

Camila Cassano



Doutoranda do curso de Ecologia, IB/USP

Exercícios

[exercício 1](#)

[exercício 2](#)

[exercício 3](#)

dados para exercício 3: [registros60min](#)

[exercício 4](#)

[exercício 5](#)

[exercício 6](#)

[exercício 7](#)

[exercício 8](#)

[exercício 9](#)

Trabalho final

[Planos A e B](#)

Função

```
sanfona.table = function (tab1, tab2, match.col=c("match.col1",  
"match.col2"), add.col= "add.col1", "add.col2", "add.col3", ...),  
ord.col1=F)  
{  
  dim11 = dim(tab1)[1]  
  dim12 = dim(tab1)[2]  
  dim21 = dim(tab2)[1]  
  nvar = length(add.col)  
  novacol= matrix(NA, nrow=dim11, ncol=nvar)  
  novalin = rep(0, dim21)  
  z=0  
  a = which(colnames(tab1)==match.col[1])
```

```
b = which(colnames(tab1)==match.col[2])
d = which(colnames(tab2)==match.col[1])
e = which(colnames(tab2)==match.col[2])
for (i in 1:dim11)
{
  posi = which(tab2[,d]==tab1[i,a] & tab2[,e]==tab1[i,b])
  novalin [posi] = 1
}
novalin
count = dim21 - sum(novalin)
novamat = matrix(NA, nrow= count, ncol= (dim12+nvar))
for (i in 1:dim21)
{
  if(novalin[i]==1)
  {
    for (j in 1:nvar)
    {
      novacol[tab2[i,d]==tab1[,a] & tab2[i,e]==tab1[,b],j] =
tab2[i,add.col[j]]
    }
  }
  else
  {
    z=z+1
    for(k in 1:dim12)
    {
      if(sum(colnames(tab1[k])==colnames(tab2))==1)
      {
        novamat[z,k]=tab2[i,which(colnames(tab2)==colnames(tab1[k]))]
      }
    }
    for (l in 1:nvar)
    {
      novamat[z,sum(dim12,l)] = tab2[i,add.col[l]]
    }
  }
}
novamat
novatab = data.frame(tab1, novacol)
colnames (novatab) = c(colnames(tab1), add.col)
for(m in 1:dim(novamat)[1])
{
  novatab[dim11+m, ] = novamat[m, ]
}
novatab
if(ord.col1!=F)
{
  novatab = novatab[order(novatab[,ord.col1]), ]
}
return(novatab)
```

```
}
```

Help

sanfona.table {}
Documentation

SANFONA TABLE

R

Description:

Unir duas tabelas em função de linhas comuns e colunas de mesmo nome

Usage:

```
sanfona.table = function (tab1, tab2, match.col=c("match.col1",  
"match.col2"), add.col= c("add.col1", "add.col2", "add.col3", ...),  
ord.col1=F)
```

Arguments:

tab1 tabela onde serão inseridas novas linhas e colunas

tab2 tabela da qual serão copiadas linhas e colunas para adição em
tab1

match.col vetor com os nomes de duas colunas de referência. Essas colunas
devem ter nomes iguais em tab1 e tab2.

 A partir dessas colunas serão feitas as relações para decisão
de incluir ou não incluir nova linha em
 tab1 (ver detalhes abaixo).

add.col vetor com os nomes das colunas de tab2 que serão adicionadas à
tab1.

ord.col1 o nome de uma uma coluna para ordenação dos dados na tabela
final.

Details:

São necessárias duas colunas de referência (mach.col) para que a função
sanfona.table seja executada.

O número de colunas à serem adicionadas em tab1 (match.col) é ilimitado,
porém é necessário ao menos uma coluna.

* 1. Se a combinação das colunas de referência na linha "i" de tab2 tiver
correspondente em tab1, nenhuma linha será adicionada em tab1, mas as novas
colunas serão preenchidas com os valores correspondentes em tab2.

* 2. Se a combinação das colunas de referência na linha "i" de tab2 não
tiver correspondente em tab1, uma nova linha será adicionada em tab1 com as

informações das colunas de mesmo nome em `tab1` e `tab2` e das novas colunas retiradas de `tab2`.

A informação de coluna para ordenação dos dados na tabela final (`ordcoll`) é opcional, por default `ord.coll = FALSE`

Value:

A função `sanfona.table` retorna um `data.frame` com número de colunas igual ao número de colunas de `tab1` mais o número de colunas adicionadas e número de linhas igual ao número de linhas de `tab1` mais o número de linhas de `tab2` para as quais não existe linha correspondente em `tab1` (situação 2 em detalhes).

Warning:

A função não será executada se: número de colunas de referência for diferente de 2 e não houver ao menos uma coluna de `tab2` para ser adicionada em `tab1`

Author(s):

Camila Righetto Cassano

See Also:

`merge`

Examples:

```
# usando uma coluna "sitio" e uma coluna "data" para unir bancos de dados
```

```
data = as.character(c("20/04/2010", "23/04/2010", "25/04/2010",  
"25/04/2010", "26/04/2010", "26/04/2010", "28/04/2010", "28/04/2010",  
"30/04/2010", "28/04/2010"))  
especie = as.character(c("macaco", "macaco", "macaco", "tatu", "tatu",  
"macaco", "tatu", "macaco", "macaco", "macaco"))  
sitio = as.character(c(rep("A",4), rep("B", 3), "A", "B", "A"))  
ambiente = as.character(c(rep("mata",4), "agricultura", "mata",  
"agricultura", rep("mata",3)))  
registros = data.frame( data, especie, sitio, ambiente, stringsAsFactors =  
FALSE)
```

```
data = as.character(c("20/04/2010", "21/04/2010", "22/04/2010",  
"23/04/2010", "24/04/2010", "25/04/2010", "26/04/2010", "27/04/2010",  
"28/04/2010", "29/04/2010", "30/04/2010"))
```

```
sitio = as.character(rep(c("A", "B"), each = 11))
observador = as.character(rep(c("observador1" , "observador2" ), each = 11))
amostragem = data.frame (data, sitio, observador, stringsAsFactors = FALSE)

sanfona.table (registros, amostragem, match.col=c("data", "sitio"), add.col=
c("observador"), ord.coll="sitio")
```

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2010:alunos:trabalho_final:camilacassano:start 

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