

Família apply

(apply, lapply, sapply, mapply, tapply)



Carlos Roberto de Nazaré Carvalho Junior

Fernanco José Pessoa Andrade

Ivanildo Batista da Silva Júnior

Vanessa Karoline Inacio Gomes

Vaniele da Silva Barros

31 de Maio de 2021

Universidade Federal Rural de Pernambuco

Programa de Pós-Graduação em Biometria e Estatística Aplicada (PPGBEA)

Uso de Software na Análise de Dados Biométricos

Professor : Antonio Samuel Alves da Silva

1. O que é a família apply
2. Função apply
3. Função lapply
4. Função sapply
5. Função mapply
6. Função tapply
7. Outras funções
8. Conclusão

O que é a família apply

Família apply

A família de funções *apply* é considerada como funções de *loop* que funcionam em tarefas repetidas e reduz a redundância que aparece devido ao *loop*. Essas funções estão embutidas no R e você há a necessidade de instalação separadamente. Elas permitem manipular *frames* de dados, *arrays*, matrizes, vetores; sendo alternativas aos *loops*, porém são mais eficientes pela rapidez no nível de execução.

As principais funções são

- `apply()`
- `lapply()`
- `sapply()`
- `mapply()`
- `tapply()`

Função apply

Função *apply*

A função *apply* permite aplicar uma função em linhas ou colunas (*MARGINS*) de uma matriz ou em *dataframes*. Abaixo podemos ver a sintaxe da função:

```
apply(X, MARGIN, FUN, ...)
```

- *X* é um *array*, incluindo matrizes;
- *MARGIN* é um vetor que define em qual parte do *array* será aplicada a função;
- *FUN* é a função que será aplicada;
- As **reticências** são argumentos da função, que podem ser inseridos opcionalmente.

A função retorna um vetor ou matriz ou lista de valores obtidos pela aplicação de uma função às margens de um *array* ou matriz.

Exemplo I - função apply

Criando uma matriz

```
vetor1 <- 1:25  
matriz = matrix(vetor1, 5,5)  
matriz
```

```
##      [,1] [,2] [,3] [,4] [,5]  
## [1,]    1    6   11   16   21  
## [2,]    2    7   12   17   22  
## [3,]    3    8   13   18   23  
## [4,]    4    9   14   19   24  
## [5,]    5   10   15   20   25
```

Somando as linhas da matriz

```
apply(matriz, 1, sum)
```

```
## [1] 55 60 65 70 75
```

Somando as colunas da matriz:

```
apply(matriz, 2, sum)
```

```
## [1] 15 40 65 90 115
```

Exemplo II - função apply

Média das colunas do *dataset mtcars*.

```
apply(mtcars, 2, mean)
```

##	mpg	cyl	disp	hp	drat	wt	qsec
##	20.090625	6.187500	230.721875	146.687500	3.596563	3.217250	17.848750
##	vs	am	gear	carb			
##	0.437500	0.406250	3.687500	2.812500			

Aplicando a função de desvio padrão para as colunas *Sepal.Length* e *Petal.Length* do *dataset iris*.

```
apply(iris[c('Sepal.Length', 'Petal.Length')], 2, sd)
```

##	Sepal.Length	Petal.Length
##	0.8280661	1.7652982

Exemplo III - função apply

Para calcular os quantis é possível inserir argumentos para a função *quantile*.

```
vetor_quantiles=c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9)
apply(matriz,1,quantile,probs = vetor_quantiles)
```

##	[,1]	[,2]	[,3]	[,4]	[,5]
## 10%	3	4	5	6	7
## 20%	5	6	7	8	9
## 30%	7	8	9	10	11
## 40%	9	10	11	12	13
## 50%	11	12	13	14	15
## 60%	13	14	15	16	17
## 70%	15	16	17	18	19
## 80%	17	18	19	20	21
## 90%	19	20	21	22	23

Função lapply

Função lapply

O *lapply* recebe uma lista como argumento e aplica uma função a cada elemento da lista por *loop*. Possui sintaxe conforme abaixo:

```
lapply(X, FUN, ...)
```

- *X* é uma lista;
- *FUN* é a função que será aplicada;
- As **reticências** são argumentos da função, que podem ser inseridos opcionalmente.

Diferente da função *apply*, não há a necessidade de especificar a *MARGIN*. Essa função retorna uma lista.

Exemplo I -função lapply

Criando uma lista com três elementos:

```
listal = list(a = 1:30, b=c(23,56,89,96,36,25,74), c=rnorm(10))  
listal
```

```
## $a  
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
## [26] 26 27 28 29 30  
##  
## $b  
## [1] 23 56 89 96 36 25 74  
##  
## $c  
## [1] -0.2813623 0.6198714 -0.1548512 -1.0946567 -0.9147516 -0.1356464  
## [7] 0.3793756 -0.5464622 0.1270922 1.3742296
```

Exemplo II - função lapply

Somatório

```
lapply(listal, sum)
```

```
## $a  
## [1] 465  
##  
## $b  
## [1] 399  
##  
## $c  
## [1] -0.6271615
```

Média

```
lapply(listal, mean)
```

```
## $a  
## [1] 15.5  
##  
## $b  
## [1] 57  
##  
## $c  
## [1] -0.06271615
```

Variância

```
lapply(listal, var)
```

```
## $a  
## [1] 77.5  
##  
## $b  
## [1] 909.3333  
##  
## $c  
## [1] 0.5387446
```

Desvio padrão

```
lapply(listal, sd)
```

```
## $a  
## [1] 8.803408  
##  
## $b  
## [1] 30.15515  
##  
## $c  
## [1] 0.7339922
```

Função supply

Função `sapply`

Essa função recebe uma lista, vetor ou *dataframe* como entrada e dá uma saída em vetor ou matriz. É útil para operações em objetos de lista e retorna um objeto de lista do mesmo comprimento do conjunto original. Faz o mesmo que a *lapply()*, mas retorna um vetor. Abaixo pode-se ver a sua sintaxe:

```
sapply(X, FUN, ..., simplify = TRUE, USE.NAMES = TRUE)
```

- *X* é uma lista;
- *FUN* é a função que será aplicada;
- As **reticências** são argumentos da função, que podem ser inseridos opcionalmente.
- *simplify* é um argumento lógico. argumento que especifica se queremos simplificar os resultados ou não.
- *USE.NAMES* usado para acessar valores na saída.

O objetivo dessa função é a simplificação da saída.

Exemplo I - função sapply

Simplificação: A saída retorna os valores um do lado do outro, ocupando menos espaço.

```
sapply(listal, sum, simplify = TRUE)
```

```
##           a           b           c  
## 465.000000 399.000000  1.112708
```

Sem simplificação: A saída retorna os valores igual na função *lapply()*

```
sapply(listal, sum, simplify = FALSE)
```

```
## $a  
## [1] 465  
##  
## $b  
## [1] 399  
##  
## $c  
## [1] 1.112708
```


Exemplo II - função `sapply`

Aplicando nas colunas *mpg* e *hp* do dataset *mtcars*.

```
sapply(mtcars[c('mpg','hp')], mean, simplify = TRUE)
```

```
##      mpg      hp  
## 20.09062 146.68750
```

```
sapply(mtcars[c('mpg','hp')], mean, simplify = FALSE)
```

```
## $mpg  
## [1] 20.09062  
##  
## $hp  
## [1] 146.6875
```

Função map

Função mapply

O *mapply()* é uma versão multivariada da função *sapply()* em R. O *mapply()* aplica uma função em paralelo ao conjunto de argumentos fornecido. Ele aplica a mesma função a cada argumento passado.

Quando existirem estruturas de dados diferentes (exemplo: vetores, listas) e haver a necessidade aplicar a função para os primeiros elementos de cada, depois para os segundos e assim por diante, forçando o resultado em um vetor ou *array* como em *sapply*.

Função mapply

A sintaxe para a função *mapply()* é mostrada abaixo:

```
mapply(FUN, ..., MoreArgs = NULL, SIMPLIFY = TRUE,  
       USE.NAMES = TRUE)
```

- *FUN* é uma função a ser aplicada;
- As *reticências* contêm objetos *R* para aplicar na função;
- *MoreArgs* é uma lista de outros argumentos para a função;
- *simplify* indica se o resultado deve ser simplificado;

Exemplo I - função mapply

Produto entre os valores das sequências.

```
mapply(prod, 1:2, 1:2, 1:2)
```

```
## [1] 1 8
```

Somatório entre os valores das sequências.

```
mapply(sum, 1:3, 10:12, 2:4, 7:9)
```

```
## [1] 20 24 28
```

rep dos valores de uma sequência com base em outra sequência.

```
mapply(rep, 5:8, 4:1)
```

```
## [[1]]  
## [1] 5 5 5 5  
##  
## [[2]]  
## [1] 6 6 6  
##  
## [[3]]  
## [1] 7 7  
##  
## [[4]]  
## [1] 8
```

Exemplo II - função mapply

Abaixo está uma função onde os valores do primeiro vetor são elevados ao quadrado e somados aos valores do segundo vetor, conforme suas respectivas ordens.

```
mapply(function(x, y) x^2 + y,  
       c(a = 11, b = 25, c = 43),  
       c(A = 10, B = -5, C = 36))
```

```
##      a      b      c  
##  131   620  1885
```

Função tap

Função `tapply`

É usada para aplicar uma função em subconjuntos de um vetor. É usada principalmente quando um conjunto de dados que pode ser dividido em grupos (por meio de fatores) e queremos dividir o conjunto de dados em grupos; e dentro de cada grupo, queremos aplicar uma função. A sintaxe dessa função é

```
tapply(X, INDEX, FUN = NULL, ..., default = NA, simplify = TRUE)
```

- *X* é o vetor em que a função será aplicada;
- *INDEX* é o vetor de fatores;
- *FUN* é a função que será aplicada no subgrupo;
- *simplify* é o parâmetro de simplificação (*TRUE* ou *FALSE*).

Exemplo I - função `tapply`

A variável `x` possui uma sequência que vai de 1 a 40. A variável `y` uma sequência de letras que vai de `a` até `h`, repetindo cada uma 5 vezes e transformadas em fatores, estando cada letra em ordem conforme a sequência de valores de `x`.

Aplicando a função `tapply()` a saída retornará a média para cada uma das letras (fatores).

```
x = 1:40
y = factor(rep(letters[1:8], each = 5))

tapply(x,y,mean)
```

```
##  a  b  c  d  e  f  g  h
##  3  8 13 18 23 28 33 38
```

Exemplo II - função `tapply`

Extraindo estatísticas da variável *mpg* por número de cilindros (*cyl*).

Média

Mediana

```
tapply(mtcars$mpg, factor(mtcars$cyl), mean)
```

```
##           4           6           8  
## 26.66364 19.74286 15.10000
```

```
tapply(mtcars$mpg, factor(mtcars$cyl), median)
```

```
##           4           6           8  
## 26.0 19.7 15.2
```

Variância

Desvio padrão

```
tapply(mtcars$mpg, factor(mtcars$cyl), var)
```

```
##           4           6           8  
## 20.338545  2.112857  6.553846
```

```
tapply(mtcars$mpg, factor(mtcars$cyl), sd)
```

```
##           4           6           8  
## 4.509828 1.453567 2.560048
```

Outras funções

- ***vapply()*** : quando querer utilizar o *sapply()*, mas caso precise de um código mais rápido.
- ***rapply()*** : aplicação de uma função para cada elemento de uma lista aninhada de forma recursiva.
- ***eapply()*** : aplica uma função aos valores nomeados de um ambiente (*environment*) e retorna os resultados como uma lista

Conclusão

Função	Tipos de dados de entrada	Tipos de dados de saída
<i>apply()</i>	<i>Dataframes</i> , matrizes ou <i>array</i> (com margens)	Vetor, matriz, <i>array</i> ou lista
<i>lapply()</i>	Vetor, lista, variáveis em <i>dataframes</i> ou matrizes	Lista
<i>sapply()</i>	Vetor, lista, variáveis em <i>dataframes</i> ou matrizes	Matriz, vetor ou lista
<i>mapply()</i>	Vetor, lista, variáveis em <i>dataframes</i> ou matrizes	Matriz, vetor ou lista
<i>tapply()</i>	<i>Ragged array</i> (<i>array</i> irregular)	<i>Array</i>

Referências

Documentação das funções da família *apply* no *R documentation*:

[apply](#)

[lapply](#)

[sapply](#)

[mapply](#)

[tapply](#)

[eapply](#)

[rapply](#)

Obrigado !
