

Introdução ao Programa R: Operações básicas de programação

Gilberto Rodrigues Liska; Josiane Rodrigues; Juliano Bortolini
gilbertoliska@ufscar.br; josirodrigues@ufscar.br;
julianobortolini@ufmt.br

Material de Apoio



Sumário

1 Introdução

2 Estrutura Condicional

3 Estrutura de Repetição

O que veremos nessa aula?

- (1) Conceitos básicos de programação, começando pelo módulo da divisão.
- (2) Estruturas condicionais: If, Else
- (3) Estruturas de repetição: For, While



Fonte: <https://pixabay.com/images/id-2298286/>

Módulo da Divisão

- O **Módulo da Divisão** é o operador que extrai o resto da divisão. A linguagem em R, bem como a maioria das linguagens de programação, disponibiliza um operador para este fim.
- Matematicamente, o módulo da divisão é formalizado por

$$y = w \bmod x, \quad (1)$$

que significa que y é o módulo (resto) da divisão de x por w .

- O sinal de porcentagem `%` é o operador módulo em várias linguagens de programação. No R, o comando é `%%`



Fonte: <https://pixabay.com/images/id-1300226/>

Módulo da Divisão

Uma das utilidades do operador módulo é a possibilidade de identificar números pares e ímpares.

Exemplo 1

Vamos resolver a seguinte operação envolvendo módulo da divisão:

$$3 \bmod 10$$

SOLUÇÃO:

Não é difícil de perceber que ao dividir 10 por 3, teremos como resultado o número 1.

Crie um script (tela 1) no R. Essa operação é feita da seguinte forma. Copie e cole os códigos abaixo no seu script.

Exemplo 1 (a) no R

```
#### forma 1
num1=10
num2=3
divisao=num1/num2; divisao
resto=num1%%num2; resto
```

Módulo da Divisão

- Existem várias formas de fazer o procedimento anterior no R, mas todas utilizando o `%/%`.
- Será apresentada uma outra forma de se fazer esse procedimento, que é o uso de funções.
- A ideia de apresentar essa outra possibilidade é, no momento, de apenas mostrar a flexibilidade que o R apresenta para programação.

Exemplo 1 (b) no R

```
#### forma 2
modm=function(num1, num2){

  divisao=num1/num2
  resto=num1%/%num2

  return(list(divisao=divisao,
              resto=resto))
}

modm(10,3)
#tente mais uma vez com os números 9 e 3
modm(9,3)
```

Observações

- Qual a diferença entre os procedimentos feitos no Exemplo 1 (a) e (b)?
- No **Exemplo 1 (a)**, toda vez que quisermos utilizar o módulo da divisão, temos que eleger os objetos *num1* e *num2* e executar as quatro linhas de comando.
- No **Exemplo 1 (b)**, uma vez construída a função, chamada de *modm*, toda vez que quisermos utilizar o módulo da divisão, basta executar apenas a última linha do código informando os dois números dos quais se deseja fazer a operação.
- **IMPORTANTE:** A principal vantagem do que foi feito no exemplo 1 (b) é tornar a usabilidade do programa R mais fluída, não tendo que repetir os mesmos comandos toda vez que se quiser fazer uma operação.
- *Contudo, não se assuste!* A construção de funções será abordada com mais detalhes na quarta aula.



Fonte:
<https://pixabay.com/images/id-1184858/>

Módulo da Divisão

Números pares

- Podemos saber se um número é par com o uso do operador módulo, para isto, basta verificarmos se a divisão possui resto, ou melhor, se o valor retornado é diferente de zero.
- Isso significa que uma condição pode ser imposta ao programa, de tal modo que o programa, ao realizar a divisão por 2, informa se o número é par ou ímpar.

Exemplo 2

Para ilustrar a ideia e utilizando o módulo da divisão, vamos resolver

$$2 \bmod 15$$

SOLUÇÃO:

Não é difícil de perceber que ao dividir 15 por 2, teremos como resultado o número 1, o que indica que o número é **ímpar**.

Exemplo 2 (a) no R

```
num1=15  
num2=2
```

```
divisao=num1/num2; divisao  
resto=num1%%num2; resto  
[1] 1
```


Módulo da Divisão

- Com um raciocínio análogo ao feito no Exemplo 1 (b), podemos fazer a mesma coisa com esse exemplo 2
- **QUESTÃO:** E se quiséssemos que o programa automaticamente identifique se o número é par ou ímpar?
- **RESPOSTA:** Precisamos impor estruturas condicionais.
- Copie e cole o código ao lado e veja o que acontece.

Exemplo 2 (b) no R

```
num1=16 # digite o número
if(num1%%2==0)
  print("Número par") else
  print("Número ímpar")
```

```
## mais elegante
modmpar=function(num1){
  num2=2

  if(num1%%num2==0)
    print(paste("O número", num1,"é par"))
  else
    print(paste("O número", num1,"é ímpar"))
}
modmpar(16)
```

Sumário

- 1 Introdução
- 2 Estrutura Condicional
- 3 Estrutura de Repetição

Estruturas Condicionais

- Uma estrutura de condição verifica a condição de argumentos passados e executa um comando caso a condição seja verdadeira.
- Antes de entrarmos na análise da estrutura em R vamos conhecer os Operadores Condicionais do R. Os Operadores Condicionais são utilizados para fazer as comparações dos valores que são passados e retornam o valor **Verdadeiro** ou **Falso**.

Tabela 1: Operadores condicionais

Operador	Tipo	Valor
==	Igualdade	Verifica a igualdade entre dois valores
!=	Diferença	Verifica a diferença entre dois valores
>	Comparação	Verifica se o valor A é maior que o valor B
<	Comparação	Verifica se o valor A é menor que o valor B
<=	Comparação	Verifica se o valor A é menor ou igual ao valor B
>=	Comparação	Verifica se o valor A é maior ou igual ao valor B

Estruturas Condicionais

IF

- A sintaxe do IF em R é a seguinte:

```
se (condição)
  < bloco de código >
fim
```

Exemplo 3

Construa um programa que identifique números menores do que 10 e imprima na tela o resultado.

```
se (num < 10 )
  < print("Menor que 10") >
fim
```

Exemplo 3 (a) no R

```
num=8 # digite o número
if(num<10)
  print("Menor que 10")
```

```
#note que
num<10
[1] TRUE
```

Exemplo 3 (b) no R

```
## mais elegante
n10=function(num){
  if(num<10)
    print(paste("menor que 10"))
}
n10(8)
[1] "menor que 10"
n10(10)
n10(11)
```

Observações

- Qual a diferença entre os procedimentos feitos no Exemplo 3 (a) e (b)?
- No **Exemplo 3 (a)**, toda vez que quisermos avaliar se a condição ($num < 10$) se verifica, temos que eleger o objeto *num* e executar as três linhas de comando.
- No **Exemplo 3 (b)**, uma vez construída a função, chamada de *n10*, toda vez que quisermos saber se o número é inferior a 10, basta executar apenas a última linha do código informando o número que se deseja verificar a condição ($num < 10$).
- Note que no **Exemplo 3 (a)**, foi retornado o resultado de uma condição (**TRUE**). Portanto, se a condição for verdadeira (**TRUE**), imprime o resultado, mas se for **FALSE**, nada acontece. Todas as estruturas de programação (condicionais e de repetição) trabalham com esses operadores condicionais, como apresentado na tabela 1.



Fonte: <https://pixabay.com/images/id-294525>

Estruturas Condicionais

Exemplo 4

Crie uma rotina que pergunte quantos anos tem um carro e que, caso o tempo seja de até 3 anos, o carro seja considerado novo.

```
Início: tempo  
se (tempo < 3 )  
    < print("Carro novo") >  
fim
```

Exemplo 4 (a) no R

```
tempo=2 # digite o número  
if(tempo<3)  
    print("Carro novo")  
[1] "Carro novo"
```

Exemplo 4 (b) no R

```
tempo=function(num){  
    if(num<3)  
        print("Carro novo")  
}  
tempo(8)  
tempo(2)  
[1] "Carro novo"
```

Estruturas Condicionais

IF - ELSE

- Quando a condição não é atendida com o comando **if**, é necessário utilizarmos o comando **else**;
- Neste caso, estamos trabalhando com uma estrutura condicional composta que executa um comando quando a condição for verdadeira e outra condição quando for falsa.
- A sintaxe do IF - ELSE em R é a seguinte:

se (Condição)

< bloco de código >

se não

< bloco de código >

fim se



Fonte: <https://pixabay.com/images/id-3173613/>

Estruturas Condicionais

Exemplo 5

Digite um número e identifique se ele é menor do que 10 ou no mínimo 10.

Início: tempo

se (num < 10)

< print("Menor que 10") >

se não

< print("No mínimo 10") >

fim se

Exemplo 5 (a) no R

```
num=8 # digite o número
if(num<10)
  print("Menor que 10") else
  print("No mínimo 10")
[1] "Menor que 10"
```

Exemplo 5 (b) no R

```
n10=function(num){
  if(num<10)
    print("menor que 10")
  else
    print("no mínimo 10")
}
n10(8)
n10(10)
n10(12)
```


Estruturas Condicionais

Exemplo 6

Criar uma rotina que pergunte quantos anos tem um carro. Se ele tiver mais de 3 anos, é considerado velho e, caso contrário, é considerado novo.

```
Início: tempo  
se (tempo <= 3)  
    < print("Carro novo") >  
se não  
    < print("Carro velho") >  
fim se
```

Exemplo 6 (a) no R

```
tempo=8  
if(tempo<=3)  
  print("Carro novo") else  
  print("Carro velho")  
[1] "Carro velho"
```

Exemplo 6 (b) no R

```
tempo=function(num){  
  if(num<=3)  
    print("Carro novo")  
  else  
    print("Carro velho")  
}  
tempo(2)  
tempo(8)
```

Observações

- Copie e cole os códigos a seguir, denominado de “Exemplo 6 (c) no R” no seu script.

Exemplo 6 (c) no R

```
tempo=function(){  
  tempo=readline(prompt="Quantos anos tem seu carro? ")  
  tempo=as.integer(tempo)  
  
  if(tempo<=3)  
    print("Carro novo")  
  else  
    print("Carro velho")  
}  
tempo()
```



Fonte: <https://pixabay.com/images/id-294525/>

- Note que ao executar a última linha, a entrada da informação se dá por meio do *prompt* (tela 3), na qual é feita a pergunta e o *prompt* espera até que uma resposta seja dada.

Observações

- A estrutura condicional trabalhada pode ser simplificada. O comando no R que faz isso é o *ifelse*.

ifelse(Condição, Resultado se verdadeiro, Resultado se falso)

Exemplo 6 (d) no R

```
tempo=5  
ifelse(tempo<=3, "Carro novo", "Carro velho")
```

Exemplo 6 (e) no R

```
tempo=function(num){  
  ifelse(num<=3, "Carro novo", "Carro velho")  
}  
tempo(5)
```

Exemplo 6 (f) no R

```
tempo=function(){  
  tempo=readline(prompt="Quantos anos tem seu carro? ")  
  ifelse(tempo<=3, "Carro novo", "Carro velho")  
}  
tempo()
```

Estruturas Condicionais

Exemplo 7

Criar uma rotina que calcule a média entre duas notas, ou seja,

$$n = \frac{n_1 + n_2}{2}$$

Se a média for maior ou igual a 6, exiba a mensagem “Parabéns”. Se a média for menor que 6, exiba a mensagem “Estude Mais”.

Início: $n_1, n_2, n = \frac{n_1 + n_2}{2}$

se ($n \geq 6$)

< print("Parabéns") >

se não

< print("Estude Mais") >

fim se

Exemplo 7 (a) no R

```
n1=5
n2=6
n=(n1+n2)/2
print(paste("A sua média foi",n))
print(ifelse(n>=6,"Parabéns",
             "Estude Mais"))
[1] "A sua média foi 6.5"
[1] "Parabéns"
```

Exemplo 7 (b) no R

```
n=function(n1,n2){
  n=(n1+n2)/2
  print(paste("A sua média foi",n))
  print(ifelse(n>=6,"Parabéns",
              "Estude Mais"))
}
n(5,8)
n(3,8)
```

Estruturas Condicionais

IF - ELSE IF - ELSE

- Se existir mais de uma condição alternativa que precisa ser verificada, devemos utilizar o **else if** para avaliar as expressões intermediárias antes de usar o **else**;
- Neste caso, estamos trabalhando com estruturas condicionais aninhadas, que são várias condições em cascatas, ou seja, um **if** dentro de outro **if**.

A sintaxe do IF - ELSE IF - ELSE é a seguinte

Início
se (condição)
 < bloco de código >
senão se (condição)
 < bloco de código >
senão
 < bloco de código >
fim

Estruturas Condicionais

Exemplo 8

Criar uma rotina que peça para o usuário digitar um número e identifique se ele é menor do que 10, igual a 10 ou maior que 10.

```
Início: num
se ( num < 10 )
    < print("Menor que 10") >
senão se ( num == 10 )
    < print("Igual a 10") >
senão
    < print("Maior que 10") >
fim
```

Exemplo 8 (a) no R

```
num=10
if(num<10)
  print("menor que 10") else if(num==10)
  print("Igual a 10") else
  print("maior que 10")
[1] "Igual a 10"
```

Exemplo 8 (b) no R

```
n10=function(num){
  if(num<10)
    print("menor que 10")
  else if(num==10)
    print("Igual a 10")
  else print("maior que 10")
}
n10(10)
[1] "Igual a 10"
```

Sumário

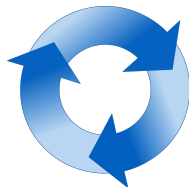
- 1 Introdução
- 2 Estrutura Condicional
- 3 Estrutura de Repetição

Estruturas de Repetição

- **QUESTÃO:** Como deveríamos construir um programa que fosse executado até que uma determinada condição fosse satisfeita?
- **RESPOSTA:** Devemos usar estruturas de repetição.

Laços de Repetição

- Laços (*loops*) são estruturas de repetição, geralmente usadas para processar coleções de dados, tais como linhas de um arquivo ou registros de um banco de dados, que precisam ser processados por um mesmo bloco de código.
- As mais comuns são o “FOR” e “WHILE”.



Fonte: <https://pixabay.com/images/id-311551/>

Estruturas de Repetição

FOR

- É a estrutura de repetição mais usada no R. A instrução aceita não só sequências estáticas, mas também sequências geradas por iteradores;
- Iteradores são estruturas que permitem iterações, ou seja, acesso aos itens de uma coleção de elementos, de forma sequencial.

FOR

- A sintaxe do FOR em R é a seguinte:

```
for (referência) in < sequência >  
  < bloco de código >  
fim
```

Estruturas de Repetição

Exemplo 9

Crie um programa que mostre em tela a sequência numeral de 1 a 5

```
for (i) in < 1:5 >  
  < print(i) >  
fim
```

Exemplo 9 no R

```
for(i in seq(1,5)){  
  print(i)  
}  
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5
```

```
## outra forma  
gera=function(num1,num2,pula){  
  for(x in seq(num1,num2,pula)){  
    print(x)  
  }  
  print("FIM")  
}  
gera(1,5,1)
```

Estruturas de Repetição

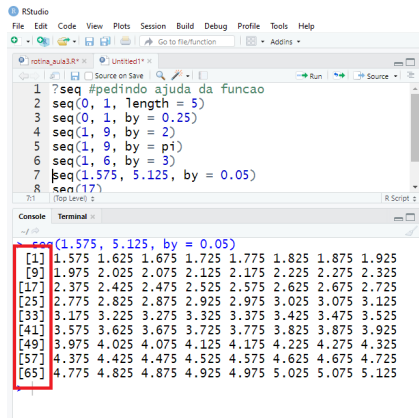
- A função $seq(m, n, p)$, é muito útil em laços, pois retorna uma lista de inteiros, começando em m e menores que n , em passos de comprimento p , que podem ser usados como sequência para o laço.
- Copie e cole os exemplos a seguir no seu script e execute cada linha de comando.
- Qual a diferença entre as linhas 2 e 3?
- Se não informarmos o comprimento do passo, o que o R faz por padrão? (Veja isso na última linha de comando)

Exemplo 10 no R

```
?seq #pedindo ajuda da funcao
seq(0, 1, length = 5)
seq(0, 1, by = 0.25)
seq(1, 9, by = 2)
seq(1, 9, by = pi)
seq(1, 6, by = 3)
seq(1.575, 5.125, by = 0.05)
seq(17)
```

Observação

- Voltando aos comandos do exemplo 10, por exemplo na linha 7, muitos números aparecem na tela 3 do programa.
- Os números indicados por [xx], posicionados na porção esquerda da tela 3, são indicadores de linha.
- Por exemplo, na figura 1, o número [9] dentro da caixa vermelha indica que o elemento 1.975 está na posição 9 da sequência de números gerada.
- **IMPORTANTE:** Essa numeração pode ser diferente em telas de diferentes usuários, o que não é um problema.



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
1 ?seq #pedindo ajuda da funcao
2 seq(0, 1, length = 5)
3 seq(0, 1, by = 0.25)
4 seq(1, 9, by = 2)
5 seq(1, 9, by = p1)
6 seq(1, 6, by = 3)
7 seq(1.575, 5.125, by = 0.05)
8 seq(17)
[1] 1.575 1.625 1.675 1.725 1.775 1.825 1.875 1.925
[9] 1.975 2.025 2.075 2.125 2.175 2.225 2.275 2.325
[17] 2.375 2.425 2.475 2.525 2.575 2.625 2.675 2.725
[25] 2.775 2.825 2.875 2.925 2.975 3.025 3.075 3.125
[33] 3.175 3.225 3.275 3.325 3.375 3.425 3.475 3.525
[41] 3.575 3.625 3.675 3.725 3.775 3.825 3.875 3.925
[49] 3.975 4.025 4.075 4.125 4.175 4.225 4.275 4.325
[57] 4.375 4.425 4.475 4.525 4.575 4.625 4.675 4.725
[65] 4.775 4.825 4.875 4.925 4.975 5.025 5.075 5.125
  
```

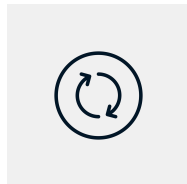
Figura 1: Tela de um dos comandos do exemplo 10.

Estruturas de Repetição

WHILE

- O comando `while` faz com que um conjunto de instruções seja executado enquanto uma condição é atendida. Quando o resultado dessa condição passa a ser falso, a execução do *loop* é interrompida.
- A sintaxe do laço `WHILE` em R é a seguinte:

```
while (condição)  
  < bloco de código >  
fim
```



Fonte: <https://pixabay.com/images/id-2935463/>

Estruturas de Repetição

Exemplo 11

Crie um programa que mostre em tela a sequência numeral de 1 a 10

```
Início: x = 1
while ( x < 11 )
    < print(x) >
fim
```

Exemplo 11 (a) no R

```
x=1
while(x<11){
  print(x)
  x=x+1
}
```

Exemplo 11 (b) no R

```
gera=function(num1,num2,pula){
  while(num1 < num2){
    print(num1)
    num1=num1+pula
  }
  print("FIM")
}
gera(1,11,1)
```

Exercício

Exercício

Criar uma rotina que pergunte:

- Digite um valor
- Quer continuar? E crie duas opções (SIM/NÃO)
- Se escolher a opção SIM \Rightarrow Digite um valor
- Se não \Rightarrow FIM

Início: $r = "S"$

while ($r = "S"$)

< print("Digite um valor") >

< print("Quer continuar?"[S/N]) >

fim



Fonte: <https://pixabay.com/images/id-4997565/>

Exercício: Solução

Exercício (a) no R

```
r="S"
while(r=="S"){
  n=readline(prompt="Digite um valor: ")
  n=as.integer(n)
  r=readline(prompt="Quer continuar? [S/N] ")
  r=as.factor(r)
}
print("FIM")
```

Exercício (b) no R

```
p1=function(){
  r="S"
  while(r=="S"){
    n=readline(prompt="Digite um valor: ")
    n=as.integer(n)
    r=readline(prompt="Quer continuar? [S/N] ")
    r=as.factor(r)
  }
  print("FIM")
}
p1()
```



Fonte: <https://pixabay.com/images/id-4997565/>