

Vetor e Fator

Osemar Xavier

1 de Setembro de 2018

Criacao de um vetor com a funcao C

```
vetor <-c(1,2,3,4,5,6,7)
vetor
```

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

Vetor de characters

```
nomes <-c("Maria", "Joao", "Luiz")
nomes
```

```
## [1] "Maria" "Joao"  "Luiz"
```

Tamanho do vetor

```
length(nomes)
```

```
## [1] 3
```

```
length(vetor)
```

```
## [1] 7
```

verificar se a colecao e um vetor

```
is.vector(vetor)
```

```
## [1] TRUE
```

vetor numerico

```
numeros <-1:10
numeros
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

funcao seq

```
numero2 <- seq(0,1, by= 0.1)
numero2
```

```
## [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

vetores -> de tipos diferentes

```
c("a", numeros)
```

```
## [1] "a" "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"
```

Operacoes matematicas com vetores em R

```
operacao <- c(10,20,30,40)
operacao + 3
```

```
## [1] 13 23 33 43
```

```
operacao - 3
```

```
## [1] 7 17 27 37
```

```
operacao * 3
```

```
## [1] 30 60 90 120
```

```
operacao / 3
```

```
## [1] 3.333333 6.666667 10.000000 13.333333
```

vetor de elementos repetidos

```
repetidos <- rep(1,5)
repetidos
```

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

```
repetidos2 <- rep(c(1,2), c(3,4))
repetidos2
```

```
## [1] 1 1 1 2 2 2 2
```

Funcao utilizada para criar vetores de caracteres .paste()

```
n<- c("lucas", "Bia", "Rui")
paste(n, 1:3)
```

```
## [1] "lucas 1" "Bia 2"   "Rui 3"
```

```
paste("Maria", 2)
```

```
## [1] "Maria 2"
```

```
rep(paste(n, 1:3, sep=""), c(4,4,3))
```

```
## [1] "lucas1" "lucas1" "lucas1" "lucas1" "Bia2"   "Bia2"   "Bia2"
## [8] "Bia2"   "Rui3"   "Rui3"   "Rui3"
```

```
pas <- paste(n, 1:3, sep="")
rep(pas, c(4,4,3))
```

```
## [1] "lucas1" "lucas1" "lucas1" "lucas1" "Bia2"   "Bia2"   "Bia2"
## [8] "Bia2"   "Rui3"   "Rui3"   "Rui3"
```

Definindo e diferenciando vetor de um factor

Exemplo de um vetor

Class -> Em vetores são identificados como 'Caracteres'

Class -> Em fatores são identificados como 'Levels'

Atributo -> Identifica se é um vetor ou um fator

```
estados <- c("RJ", "SP", "MG")
estados
```

```
## [1] "RJ" "SP" "MG"
```

```
class(estados)
```

```
## [1] "character"
```

```
attributes(estados)
```

```
## NULL
```

Exemplo de um fator

```
estadosf <- factor(estados)
estadosf
```

```
## [1] RJ SP MG
## Levels: MG RJ SP
```

```
class(estadosf)
```

```
## [1] "factor"
```

```
attributes(estadosf)
```

```
## $levels
## [1] "MG" "RJ" "SP"
##
## $class
## [1] "factor"
```

Table -> Usado para contar e trazer o quantitativo na forma de tabela

```
peessoas <-c("Rui", "Bia", "Ana", "Leo", "Nat", "Rui", "Leo", "Bia", "Leo")
peessoas
```

```
## [1] "Rui" "Bia" "Ana" "Leo" "Nat" "Rui" "Leo" "Bia" "Leo"
```

```
table(peessoas)
```

```
## pessoas
## Ana Bia Leo Nat Rui
##    1  2  3  1  2
```

O Table e o factor foi utilizado nesse exemplo para que faça a contagem do vetor acima, identificando os levels, como o 'Bob' não existe

o valor dele será zero.

```
peessoasn <-factor(peessoas, levels = c("Ana", "Rui", "Bia", "Nat", "Leo", "Bob"))
table(peessoasn)
```

```
## pessoasn
## Ana Rui Bia Nat Leo Bob
##    1  2  2  1  3  0
```

reagrupando fatores

```
perssoassite <- pessoasn  
levels(perssoassite)
```

```
## [1] "Ana" "Rui" "Bia" "Nat" "Leo" "Bob"
```

Criterio de grupo

```
levels(perssoassite) <- c("Site", "Site", "Site", "Site", "App", "App")  
table(perssoassite)
```

```
## perssoassite  
## Site App  
##      6   3
```

ordenacao de fatores

```
grau <- c("medio", "baixo", "medio", "baixo", "alto", "medio", "alto", "medio")  
table(grau)
```

```
## grau  
##  alto baixo medio  
##    2     2     4
```

```
factor(grau, ord=T)
```

```
## [1] medio baixo medio baixo alto  medio alto  medio  
## Levels: alto < baixo < medio
```

```
factor(grau, ord=T, levels = c("baixo","medio","alto"))
```

```
## [1] medio baixo medio baixo alto  medio alto  medio  
## Levels: baixo < medio < alto
```

Funcao ordered

```
ordered(grau, levels = c("baixo", "medio", "alto"))
```

```
## [1] medio baixo medio baixo alto  medio alto  medio  
## Levels: baixo < medio < alto
```

Ordem inversa

```
vec <- (1:10)  
vec
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
rev(vec)
```

```
## [1] 10 9 8 7 6 5 4 3 2 1
```

criacao dinamica

```
vec <- round(rnorm(7, m=70, sd=10))  
vec
```

```
## [1] 81 68 64 57 69 78 65
```

ordenar

```
sort(vec)
```

```
## [1] 57 64 65 68 69 78 81
```

ranqueamento

```
rank(vec)
```

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

which -> posicao do elemento que obedece a um determinado criterio

```
which(vec > 75)
```

```
## [1] 1 6
```

maior valor

```
which.max(vec)
```

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

menor valor

```
which.min(vec)
```

```
## [1] 4
```

remover elementos

```
vec[-5]
```

```
## [1] 81 68 64 57 78 65
```

```
vec
```

```
## [1] 81 68 64 57 69 78 65
```

remover um intervalo

```
vec[-(2:4)]
```

```
## [1] 81 69 78 65
```