

Segunda Lista de Exercícios: Fundamentos de R

Exercício 1. Crie os seguintes vetores:

- (a) $(1, 2, 3, \dots, 19, 20)$;
- (b) $(20, 19, 18, \dots, 2, 1)$;
- (c) $(1, 2, 3, \dots, 19, 20, 19, 18, \dots, 2, 1)$;
- (d) $(0.1^3 0.2^1, 0.1^6 0.2^4, \dots, 0.1^{36} 0.2^{34})$;

Use a função `help` para descobrir o funcionamento da função `rep`. Em seguida, tente utilizá-la para responder os próximos itens.

- (e) $(4, 6, 3, 4, 6, 3, \dots, 4, 6, 3)$, em que há 10 ocorrências do número 4;
- (f) $(4, 6, 3, 4, 6, 3, \dots, 4, 6, 3, 4)$, em que há 11 ocorrências do número 4, 10 ocorrências do número 6 e 10 ocorrências do número 3;

Exercício 2. Crie um vetor cujas entradas são dadas pelo valor de $e^x \cos(x)$ nos pontos $x = 3, 3.1, 3.2, 3.3, \dots, 6$.

Exercício 3. Calcule:

- (a) $\sum_{i=10}^{100} (i^3 + 4i^2)$;
- (b) $\sum_{i=10}^{25} \left(\frac{2^i}{i} + \frac{3^i}{i^2} \right)$.

Exercício 4. A função `sample` é utilizada para realizar sorteios de um dado conjunto. Exemplo: `sample(x = 1:5, size = 10, replace = TRUE)` sorteará 10 (`size = 10`) números do conjunto/vetor $\{1, 2, 3, 4, 5\}$ (`x = 1:5`) com reposição (`replace = TRUE`). Caso se deseje realizar um sorteio sem reposição deve-se utilizar `replace = FALSE`.

Execute as seguintes linhas de comando:

```
xVec <- sample(0:999, 250, replace=T)
yVec <- sample(0:999, 250, replace=T)
```

Suponha que `xVec` seja representado por (x_1, x_2, \dots, x_n) e que o vetor `yVec` seja representado por (y_1, x_2, \dots, y_n) em que $n = 250$.

- (a) Crie um vetor com as entradas ímpares de `xVec`;
- (b) Crie o vetor $(y_2 - x_1, \dots, y_n - x_{n-1})$;
- (c) Crie o vetor $\left(\frac{\sin(y_2)}{\cos(x_1)}, \frac{\sin(y_3)}{\cos(x_2)}, \dots, \frac{\sin(y_n)}{\cos(x_{n-1})} \right)$;
- (d) Crie o vetor $(x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, \dots, x_{n-2} + 2x_{n-1} - x_n)$;

(e) Calcule $\sum_{i=1}^{n-1} \frac{e^{-x_{i+1}}}{x_i+10}$.

Exercício 5. Este exercício ainda utilizará os vetores `xVec` e `yVec` definidos no Exercício 4. Tente utilizar, a seguir, algumas das seguintes funções: `sort`, `order`, `mean`, `sqrt`, `sum`, `abs`, `max`, `which`.

- (a) Quais são os índices do vetor `yVec` que correspondem a valores maiores do que 600?
- (b) Selecione os valores de `yVec` que sejam maiores do que 600.
- (c) Quais valores do vetor `xVec` correspondem às entradas do vetor `yVec` que são maiores do que 600?
- (d) Crie o vetor $(|x_1 - \bar{x}|^{1/2}, |x_2 - \bar{x}|^{1/2}, \dots, |x_n - \bar{x}|^{1/2})$, em que \bar{x} representa a média do vetor `xVec`.
- (e) Quantos valores do vetor `yVec` estão a uma distância de até 200 do valor máximo deste vetor?
- (f) Quantos números de `xVec` são divisíveis por 2?
- (g) Ordene os valores de `xVec` na ordem que deixa os valores de `yVec` crescentes.
- (h) Selecione os valores de `yVec` que estão nas posições 1, 4, 7, 10,...

Exercício 6. Use um loop para escrever uma função chamada `num.impar` cuja entrada seja um vetor `v` e cuja saída retorne quantos números ímpares existem no vetor `v`. Em seguida, escreva uma função `num.impar2` que faça a mesma coisa sem usar um loop.

Exercício 7. Os dois primeiros termos da sequência de Fibonacci são iguais a 1. Os termos subsequentes da sequência são encontrados somando os dois termos imediatamente anteriores. Escreva uma função chamada `fibonacci` cuja entrada seja `n` e cuja saída retorne os primeiros `n` termos da sequência de Fibonacci para qualquer $n \geq 3$. Exemplo:

```
> fibonacci(6)
[1] 1 1 2 3 5 8
```

Exercício 8. Escreva uma função chamada `tamanho.seq` que simule o lançamento de uma moeda com probabilidade p de cara até que a primeira coroa seja vista. A função deve retornar a quantidade de caras que foram obtidas até a obtenção da primeira coroa. A entrada da função deverá ser a probabilidade p de sair cara. Por fim, replique a execução desta função 100 vezes com uma moeda honesta e encontre a maior sequência de caras obtidas nas 100 realizações do experimento.

Exemplo: supondo que coroa seja 0 e que cara seja 1, se a sequência obtida da realização de um experimento foi 1,1,1,1,0 então a função retornará o valor 4.

Exercício 9. Na clássica formulação da sequência de Fibonnaci um par de coelhos em idade reprodutiva pode originar apenas outro par novos coelhos. É possível generalizar o problema para que cada par de coelhos em idade reprodutiva possa originar k pares de novos coelhos. Para esse exercício, crie uma função cuja entrada seja n (número de meses) e k (número de novos pares por casal) e cuja saída seja a quantidade total de coelhos no mês n . As intruções sobre a formulação do problema podem ser lidas em: <http://rosalind.info/problems/fib/>

Exercício 10. Instale o pacote `dslabs` e em seguida leia o conjunto de dados `murders`:

```
install.packages(dslabs)
library(dslabs)
data(murders)
```

- (a) Calcule a média da variável `population`. Qual o estado mais populoso? E qual o estado menos populoso?
- (b) Crie uma nova coluna no data frame `murders` para representar a taxa por estado de mortes por arma de fogo por 10 mil habitantes.
- (c) Imprima o nome dos estados por ordem crescente de taxa de mortalidade.
- (d) Quais estados apresentam uma taxa de mortalidade inferior a 0.05? E superior a 0.5?
- (e) Qual estado possui a maior taxa de mortalidade e qual estado possui a menor taxa de mortalidade?
- (f) Divida o data frames por região. Em seguida, calcule a média e o desvio padrão da taxa de morte para cada região. Qual região possui a maior taxa de mortalidade? E a menor?
- (g) Apresente simultaneamente os boxplots da taxa de mortalidade por região. Interprete os resultados.