

Lista de exercícios I - MAP2112

Introdução à Lógica de Programação e Modelagem Computacional

Prof. Luis Carlos de Castro Santos

Monitora Lucy Anne de Omena Evangelista

Universidade de São Paulo

1 de abril de 2020

Exercício 1. Considere dois vetores a e b . Sendo, $a=c(1,2,4,5,6)$ e $b=c(3,2,4,1,9)$. Qual é o valor de `cbind(a,b)`?

Exercício 2. Se $x=c('blue',10,'green',20)$, qual é o valor de `is.character(x)`? Qual será a saída da expressão em R `(c(FALSE, TRUE)) || (c(TRUE, TRUE))`?

Exercício 3. Considere o vetor $x <- c(4, 6, 5, 7, 10, 9, 4, 15)$. Qual é o valor de `c(4,6,5,7,10,9,4,15) < 7` ?

- a. TRUE, FALSE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE
- b. TRUE, TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE
- c. FALSE, TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE
- d. TRUE, TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, FALSE
- e. TRUE, TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE

Exercício 4. Considere o vetor $x <- c(1,2,3,4)$. Qual é o valor de k para `(x+2)[(!is.na(x)) & x > 0] -> k` ?

- a. 1, 2, 3, 4
- b. 1, 4, 9, 16
- c. Error: object 'k' not found
- d. 3, 4, 5, 6
- e. numeric(0)

Exercício 5. Se $X <-c(22,3,7,NA,NA,67)$, qual será a saída da declaração `length(X)` ?

Exercício 6. Se `X = c(34,33,65,37,89,NA,43,NA,11,NA,23,NA)`, então qual sentença abaixo retornará a conta do número de ocorrências NA em X?

- a. `sum(X==NA)`
- b. `sum(X == NA, is.na(X))`
- c. `sum(is.na(X))`

Exercício 7. Carregue o conjunto de dados ‘Orange’ do R usando o comando `data(Orange)`. Substitua todos os valores de `age = 118` para NA.

Exercício 8. Se `x <- c("ss", "aa", "ff", "kk", "bb")`, qual será o terceiro valor na operação de índice de vetor `x[c(2, 4, 4)]`?

- a. `"uu"`
- b. `NA`
- c. `"kk"`
- c. `"ss"`
- c. `"aa"`

Exercício 9. Considere o dataframe `s <- data.frame(first= as.factor(c("x", "y", "a", "b", "x", "z")), second=c(2, 4, 6, 8, 10, 12))`. Escreva uma sentença em R que retornará a saída 2, 4, 10, usando a variável `first` como um vetor de índices(index vector).

Exercício 10. Se `x = c(1, 2, 3, 3, 5, 3, 2, 4, NA)`, quais são os níveis de `factor(x)`?

- a. 1, 2, 3, 4, 5
- b. NA
- c. 1, 2, 3, 4, 5, NA
- c. 1, 2, 3, 3, 5, 3, 2, 4
- c. 1, 2, 2, 3, 3, 3, 4, 5

Exercício 11. Se `z <- c("p", "a", "g", "t", "b")`, qual das seguintes expressões em R vão substituir o terceiro elemento em `z` com `"b"`?

- a. `factor(z[3]) <- "b"`
- b. `levels(z[3]) <- "b"`

c. `z[3] <- "b"`

Exercício 12. Crie três vetores `x,y,z` com inteiros e cada vetor possuindo 3 elementos. Combine os três vetores para formar uma matriz `A`, 3x3, onde cada coluna representa um vetor. Mude os nomes das colunas para `a`, `b` e `c`. Como você pode mudar seu código para implementar o caso de cada linha representar um vetor? Isso é possível?. Verifique o resultado do exercício usando `is.matrix(A)`.

Exercício 13. Crie matrizes a partir dos vetores abaixo, por juntá-los a partir das colunas. Primeiro, sem programar, escreva quais junções dos vetores em matrizes são possíveis; para cada possibilidade, mostre a matrix resultante e seu modo (`character`, `numeric`, etc.). Por fim cheque sua resposta usando R. Repita o exercício fazendo a junção pela linha ao invés da coluna enquanto evita os nomes das linhas.

a. `a <- 1:5 ; b <- 1:5`

b. `a <- 1:5 ; b <- c('1', '2', '3', '4', '5')`

c. `a <- 1:5 ; b <- 1:4; c <- 1:3`

Exercício 14. Crie novos dataframes a partir dos dataframes abaixo, por juntá-las (`binding`) pela coluna. Primeiro escreva quais junções são possíveis, e a classe resultante do dataframe e a classe de cada coluna (`integer`, `character`, `factor` etc.). Depois verifique sua resposta usando R. Como adicional, use `cbind()` e `rbind()` nos dataframes abaixo para juntar as matrizes a partir das colunas, adicionando NA nas células vazias.

a. `a <- data.frame(v1=1:5, v2=LETTERS[1:5]) ;`
`b <- data.frame(var1 = 6:10, var2 = LETTERS[6:10])`

b. `a <- data.frame(v1 = 1:6, v2 = LETTERS[1:6]) ;`
`b <- data.frame(var1 = 6:10, var2 = LETTERS[6:10])`

Exercício 15. Se temos `p <- c(2,7,8)`, `q <- c("A", "B", "C")` e `x <- list(p, q)`, qual é o valor de `x[2]` ?

a. `NULL`

b. `"A" "B" "C"`

a. `"7"`

Exercício 16. Se `b <- list(a=1:10, c="Hello", d="AA")`, escreva uma expressão em R que retornará todos os elementos, exceto o segundo, do primeiro vetor de `b`.

Exercício 17. Seja `y <- list("a", "b", "c")` e `q <- list("A", "B", "C", "a", "b", "c")`, escreva uma expressão em R que retornará todos os elementos de `q` que não estão em `y` com o seguinte resultado:

```
[[1]]
[1] "A"
[[2]]
[1] "B"
[[3]]
[1] "C"
```

Exercício 18. Crie o dataframe abaixo, e em seguida inverta o sexo de todos os indivíduos.

	Idade	Altura	Peso	Sexo
Alex	25	177	57	F
Lilly	31	163	69	F
Mark	23	190	83	M
Oliver	52	179	75	M
Martha	76	163	70	F
Lucas	49	183	83	M
Caroline	26	164	53	F

Exercício 19. Use o dataset interno VADeaths para este exercício. Abra-o de forma semelhante ao exercício 7.

- Verifique se o objeto é um dataframe. Se não for, converta-o para dataframe.
- Crie uma nova variável, chamada Total, que deve receber a soma de cada linha.
- Mude a ordem das colunas de forma tal que Total seja a primeira variável.

Exercício 20. Leia o arquivo Table0.txt. Mude os nomes das colunas para Nome, Idade, Altura, Peso e Sexo e mude os nomes das linhas de tal forma que eles fiquem iguais aos da coluna Nome, e remova a variável Nome.

Exercício 21. Leia o arquivo Table1.txt, quantas linhas e colunas ele possui? Releia o arquivo e faça com que a variável Name nomeie os nomes das colunas. Tenha certeza de ler a variável como caracteres, não como fatores.

Exercício 22. Leia o arquivo Table6.txt, verificando-o primeiro. Perceba que há informações que estão repetidas, e queremos somente as primeiras não repetidas. Retire os comentários, e tenha certeza de criar somente caracteres e não fatores durante a conversão.