

Programação Procedural em R

Nelson Seixas 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 Definições Preliminares
- 3 Funções e Programação Procedural em R
- 4 Programação modular em R
- 5 Exercícios
- 6 Referências

Introdução

Problema

Como escrever programas com grande número de linhas no código-fonte sem se perder na navegação no arquivo e conseguir encontrar bugs facilmente?

Contexto do problema

- Normalmente, é difícil navegar em longos códigos-fonte e achar partes específicas dentro do código.
- Além disso, mesmo quando se encontra o bug é muito difícil saber o estado do programa (isto é, o valor de todas as variáveis em um dado momento do tempo de execução) para poder identificar exatamente o que está acontecendo.
- Tal dificuldade se eleva à medida que o código-fonte cresce, tornando praticamente impossível fazer qualquer alteração no programa quando o código-fonte passa de milhares de linhas de código.

Solução

Para resolver esse problema, a solução encontrada foi dividir o código-fonte em unidades menores chamadas de subprogramas.

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

Definições Preliminares

Programação Procedural: definição

Programação procedural é o paradigma de programação onde o código-fonte do programa é dividido em subprogramas de dois tipos: funções e procedimentos.

Tipos de Subprogramas

- Há dois tipos de subprogramas: procedimentos e funções.
- Procedimentos são uma sequência de comandos a qual se atribui um nome. Toda vez que se deseja executar aquela sequência de comandos, utiliza-se o nome definido e seus comandos são executados na sequência definida no procedimento.
- Funções são procedimentos cujo resultado da execução retorna um valor para o corpo principal do programa.
- R, por ser uma linguagem funcional, implementa apenas funções.

Funções: definição

Uma função é um objeto que recebe um valor de entrada e fornece um valor de saída. Propriedades:

- Servem para armazenar trechos de código que serão reutilizados em diversas partes do programa.
- Facilitam a leitura do programa.
- Evita a repetição do código da mesma em diversas partes do programa.
- Implementa o princípio de programação denominado pela sigla em inglês DRY ("Don't Repeat Yourself")

Funções e Programação Procedural em R

Escrevendo uma função em R

- Sintaxe de definição de funções

```
nome_da_função <- function(a,b){  
  <comando 1>  
  <comando 1>  
  .  
  .  
  .  
  <comando N>  
  return(valor_calculado)  
}
```

Escrevendo uma função em R: exemplo 1

Função que soma os valores da sua entrada.

```
# Definição da função  
soma <- function(a,b){  
    return(a+b)  
}  
  
# Chamada da função  
soma(2, 3)
```

Escrevendo uma função em R: exemplo 2

Função que soma o produto dos valores de entrada.

```
# Definição da função
somaproduto <- function(x,y,z,w){
    return(x*y+z*w)
}

# Chamada da função
somaproduto(1,2,3,4)
```

Programação Modular

Problema

- À medida que os subprogramas de um programa crescem em número e tamanho, a programação procedural passa a enfrentar as mesmas dificuldades de manutenção do código-fonte encontradas na programação estruturada.
- A solução para este problema foi dar aos tradutores (interpretadores ou compiladores) das linguagens de programação a capacidade de criar um único código executável a partir de vários arquivos de código-fonte.
- Programação modular é uma técnica de desenho de algoritmo onde os subprogramas do código-fonte de um programa são divididos em vários arquivos distintos denominados módulos.

Programação modular em R: definição

É um paradigma de programação voltado para escrita de programas grandes onde o código fonte, além de ser procedural, é dividido em diversos arquivos (denominados módulos), que guardam cada um deles funções correlatas entre si.

Programação modular em R: características

- É um aprofundamento do paradigma de programação procedural.
- O algoritmo de solução do problema posto é escrito em um arquivo principal (denominado módulo principal) que chama as funções definidas nos demais módulos.
- Facilita ainda mais a leitura, entendimento e manutenção do código.
- Cada módulo é chamado pelo módulo principal pelo comando `source('nomedoarquivo.R')`.

Pacotes: definição

Pacotes são conjuntos de módulos cuja utilização requer instalação no ambiente R e que servem para adicionar funcionalidades ao ambiente como um todo.

Pacotes: características

- Podem ser escritos por qualquer pessoa;
- para funcionarem, é preciso que estejam em local que possa ser lido pelo R e instalados;
- Uma vez instalado, um pacote pode ser usado em qualquer programa que será executado no ambiente R, e
- Para usar um pacote, basta adicionar a instrução `library('nomedopacote')` que determina o carregamento do pacote desejado.

Alguns pacotes importantes disponíveis em R

Os pacotes a seguir permitem realizar todas as operações de inferência econométrica com os mais usados modelos econométricos.

- GGLOT2 - Visualização de dados - [link](#)
- URCA - Testes de raiz unitária e estacionariedade - [link](#)
- PLM - Dados em painel - [link](#)
- VARS - Vetores auto-regressivos - [link](#)

Exercícios

Exercícios

Exercício 1

Instale o Rstudio e o utilize para resolver os exercícios a seguir.

- 1 Faça os exercícios de [Análise Estatística de Ações 1](#)
- 2 Faça os exercícios de [Análise Estatística de Ações 2](#)
- 3 Faça os exercícios de [Análise Estatística de Ações 3](#)

Referências

Referências

Referências



R Core Team. *An Introduction to R*, The R Project for Statistical Computing, 9, 2021, Disponível em [link](#).



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