

# Família de funções Apply no R-Base

*Vinícius Riffel*

*June 22, 2019*

## Resumo

Aqui você encontrará material sobre a família de funções Apply no R. Na primeira parte é apresentado cada função com exemplo. Na segunda parte é apresentada aplicações das funções.

## Sumário

<b>1</b>	<b>Função apply</b>	<b>2</b>
1.1	Funcionamento . . . . .	2
1.2	Argumentos . . . . .	2
1.3	Output . . . . .	2
1.4	Exemplo . . . . .	2
<b>2</b>	<b>Função mapply</b>	<b>2</b>
2.1	Funcionamento . . . . .	2
2.2	Argumentos . . . . .	2
2.3	Output . . . . .	3
2.4	Exemplo . . . . .	3
<b>3</b>	<b>Função lapply</b>	<b>3</b>
3.1	Funcionamento . . . . .	3
3.2	Argumentos . . . . .	3
3.3	Output . . . . .	3
3.4	Exemplo . . . . .	3
<b>4</b>	<b>Função rapply</b>	<b>4</b>
4.1	Funcionamento . . . . .	4
4.2	Argumentos . . . . .	4
4.3	Output . . . . .	5
4.4	Exemplo . . . . .	5
<b>5</b>	<b>Função tapply</b>	<b>5</b>
5.1	Funcionamento . . . . .	5
5.2	Argumentos . . . . .	5
5.3	Output . . . . .	5
5.4	Exemplo . . . . .	5

# 1 Função apply

## 1.1 Funcionamento

Aplica uma função pré-determinada nas margens de uma array.<sup>1</sup>

## 1.2 Argumentos

- **X**: Array com duas dimensões
- **MARGIN**: Vetor indicando em que sentido a operação será aplicada em **X**. 1 significa por linha e 2 significa por coluna.
- **FUN**: Função a ser aplicada.
- **...**: Argumentos opcionais para **FUN**.

## 1.3 Output

Retorna um vetor ou lista com o valor da função aplicada em X.

## 1.4 Exemplo

```
data(mtcars)
apply(X = mtcars, MARGIN = 2, FUN = mean)
```

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

# 2 Função mapply

## 2.1 Funcionamento

Aplica uma função nos elementos de uma array.

## 2.2 Argumentos

- **FUN**: Função a ser aplicada.
- **...**: Elemento a ser vetorizado.

---

<sup>1</sup>Documentação disponível em:  
<https://www.rdocumentation.org/packages/base/versions/3.6.0/topics/apply>

Para verificar todos os argumentos de ***mapply*** acesse a documentação<sup>2</sup>

## 2.3 Output

Uma array com os valores de **FUN** aplicados a **X**

## 2.4 Exemplo

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

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    1    2    3    4
## [3,]    1    2    3    4
## [4,]    1    2    3    4
```

# 3 Função lapply

## 3.1 Funcionamento

Aplica uma função em um vetor (lista ou atomic vetor).

## 3.2 Argumentos

Os principais argumentos são:<sup>3</sup>

- **X**: Vetor ou atomic vetor.
- **FUN**: Função a ser aplicada em **X**
- **...**: Argumentos opcionais de **FUN**

## 3.3 Output

Retorna uma lista do mesmo tamanho de **X**, cada elemento da lista é a função aplicada em **X** na correspondente posição

## 3.4 Exemplo

```
lapply(X = mtcars, FUN = mean)
```

---

<sup>2</sup><https://www.rdocumentation.org/packages/base/versions/3.6.0/topics/mapply>

<sup>3</sup>Para verificar todos os argumentos acesse a documentação:

<https://www.rdocumentation.org/packages/base/versions/3.6.0/topics/lapply>

```
## $mpg
## [1] 20.09062
##
## $cyl
## [1] 6.1875
##
## $disp
## [1] 230.7219
##
## $hp
## [1] 146.6875
##
## $drat
## [1] 3.596563
##
## $wt
## [1] 3.21725
##
## $qsec
## [1] 17.84875
##
## $vs
## [1] 0.4375
##
## $am
## [1] 0.40625
##
## $gear
## [1] 3.6875
##
## $carb
## [1] 2.8125
```

## 4 Função rapply

### 4.1 Funcionamento

Aplica recursivamente uma função em uma array (vetor ou atomic vetor). Pode ser visto como uma versão recursiva da função *lapply*.

### 4.2 Argumentos

- **object**: Uma lista.

- **f**: A função que será aplicada.
- **...**: Argumentos opcionais de **f**.

Para verificar todos os argumentos de **rapply** acesse a documentação<sup>4</sup>

## 4.3 Output

Retorna uma lista do mesmo tamanho de **X**, cada elemento da lista é a função aplicada em **X** na correspondente posição

## 4.4 Exemplo

```
rapply(object = mtcars, f = mean)
```

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

# 5 Função tapply

## 5.1 Funcionamento

Aplica uma função em grupos de uma array.

## 5.2 Argumentos

- **X**: Um objeto que seja possível aplicar o método *split*.
- **INDEX**: Uma lista de fatores do mesmo tamanho de **X**. A lista será automaticamente convertida para um fator.
- **FUN**: Função a ser aplicada em **X**.
- **...**: Argumentos opcionais de **FUN**.

## 5.3 Output

Retorna um vetor com **FUN** aplicada em **X**.

## 5.4 Exemplo

---

<sup>4</sup><https://www.rdocumentation.org/packages/base/versions/3.6.0/topics/rapply>

```

n <- 17; fac <- factor(rep_len(1:3, n), levels = 1:5)
tapply(1:n, fac, sum)

## 1 2 3 4 5
## 51 57 45 NA NA

## Cálculo de probabilidades
## Uma variável aleatória segue uma distribuição binomial de parâmetros
## n = 250 e p = 0.17. Calcule o valor esperado de sucessos para os 50
## primeiros experimentos.
range <- seq(from = 0, to = 50, by = 1)
system.time(
  for (i in range) {
    Ex <- dbinom(x = i, size = 250, prob = 0.17) * 50
    Ex <- round(Ex, 2)
    print(paste0("Valor esperado de sucessos para X = ", i, " É ", Ex))
  }
)

## [1] "Valor esperado de sucessos para X = 0 É 0"
## [1] "Valor esperado de sucessos para X = 1 É 0"
## [1] "Valor esperado de sucessos para X = 2 É 0"
## [1] "Valor esperado de sucessos para X = 3 É 0"
## [1] "Valor esperado de sucessos para X = 4 É 0"
## [1] "Valor esperado de sucessos para X = 5 É 0"
## [1] "Valor esperado de sucessos para X = 6 É 0"
## [1] "Valor esperado de sucessos para X = 7 É 0"
## [1] "Valor esperado de sucessos para X = 8 É 0"
## [1] "Valor esperado de sucessos para X = 9 É 0"
## [1] "Valor esperado de sucessos para X = 10 É 0"
## [1] "Valor esperado de sucessos para X = 11 É 0"
## [1] "Valor esperado de sucessos para X = 12 É 0"
## [1] "Valor esperado de sucessos para X = 13 É 0"
## [1] "Valor esperado de sucessos para X = 14 É 0"
## [1] "Valor esperado de sucessos para X = 15 É 0"
## [1] "Valor esperado de sucessos para X = 16 É 0"
## [1] "Valor esperado de sucessos para X = 17 É 0"
## [1] "Valor esperado de sucessos para X = 18 É 0"
## [1] "Valor esperado de sucessos para X = 19 É 0"
## [1] "Valor esperado de sucessos para X = 20 É 0"
## [1] "Valor esperado de sucessos para X = 21 É 0"
## [1] "Valor esperado de sucessos para X = 22 É 0"
## [1] "Valor esperado de sucessos para X = 23 É 0.01"
## [1] "Valor esperado de sucessos para X = 24 É 0.02"

```

```
## [1] "Valor esperado de sucessos para X = 25 É 0.03"
## [1] "Valor esperado de sucessos para X = 26 É 0.05"
## [1] "Valor esperado de sucessos para X = 27 É 0.09"
## [1] "Valor esperado de sucessos para X = 28 É 0.15"
## [1] "Valor esperado de sucessos para X = 29 É 0.23"
## [1] "Valor esperado de sucessos para X = 30 É 0.34"
## [1] "Valor esperado de sucessos para X = 31 É 0.5"
## [1] "Valor esperado de sucessos para X = 32 É 0.7"
## [1] "Valor esperado de sucessos para X = 33 É 0.95"
## [1] "Valor esperado de sucessos para X = 34 É 1.24"
## [1] "Valor esperado de sucessos para X = 35 É 1.57"
## [1] "Valor esperado de sucessos para X = 36 É 1.92"
## [1] "Valor esperado de sucessos para X = 37 É 2.27"
## [1] "Valor esperado de sucessos para X = 38 É 2.61"
## [1] "Valor esperado de sucessos para X = 39 É 2.9"
## [1] "Valor esperado de sucessos para X = 40 É 3.14"
## [1] "Valor esperado de sucessos para X = 41 É 3.29"
## [1] "Valor esperado de sucessos para X = 42 É 3.36"
## [1] "Valor esperado de sucessos para X = 43 É 3.32"
## [1] "Valor esperado de sucessos para X = 44 É 3.2"
## [1] "Valor esperado de sucessos para X = 45 É 3"
## [1] "Valor esperado de sucessos para X = 46 É 2.74"
## [1] "Valor esperado de sucessos para X = 47 É 2.44"
## [1] "Valor esperado de sucessos para X = 48 É 2.11"
## [1] "Valor esperado de sucessos para X = 49 É 1.78"
## [1] "Valor esperado de sucessos para X = 50 É 1.47"
```

```
##      user  system elapsed
##    0.005    0.000    0.006
```

```
## Com apply
fx <- function(x) {
  Ex <- dbinom(x = x, size = 250, prob = 0.17) * 50
  return(round(Ex, 2))
}
system.time(
  lapply(X = 1:50, FUN = fx)
)
```

```
##      user  system elapsed
##    0.003    0.000    0.003
```