filogeografia comparada de duas espécies de lagartos que possuem distribuições disjuntas e sobrepostas na região da Bacia Amazônica. Tenho como objetivo testar cenários demográficos para explicar tais padrões de distribuição através de reconstruções filogenéticas com dados moleculares e cálculos de estatísticas filogeográficas.

Exercícios aulas

Link para a página com meus [Exercícios](#) resolvidos.

Proposta trabalho final

Plano A: Receita de bolo

Quem nunca pensou em vender uma comidinha caseira para agregar na renda? Você quer vender, mas se perde na hora de fazer o cálculo de ingredientes e estipular assim um valor justo para seu produto (com lucro, claro). O acesso à receita é fácil, sabemos quais e quanto de cada ingrediente; porém na hora de fazer as compras cada um vem em um tamanho diferente de embalagem, numa determinada quantidade e portanto rendendo mais ou menos na hora de cozinhar....ah que bom seria se fosse tudo padronizado.

A proposta é criar uma função que realize esse cálculo: do valor gasto na produção do produto e o valor que este deve ser vendido para se obter determinado lucro, assim como o quanto deve ser produzido até que se obtenha lucro. Esta função será restrita para receitas de doces com ingredientes mais genéricos (como: bolos, tortas, brigadeiros, mousses) por conta da conversão peso/volume específica para cada ingrediente.

Input:

`receita(data.frame(receita), lucro, prod, n)`

- Dataframe com 6 colunas: 1. Ingredientes, 2. Receita (quanto de cada ingrediente), 3. Quantidade na embalagem, 4. Unidade de medida da quantidade (g, kg, ml, l), 5. Respectivos preços.
- lucro: quanto % a mais deseja se obter do valor inicial gasto
- prod: tamanho do produto final (em gramas) ou quantas unidades serão produzidas(?)
- n: quantas receitas serão feitas

Output:

- P: Preço da unidade

- plot: gráfico de lucro por vendas

Verificando parâmetros:

Arquivo de entrada (classe: data.frame) Dados nas colunas “quantidade na embalagem”, “receita” e “preços” (classe: numeric) Unidades de medida padronizadas (“g” ou “kg”, “ml” ou “l”)

Pseudo-código:

Entrar com uma tabela de conversão de massa para volume (ou vice-versa) para determinados ingredientes padrões (farinha, açúcar, chocolate em pó, fermento, leite condensado...).

Criar um objeto com a soma dos preços dos ingredientes - investimento inicial.

Criar um vetor composto pelos valores resultantes das diferenças entre as quantidades necessárias para uma receita e as quantidades das embalagens, respectivo para cada ingrediente.

- Os valores das diferenças irão entrar para as próximas receitas

Criar um vetor que contenha os valores relativos às quantidades usadas de cada ingrediente (basicamente regras de três). Criar um objeto com a soma dos valores acima - total gasto na produção de uma receita.

Criar um objeto com o valor da unidade (no caso se for mais de um, dividir o total gasto na produção da receita pela quantidade determinada na função) e somar a % de lucro desejada.

Criar um gráfico que simule o lucro após X vendas (incluindo investimentos necessários para próximas receitas)

Plano B. Correlação da composição de espécies entre áreas

As espécies se distribuem de maneira heterogênea no espaço devido as suas diferentes demandas e restrições biológicas. Portanto, não é incomum observamos em uma mesma região diferentes fragmentos (não muito distantes) com diferentes diversidades de animais, que podem variar não apenas em sua composição de espécies, como também nas respectivas abundâncias. Talvez um primeiro passo para se entender certos padrões de distribuições seja realmente identificar estes padrões, entendendo onde estão antes de se perguntar do por quê estão onde estão.

Para isto existe um cálculo simples de similaridade entre áreas, conhecido como Índice de Jaccard. A ideia desta função é realizar esse cálculo para analisar a correlação da composição de espécies entre áreas.

Input: jaccard(matrix())

Matriz de presença/ausência, com espécies nas colunas e áreas nas linhas

Output:

Cladograma com relação entre áreas a partir dos índices calculados por pares de áreas (como o

método de UPGMA)

Verificando parâmetros:

Matriz de entrada deve conter apenas 0s e 1s

Pseudo-código:

Criar uma matriz(A), a partir da primeira, que correlacione áreas com áreas pelo número de espécies em comum entre elas.

Criar uma matriz (B), a partir da primeira, que correlacione áreas com áreas pelas somas dos números de espécies de cada par.

Cálculo do índice de Jaccard:



Será criada então uma terceira matriz, resultante da razão entre as intersecções (matriz A) pelas uniões (matriz B) - índices entre áreas.

Construir um cladograma com os tamanhos dos ramos sendo o índice calculado para o par de áreas dividido por dois (como no método UPGMA)

Oi Carolina. Adorei a ideia da receita de bolo! Só tenho algumas sugestões de informações bônus que vc poderia adicionar ao objeto de saída (que provavelmente vai acabar sendo uma lista) se der tempo. (1) Uma lista de comprar especificando a quantidade de cada ingrediente que deve ser comparada para fazer n receitas. (2) Como os tamanhos de pacotes são todos não-padronizados, vc poderia devolver a quantidade desperdiçada tb. Mas são só sugestões.

Sobre o plano B. Ele parece um ótimo exercício, no sentido de que eu acho que há muito a se aprender implementando o algoritmo de UPGMA "na unha". Mas como já tem pacotes que fazem matrizes de similaridade/dissimilaridade e análises de agrupamento, a proposta perde um pouco da graça. Mas é uma proposta válida.

Danilo G Muniz

Código função trabalho final

```
bolar <- function(receita,n,l,r=NULL,v=NULL)
{
  if(class(receita)!="data.frame") ##se objeto de entrada não for data.frame
  {
    stop("Objeto não é data.frame")
  }
  if(class(l)!="numeric") ##argumento lucro deve ser numérico
  {
```

```
stop("Objeto não numérico")
}
if(class(n)!="numeric") ##número unidades deve ser numérica
{
  stop("Objeto não numérico")
}

##vetor com nomes dos ingredientes
ing <- receita$ingredientes
##investimento inicial
inv <- sum(receita$precos)
##quanto sobra após uma receita
dif <- receita$embalagem - receita$receita
names(dif)=ing ##nomes dos ingredientes no vetor
##valor de cada ingrediente relativo a quantidade usada na receita
rel <- (receita$receita*receita$precos)/receita$embalagem
names(rel)=ing ##nome dos ingredientes no vetor
##valor do produto final, que é a soma dos valores relativos
val <- sum(rel)
##valor final com lucro
val.fim <- val+(val*l)
##dividir valor final pela número de unidades a serem vendidas para então
##ter o valor por unidade
val.un <- val.fim/n ##argumento na função
##quanto sobra das embalagens se for feita mais uma receita
sobr = dif - bolo$receita ##valores negativos são de ingredientes que
faltam
names(sobr)=ing
##vetor com quantidades precos respectivos de cada ingrediente
precos <- bolo$precos
names(precos) <- ing
##vetor com quantidades embalagens
embal <- bolo$embalagem
names(embal)=ing
##rodar os comandos até que chegue em renda pré-determinado no argumento r
da função;
##para saber o número de vendas até que se atinja renda r
lucro = val.fim-inv
if(is.null(v))
{
  lucros <- c() ##vetor vazio para guardar resultados
while (lucro < r)
{
  print.table(lucro)
  ##data.frame com as variaveis:
  ##quanto sobra de cada ingrediente se for feita mais uma receita,
  ##quanto tem em cada embalagem e os respectivos preços
  bolo2 <- data.frame(sobr,dif,embal,precos)
  neg2 <- bolo2[which(sobr<0),] ##indexar linhas com valores negativos,
  ##que são ingredientes que precisam ser comprados
```

```

    inv2 <- sum(neg2$precos) ##soma dos preços de ingredientes com valores
negativos,
    ##que é o investimento necessário para uma próxima receita
    lucro = lucro+val.fim-inv2
    ##substituir valores negativos pela quantidade de uma nova embalagem
menos o que
    ##é utilizado em uma receita (vetor dif) - para fazer a nova tabela de
sobras
    meg3=which(bolo2[,1]<0)
    bolo2[meg3,]$sobr=bolo2[meg3,]$dif
    sobr = bolo2$sobr - bolo$receita
    lucros <- c(lucros, lucro)
}
vendas <- c(1:length(lucros)) ##vetor com vendas da receita
n.vendas <- length(vendas)
##fazer gráfico de renda em função das vendas
par(pch=16, las=1, tcl=0.3, tck=1)
plot(vendas,lucros,type="l", col="red",bty="l", lwd=2, ylim =
c(length(lucro),lucro[1]))
mtext("Rendimento por venda", side=3, cex=1.4)
abline(a=0, b=0, lwd=2.5)
return(c(val.un, n.vendas))
}
if(is.null(r))
{
  for(i in 1:v)
  {
    print(lucro)
    ##data.frame com as variaveis:
    ##quanto sobra de cada ingrediente se for feita mais uma receita,
    ##quanto tem em cada embalagem e os respectivos preços
    bolo2 <- data.frame(sobr,dif,embal,precos)
    neg2 <- bolo2[which(sobr<0),] ##indexar linhas com valores negativos,
    ##que são ingredientes que precisam ser comprados
    inv2 <- sum(neg2$precos) ##soma dos preços de ingredientes com valores
negativos,
    ##que é o investimento necessário para uma próxima receita
    lucro[i+1] = lucro[i]+val.fim-inv2
    ##substituir valores negativos pela quantidade de uma nova embalagem
menos o que
    ##é utilizado em uma receita (vetor dif) - para fazer a nova tabela de
sobras
    meg3=which(bolo2[,1]<0)
    bolo2[meg3,]$sobr=bolo2[meg3,]$dif
    sobr = bolo2$sobr - bolo$receita ##novo vetor sobr
  }
}
##a última posição do vetor de lucro será a renda final
renda <- lucro[v]
vendas <- c(1:length(lucro))

##fazer gráfico de renda em função das vendas

```

```
par(pch=16, las=1, tcl=0.3, tck=1)
plot(vendas,lucro,type="l", col="red",bty="l", lwd=2)
mtext("Rendimento por venda", side=3, cex=1.4)
abline(a=0, b=0, lwd=2.5)

return(c(val.un,renda))
}
}
```

Página de ajuda (help)

```
##Calcular preço de produto e renda

##Description:
#receita pode ser usada quando se pretende calcular o valor relativo à
produção de determinada receita

##Usage:
#receita(receita,n,l,r=NULL,v=NULL)

##Arguments:
#receita data.frame contendo as colunas: ingredientes, receita, embalagem e
preços
#n      número de unidades do produto final a partir de uma receita
#l      porcentagem sobre o valor do produto
#r      renda pretendida
#v      número de vendas

##Details:
#####Entrada com data.frame que contém uma coluna com os nomes dos
ingredientes, outra coluna com a quantidade
##de cada ingrediente utilizada na receita, outra com a quantidade em cada
embalagem e uma última coluna com
##respectivos preços dos ingredientes.
#####A função devolve o valor da unidade do produto a ser vendido assim como
simula determinado número de
##vendas e faz a relação com os lucros em um gráfico.

##Value:
#####A função retorna um gráfico de lucros em função das vendas que pode ser
feito a partir de duas entradas
##que funcionam separadamente: ou através do valor de renda pretendido(r),
ou através do número de vendas(v).

##Author(s):
#####Farhat, Carolina
##Examples:
#receita(data.frame,n=10,l=0.8)  ##preço do pedaço de bolo se for dividido
```

em

##10 e com 80% de lucro sobre o valor

```
#receita(data.frame,n=20,l=1.3,r=300) ##preço do pedaço se for dividido em  
20, com 130% de lucro e ##número de vendas necessárias para  
de obter renda de 300
```

```
#receita(data.frame,n=20,l=3,v=50) ##preço do pedaço se for dividido em  
20, com 300% de lucro e ##quanto de renda após 50 vendas
```

Código função

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:carolina.farhat:start 

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