

Lista 07/08

Ana C V de Melo - Alexandre Locci

MAC113 - FEA diurno - 2020

objetivo: programação com ifElse, While e funções

Para fazer esses exercícios V. precisa ter assistido às aulas da semana e/ou consultado material didático sobre os assuntos do objetivo.

Exercício 1

1. Abra o programa RStudio.
2. Na janela de *Files* (canto direito inferior) crie um diretório com seu nome/número da lista.
3. Copie do *moodle* todos os arquivos disponíveis da Lista atual. Veja que você tem pelo menos um script (programa) para a aula.
4. Atualize o diretório de execução do RStudio para a sessão atual: clique em *Session/Set Working Directory/Choose Directory...* e escolha o diretório onde colocou os seus arquivos.
5. Abra o script para iniciar a lista.

A sua tarefa de hoje será complementar o script para o que se pede nos próximos exercícios.

Exercício 2

Saque em caixa eletrônico - 1 cliente

Você deverá modificar o script dado para verificar se um usuário possui saldo em sua conta bancária para fazer um saque. Nesse banco, todos os clientes possuem cheque especial com o limite de R\$100,00. Por isso, eles podem sacar além do seu saldo atual em conta até este valor, mas, neste caso, o seu saldo se tornará negativo.

Mediante o saque efetuado, o cliente poderá obter uma das seguintes situações.

1. ‘TRANSAÇÃO ACEITA’: se o cliente possui saldo suficiente para efetuar o saque,
2. ‘CHEQUE ESPECIAL’: se o saldo do cliente junto com o valor disponível no cheque especial é suficiente para efetuar o saque, ou
3. ‘SALDO INSUFICIENTE’: se o saldo do cliente junto com o valor disponível no cheque especial NÃO é suficiente para efetuar o saque.

Sua Tarefa será:

1. Implementar uma função que verifica o status da transação (como descrito acima), e tem como
 - entrada: o saldo atual do cliente
 - saída: uma das situações (aceita, especial ou insuficiente)
2. Implementar um programa que:
 - recebe como entrada do teclado:
 - Número da conta bancária (inteiro positivo com 5 dígitos);

- Nome do cliente (character);
- Saldo da conta (double);
- Valor do saque (double).
- usando a função do item anterior, seu programa deverá imprimir:

A-B-Saldo:C

Onde:

- A – Nome do cliente;
- B – ‘TRANSAÇÃO ACEITA’, ‘CHEQUE ESPECIAL’ ou ‘SALDO INSUFICIENTE’;
- C – Saldo disponível após o saque

Observe que quando a transação tem o status ‘SALDO INSUFICIENTE’, o saldo continua o mesmo já que não poderá ser debitado.

Aqui segue o que o seu script deve obter como resultado para os seguintes dados:

```
Digite o número da conta corrente:12345
Digite o nome do cliente: maria silva
Digite o saldo do cliente: 300
Digite o valor do saque: 200
```

```
12345 -- maria silva -- TRANSAÇÃO ACEITA -- Saldo: 100
```

Outra execução:

```
Digite o número da conta corrente:12345
Digite o nome do cliente: maria silva
Digite o saldo do cliente: 300
Digite o valor do saque: 450
```

```
12345 -- maria silva -- SALDO INSUFICIENTE -- Saldo: 300
```

Veja que V. poderá usar vários outros dados de entrada para testar o seu programa.

Exercício 3

Saque em caixa eletrônico - a partir de uma planilha com vários clientes

Semelhante ao **Exercício 2**, neste exercício vamos imprimir o status da transação de saques de clientes. Contudo, as informações sobre a transação de saque de cada cliente serão fornecidas em uma planilha. Além disso, a planilha contém informações de vários clientes.

Sua Tarefa será:

1. Implementar um programa que:
 - a. lê a planilha *clientes.saques.csv*, a qual contém as seguintes colunas:

Conta | Nome | Saldo | Saque

e cada linha da planilha contém essa informação para um cliente. Veja o conteúdo da planilha:

	Conta	Nome	Saldo	Saque
1	12345	maria diamante	200.00	150.0
2	21345	mariana coral	300.00	300.0
3	32145	pedro perola	50.25	150.0

4	42315	jose ametista	20.32	157.0
5	52341	joao onix	36.55	80.3
6	11345	ada rubi	850.00	500.0

b. para cada cliente na planilha, seu programa deverá imprimir:

A-B-Saldo:C

Onde:

- A - Nome do cliente;
- B - 'TRANSAÇÃO ACEITA', 'CHEQUE ESPECIAL' ou 'SALDO INSUFICIENTE';
- C - Saldo disponível após o saque

Veja que a planilha usada aqui é apenas um exemplo. Ela poderia ter 10, 100, 20,... qualquer número de linhas. Então, V. precisará usar um comando de repetição para conseguir fazer as mesmas tarefas para cada um dos clientes, independentemente de quantas linhas (clientes) a planilha tenha.

Exercício 4

Saque em caixa eletrônico - coluna com status da transação

Neste exercício, vamos criar um novo data-frame com toda a informação da planilha do exercício anterior, acrescida com uma coluna (*Status*), a qual deve conter o status da transação do cliente. Por exemplo, para a planilha *clientes.saques.csv* usada anteriormente, o seu script deverá gerar o seguinte data-frame:

	Conta	Nome	Saldo	Saque	Status
1	12345	maria diamante	200.00	150.0	TRANSAÇÃO ACEITA
2	21345	mariana coral	300.00	300.0	TRANSAÇÃO ACEITA
3	32145	pedro perola	50.25	150.0	CHEQUE ESPECIAL
4	42315	jose ametista	20.32	157.0	SALDO INSUFICIENTE
5	52341	joao onix	36.55	80.3	CHEQUE ESPECIAL
6	11345	ada rubi	850.00	500.0	TRANSAÇÃO ACEITA

Exercício 5

Criação de Novas Funções sobre Vetores

Existem várias funções predefinidas em R que podemos fazer as nossas próprias. Além disso, muitas vezes precisamos de variações no comportamento das funções, não providas em funções predefinidas. Aqui você irá criar novas funções para fazer operações sobre vetores (algumas delas já existem em R) para entender que sempre conseguimos criar novas funções com os comandos básicos da linguagem.

Para calcular o tamanho dos vetores (quantidade de elementos nos vetores), temos uma função predefinida em R que já faz essa operação: *length()*

a. criar os vetores e verificar os tamanhos de cada um deles

```
v1 <- as.integer(seq(1,15, length.out = 10))
v2 <- c(3:5, 23, NA, 14, NA)
tam_v1 <- length(v1)
tam_v2 <- length(v2)
print(v1)
```

```
## [1] 1 2 4 5 7 8 10 11 13 15
cat("tamanho do vetor v1:", tam_v1)

## tamanho do vetor v1: 10
print(v2)

## [1] 3 4 5 23 NA 14 NA
cat("tamanho do vetor v2:", tam_v2)

## tamanho do vetor v2: 7
b. soma de todos os elementos do vetore - eliminar os valores não definidos

soma_vetores <- function(vet){
  soma <- 0

  ...

  return(soma)
}

soma_v2 <- soma_vetores(v2)
print(v2)

## [1] 3 4 5 23 NA 14 NA
print(soma_v2)

## [1] 49
c. soma de dois vetores do mesmo tamanho

soma_2vetores_tam_igual <- function(vet1,vet2){
  vresult <- NULL

  ...

  return(vresult)
}

soma_vets <- soma_2vetores_tam_igual(v1,v1)
print(v1)

## [1] 1 2 4 5 7 8 10 11 13 15
print(soma_vets)

## [1] 2 4 8 10 14 16 20 22 26 30
```