

Programação Estruturada em R

Nelson S. dos Santos

Núcleo de Ciência de Dados e Computacional em Economia e Finanças
Faculdade de Ciências Econômicas
Universidade Federal do Rio Grande do Sul

1 de novembro de 2024

Sumário

- 1 Introdução
- 2 Teorema de Böhm-Jacopini
- 3 Estrutura de sequência
- 4 Estruturas de decisão
- 5 Estruturas de repetição
- 6 Referências

Introdução

Introdução

Problema

Como organizar programas escritos usando apenas a estrutura de sequência (isto é, justaposição do código fonte sequencialmente em entrada, processamento e saída) para facilitar seu entendimento e modificação? desorganizados.

Contexto do Problema

Programas sequenciais exigem a repetição de código um grande número de vezes o que pode gerar erros difíceis de serem detectados.

Solução

Usar outras estruturas de controle de fluxo de execução do programa, além da sequência, que permitam reduzir a repetição de código.

As referências básicas para este trabalho são [\[R Core Team \(2021a\)\]](#) e [\[R Core Team \(2021b\)\]](#).

Teorema de Böhm-Jacopini

Teorema de Böhm-Jacopini

Programação estruturada: teorema de Böhm-Jacopini I

Qualquer problema computável em uma Máquina de Turing pode ser resolvido com apenas três operações:

- 1 sequência
- 2 decisão
- 3 repetição

Tais operações são denominadas estruturas de controle de fluxo de execução.

Programação estruturada: teorema de Böhm-Jacopini II

- Diz-se que um programa é estruturado se é escrito usando apenas as três estruturas de Böhm e Jacopini.
- Programação estruturada é o paradigma de programação que estabelece todo programa deve utilizar no seu algoritmo apenas estruturas de Böhm e Jacopini

Estrutura de sequência

Estrutura de sequência

Sequência

A estrutura de sequência consiste simplesmente em escrever um comando após o outro na ordem em que se deseja que eles sejam executados sem nunca haver repetição de um comando ou múltiplas execuções possíveis em um dado comando.

Exemplo

```
x <- scan(what = double())  
print(x)
```

Observe que a execução dos comandos é feita na exata ordem em que são escritos no programa.

Estruturas de decisão

Estruturas de decisão

Estrutura de decisão if

Esta estrutura de decisão permite o programa decidir os comandos que serão executados no tempo de execução do programa a depender do estado do programa.

Veja o exemplo a seguir.

Estrutura de decisão if: exemplo

```
print("Digite um número de 1 a 10.")  
x <- scan(file = "", what = double(), nmax=1)  
  
if (x == 4){  
  print("Você digitou o número 4.")  
} else if (x == 5){  
  print("Você digitou o número 5.")  
} else{  
  print("Você não digitou nem 4 nem 5.")  
}
```

Estruturas de repetição

Estruturas de repetição

Estrutura de controle *while*

A sintaxe da estrutura *while* em R é mostrada a seguir:

```
while (x in N){  
    <comandos>  
}
```


Estrutura de controle while: exemplo

Faça um programa que imprima os números de 1 a 10.

```
i ← 1
```

```
while (i < 10){  
    print(i)  
    i ← i + 1  
}
```

Estrutura de controle repeat

A sintaxe da estrutura de controle *repeat* é mostrada a seguir:

```
repeat{  
    <comandos>  
    if(x == 3){  
        break  
    }  
}
```

Estrutura de controle repeat: exemplo

Faça um programa que imprima os números de 1 a 10.

```
i <- 1
repeat{
  print(i)
  i <- i + 1
  if (i == 10){
    break
  }
}
```

Estrutura de controle for: sintaxe

```
for (x in N){  
    <comandos>  
}
```

Estrutura de controle for: exemplos

Faça um programa que imprima os números de 1 a 10.

```
indices <- c(1,2,3,4,5,6,7,8,9)
i <- 1

for (i in indices){
  print(i)
  i <- i + 1
}
```

ReferênciasR

Referências

Referências



R Core Team. *An Introduction to R*, The R Project for Statistical Computing, 2021, Disponível em [link](#).



R Core Team. *R Language Definition*, The R Project for Statistical Computing, 2021, Disponível em [link](#).