

Érika M. Santana



Doutoranda em Ecologia, Instituto de Biociências, USP.

O título de minha tese é: “Consequências da terrestrialização da reprodução sobre a intensidade da seleção sexual em um clado de anuros neotropicais”, orientado pelo Prof. Dr. Glauco Machado.

Meus Exercícios

1.1 Basta Mudar o Diretório?; 1.2 Um Comando Perigoso; 1.3 Classes de Objetos:

[exerc1-3.r](#)

4.2 Cervejas; 4.3 Caixetais; 4.4 Eucaliptos; 4.5 Mais Caixetais:

[exerc4_2-5.r](#)

5.1 Editando alguns parâmetros gráficos; 5.2 Dois gráficos juntos; 5.3 Adivinhando o código:

[exerc5_1-3.r](#)

7.2 O modelo mais simples possível

[exerc7_2.r](#)

7.5 Seriemas e Carcarás

[107_5-seriemas_e_carcaras.r](#)

Exercícios de Regressão Múltipla

[exerc7b_regressao_multipla.r](#)

Exercícios Reamostragem e Simulação

[exercx-exercicios_reamostragem_e_simulacao.r](#)

9.2 Exercício Análise exploratória simultânea de duas variáveis

[exec9-construcao_de_funcoes.r](#)

Proposta de Trabalho Final

PROPOSTA: Função `contagem()`

A função “contagem” terá como objetivo resumir os dados de planilhas textuais. Ela foi pensada com

o intuito de organizar e contabilizar dados de coleções biológicas, mas poderá ser usada em qualquer data.frame cujos dados sejam caracteres. O intuito é contabilizar quantas ocorrências de dados no data.frame (ou linhas) há para cada índice especificado (por exemplo: número de exemplares de determinada espécie por localidade, ou por sexo, ou ambos). A função retornará um data.frame com as contagens para cada índice. O argumento 1 da função será o data.frame com os dados. No argumento 2, o usuário definirá qual o(s) índice(s) que deseja usar para fazer a contagem, sendo o índice o número da posição de uma coluna no data.frame. O terceiro argumento será TRUE quando o usuário quiser que todas as colunas até a coluna definida no argumento 2 sejam índices e FALSE para que só a coluna com o número escrito seja índice, sendo que o padrão da função será FALSE. O quarto argumento, quando TRUE, possibilitará ao usuário exportar o data.frame de contagem gerado pela função para um arquivo fora do R, em formato CSV. Nesse caso, o arquivo terá o nome do data.frame de entrada somado à palavra “contagem”. O padrão do quarto argumento será FALSE. A função retorna o dataframe.contagem, mostrando-o no console aos poucos, de forma que o usuário tenha que clicar para continuar a percorrer o dataframe.contagem.

Exemplo teórico do funcionamento da função:

```
contagem(data.frame, indice, = FALSE, salva.arquivo=FALSE){
```

- salva os dados em um objeto

- organiza os dados de acordo com os índices

- conta as ocorrências de dados

- salva a contagem em um dataframe

```
return(dataframe.contagem)
```

```
}
```

Achei a proposta A interessante! Mas talvez fosse uma boa implementá-la um pouco mais, caso você consiga pensar em mais alguma coisa que seja útil acrescentar. Talvez retornar alguns outros dados que resumem as informações da tabela, além da contagem.. Fiquei com uma dúvida: não entendi muito bem porque você criaria um outro argumento que possibilite a escolha de todas as colunas como índices (o terceiro argumento). Isso não poderia ser informado apenas com o segundo argumento?

—[Luísia Novara](#)

PLANO B: Função `indice.organo()`

A função “`indice.organo`” terá como objetivo comparar as massas relativas de órgãos de animais de

duas ou mais populações, ou seja, comparar os índices organossomáticos desse órgão. Para tanto, o usuário deverá entrar com três vetores concatenados ou um dataframe contendo os dados grupo amostral dos indivíduos, massa do órgão e massa total. A função fará a razão da massa do órgão (vetor 2 ou coluna 2 do dataframe, por padrão) pela massa do corpo (vetor 3 ou coluna 3 do dataframe, por padrão) e a multiplicará por 100, chegando assim ao índice organossomático. No argumento 2, o usuário poderá decidir o que fazer com NA, sendo TRUE a opção na qual os dados NA serão omitidos. O padrão da função será TRUE. No argumento 3, a função permitirá ao usuário escolher a análise usada para comparar os dados entre os grupos. As opções serão: - teste t: para dois grupos; a função retorna então o resultado da função t.test e um gráfico boxplot com os valores da amostra. Se a amostra tiver mais de 2 fatores a função retorna um aviso de erro. - anova: 3 ou mais grupos; nesse caso, a função faz a análise de variância (com a função lm) e em seguida o teste de Teste de Tukey a posteriori, retornando os resultados da tabela de anova e da função TukeyHSD, além de um boxplot com os dados amostrais. Se a amostra tiver menos de 3 fatores, a função retorna um aviso de erro.

Exemplo teórico do funcionamento da função:

```
indice.organo(dataframe, omitir.NA=TRUE, analise="anova"){
```

```
- salva os dados em um objeto, omitindo NAs.
```

```
- calcula o índice organossomático
```

```
- faz a anova entre os grupos
```

```
return(resultado da anova e do Teste de Tukey e boxplot com os valores dos dados)
```

```
}
```

Gostei mais da proposta A!

—[Luísa Novara](#)

Função contagem()

Código da função

```
contagem =function(data, index, save.file=FALSE) #Cria a função "contagem"
com os argumentos "data", "index" e "save.file", cujo padrão será "TRUE"
{
  if(!is.vector(index) ==TRUE) #Cria um condicional para caso o argumento
"index" não seja um vetor.
  {
    stop("'index' deve ser um vetor numérico") #Para a função contagem() e
retorna o aviso de erro contido entre as aspas.
  }
  if(!class(index)=="numeric") #Cria um condicional para caso o argumento
```

```
"index" não seja da classe "numeric"
{
  stop("'index' deve ser um vetor numérico") #Para a função contagem() e
retorna o aviso de erro contido entre as aspas.
}
if(!is.data.frame(data) ==TRUE) #Cria um condicional para caso argumento
"data" não seja um data.frame
{
  stop("'data' deve ser um data.frame") #Para a função contagem() e
retorna o aviso de erro contido entre as aspas.
}
if(length(index)>ncol(data)) #Cria um condicional para caso o número de
elementos do vetor "index" seja maior que o número de colunas do data.frame
"data".
{
  stop("Não há tantos índices no data.frame.") #Para a função contagem()
e retorna o aviso de erro contido entre as aspas.
}
combinacoes= (unique(data[,index])) #Cria e atribui ao objeto
"combinacoes" um data.frame com os tipos de linhas do data.frame "data",
removendo as linhas duplicadas. Assim, o data.frame "combinacoes" conterá
as ocorrências possíveis das combinacoes dos fatores presentes nas colunas
do data.frame "data" encontradas das posições indicadas pelos números do
vetor "index".
logica = matrix(nrow=nrow(data), ncol=length(index)) #Cria e atribui ao
objeto "logica" uma matriz lógica com o mesmo número de linhas do data.frame
"data" e o mesmo número de colunas que comprimento do argumento "index".
cont = 0 #Cria o objeto "cont" e atribui um 0 a ele.
for (i in 1:nrow(combinacoes)) #Percorre da primeira a última posição de
valores que começam em 1 e terminam no valor equivalente ao número de linhas
do data.frame "combinacoes".
{
  for(j in 1:length(index)) #Percorre da primeira a última posição de
valores que começam em 1 e terminam no valor equivalente ao número de
elementos do vetor "index".
  {
    logica[,j] = data[,index[j]] == combinacoes[i, j] #Compara todas
linhas da coluna do data.frame "data" presente na posição "j" do vetor
"index" com a célula presente na linha "i", coluna "j", do data.frame
"combinacoes". Atribui os resultados desses testes lógicos às linhas da
coluna "j" da matriz "logica".
    teste.linhas = apply(logica, 1, all) #Aplica a função all() às linhas
da matriz "logica". Atribui ao objeto "teste.linhas" um vetor com o mesmo
número de linhas que "logica", contendo "TRUE" nas linhas em que a linha
toda "logica" seja "TRUE". Caso haja pelo menos um "FALSE" na linha de
"logica", um "FALSE" é salvo no vetor, na posição dessa linha.
  }
}
### O controle de fluxo acima cria um vetor lógico no qual a quantidade de
"TRUE" equivale à quantidade de vezes em que os tipos de combinações
possíveis entre os fatores dos índices (colunas) do data.frame "data"
```

aparecem em data.

```
### Assim, é possível contar ocorrência de cara combinação de fatores dos índices escolhidos no data.frame original contando os "TRUE" presentes em "teste.linhas" (passo seguinte).
```

```
cont[i] = sum(teste.linhas) #Atribui a primeira posição do vetor "cont" a soma as linhas cujo valor é "TRUE" do objeto "teste.linha".
```

```
}
```

```
combinacoes$contagem = cont #Cria a coluna "contagem" no data.frame "combinacoes" e atribui os valores de "cont" a essa coluna
```

```
### A linha acima cria um vetor com o número de ocorrências de cada combinação listada no data.frame "combinacoes".
```

```
combinacoes = combinacoes[with(combinacoes, order(-contagem)), ] #Ordena o data.frame "combinacoes" pela ordem decrescente dos valores contidos na coluna "contagem" e atribui o data.frame ordenado ao objeto "combinacoes".
```

```
if(save.file==TRUE) #Cria um condicional para caso argumento "save.fale" seja "TRUE"
```

```
{
```

```
write.table(combinacoes, file= "Contagem.csv", sep = ";", col.names=TRUE, row.names=FALSE) #Cria um arquivo de extensão .csv no diretório do computador onde o R está operando contendo o conteúdo do data.frame "combinacoes".
```

```
}
```

```
return(combinacoes) #Retorna o data.frame "combinacoes" no console
```

```
}
```

Ajuda da função

contagem

package:unknown

R Documentation

Contagem de ocorrências de índices

Descrição:

Cria um data.frame com os valores totais de ocorrência de fatores (índices) de um data.frame de dados textuais.

Uso:

```
contagem(data, index, save.file=FALSE)
```

Argumentos:

data um data.frame de fatores.

index um vetor contendo os índices que serão contabilizados pela função. Os valores de index são as posições das colunas em data.

save.file lógico. Se "TRUE", salva o data.frame gerado pela função no diretório de trabalho.

Detalhes:

"Contagem" é útil para resumir os dados de planilhas textuais. Ela foi pensada com o intuito de organizar e contabilizar dados de coleções biológicas, mas poderá ser usada em qualquer data.frame.

Se os argumentos data e index não forem das classes requeridas pela função, a função para e retorna um aviso de erro.

Os dados de data serão tratados como fatores e sua frequência será contabilizada de acordo com os índices (colunas) apontados no argumento index.

Os valores de index equivalem às colunas de data cujas ocorrências combinadas serão contabilizadas. O número de elementos de index não pode ultrapassar o número de colunas de data. Se isso acontecer, a função para e retorna um aviso de erro.

Os espaços vazios em data serão considerados um dos fatores da coluna a qual pertencem, combinados e contados.

As ocorrências cujas frequência for 0 (zero) não serão listadas no data.frame resultante.

Valores:

Com contagem o usuário tem a opção de retornar um arquivo "Contagem.csv" com resultados das contagens dos índices em ordem decrescente.

Avisos:

Caso o número de elementos em index seja muito grande, a função poderá levar alguns segundos para rodar.

Autor(s):

Érika Marques de Santana (erika.ms@gmail.com)

Exemplos:

```
idades = c("Santos","São Paulo","Boracéia","Cubatão")
coletores = c("Ana","Paulo","João", "Sofia")
sp = sample(LETTERS, 10, replace=TRUE)
tabela = data.frame(especie=sample(sp, 100, replace=TRUE),
localidade=sample(idades, 100, replace=TRUE), coletor=sample(coletores,
100, replace=TRUE))
contagem(tabela, index=c(2,3), save.file=TRUE)
contagem(tabela, index=c(2,3), save.file=FALSE)
contagem(tabela, index=c(1,2))
```

Arquivos

[Função contagem\(\)](#)

[Ajuda - Função contagem\(\)](#)

From:

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

Permanent link:

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

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