

Introdução à linguagem S: programação sequencial no ambiente 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 Conceitos Fundamentais
- 3 A linguagem S
 - A semântica
 - O léxico
 - A sintaxe
- 4 O Ambiente Estatístico R
- 5 A estrutura do código-fonte R
- 6 O Modelo de Dados de R
 - Tipos de dados
 - Objetos
 - Principais objetos de R
- 7 O Modelo de Execução de R

Introdução

Problema

Apresentar programação sequencial na linguagem S.

Importância do problema

A linguagem S é a principal linguagem de programação estatística utilizada atualmente. Ela é popularmente conhecida pelo seu uso no ambiente estatístico R.

Ademais, programação sequencial é o fundamento para o aprendizado de programação em qualquer linguagem.

- Introdução
- Conceitos Fundamentais
- A linguagem S
- O Ambiente Estatístico R
- A estrutura do código-fonte R
- O Modelo de Dados de R
- O Modelo de Execução de R
- Entrada e saída de dados
- Referências

Metodologia de solução do problema

Apresentaremos a sintaxe e a semântica da linguagem S para fazer entrada, processamento e saída de dados.

Resultados esperados

Ao final desta apresentação o estudante deverá ser capaz de escrever e executar programas sequenciais na linguagem S por meio do ambiente estatístico R.

As referências principais utilizadas aqui são [[R Core Team \(2021a\)](#)] e [[R Core Team \(2021b\)](#)].

- Introdução
- Conceitos Fundamentais
- A linguagem S
 - O Ambiente Estatístico R
- A estrutura do código-fonte R
 - O Modelo de Dados de R
 - O Modelo de Execução de R
- Entrada e saída de dados
- Referências

Conceitos Fundamentais

Máquina abstrata

É um modelo matemático que especifica a arquitetura de um computador no que respeita à forma como este armazena dados (seu modelo de dados) bem como a execução das instruções (modelo de execução) que lhe são passadas.

Linguagem de programação

Uma linguagem de programação é um conjunto de símbolos (léxico) e regras (sintaxe) para relacionar tais símbolos a fim de passar instruções com significado (semântica) precisamente definido para uma máquina abstrata pré-estabelecida.

Implementação de uma linguagem de programação

Uma implementação de uma linguagem de programação A é um software capaz de traduzir as instruções em A para linguagem da máquina que deverá executar tais instruções.

Uma implementação pode ser um interpretador, um compilador ou uma máquina virtual.

Dialeto de uma linguagem de programação

Dada uma linguagem de programação A, um dialeto de A é uma pequena variação em seu léxico, sintaxe ou semântica, criada normalmente por meio de uma implementação que, não altera o modelo de dados ou de execução previsto originalmente na especificação formal da linguagem A.

Ambiente operacional

Um ambiente operacional é um sistema com finalidade específica formado por um conjunto de aplicativos capazes de se comunicar entre si por meio de uma linguagem de programação comum dentro do sistema para executarem tarefas de modo coordenado. A linguagem do ambiente é chamada de linguagem de script.

Exemplos: MS-Office, Eviews, GRETSL, SPSS etc.

Um script é um programa escrito na linguagem de programação de um ambiente operacional. Por isso, linguagem de programação de ambientes operacionais são comumente denominadas de linguagens de script.

Paradigmas de programação

Um paradigma de programação é um conjunto de regras para organização do código fonte de uma linguagem de programação.

Diz-se que uma linguagem de programação suporta o paradigma X se ela fornece facilidades sintáticas para escrita do código-fonte segundo as regras de X.

Paradigma de programação funcional

Diz-se que o código-fonte de um programa segue o paradigma funcional se é formado apenas por expressões e funções matematicamente bem definidas e executadas sobre valores imutáveis em memória.

A execução do programa consiste, por isso, na avaliação de valores de expressões e funções os quais, por definição, sempre retornam um valor ao usuário. Estes valores podem ser omitidos da tela se assim for conveniente.

Computação científica e o Paradigma Funcional

- A realização de cálculos significa formalmente a determinação do valor de uma função.
- Cálculos compostos por vários cálculos simples onde o resultado de um deles é usado para iniciar o próximo são facilmente representados por funções compostas.
- O paradigma de programação funcional é o que melhor representa a composição de funções.

Estatística Computacional e o Paradigma Funcional

- A estatística computacional é basicamente computação científica aplicada a problemas estatísticos. Por isso, o desenvolvimento de programas para elaboração de cálculos estatísticos se adapta particularmente bem ao paradigma de programação funcional.
- Por isso, a linguagem S dá especial suporte ao paradigma funcional para a construção de códigos-fonte.
- Com efeito, todos os comandos na linguagem são, na verdade, funções!

A linguagem S

A linguagem S

- A linguagem S é uma linguagem de programação desenvolvida para a realização de análise exploratória de dados e inferência estatística.
- Segundo [[Becker \(2015\)](#)], a primeira especificação da linguagem S foi descrita em um manual interno do Bell Laboratories em 1977.

Objetos

A linguagem manipula regiões da memória do computador denominadas genericamente de **objetos**.

Objetos são caracterizados por atributos e seus respectivos valores. Concretamente, isto se materializa por meio de uma estrutura de dados lista de tabelas onde cada tabela relaciona um atributo a um valor. Os dados inseridos pelo usuário são armazenados como atributo valor de um objeto por ele criado.

Os valores dos objetos possuem tipos os quais definem as operações que podem ser realizadas com eles. Em R, este tipo de dado é chamado de **modo**.

Objetos II

Todo objeto possui pelo menos dois atributos: tamanho e modo. Além disso, o objetos possuem o atributo classe que designa as características específicas da estrutura de dados que armazena os valores dos objetos.

Operações com objetos

Como dito anteriormente, o modo do objeto define as operações que podem ser realizadas com eles.

Não obstante, as próprias operações são elas mesmas objetos manipulados pela linguagem como qualquer outro, posto que suas instruções estão armazenadas em memória.

Expressões

- Uma expressão é uma sequência de operações com objetos.
- Todo o código-fonte R é basicamente uma composição de expressões realizadas sobre os objetos, objetivando determinar o valor final desta expressão composta.
- A semântica da linguagem (isto é, a forma de cálculo do valor das expressões) é extraída da linguagem Scheme.

O léxico da linguagem

A linguagem utiliza símbolos ASCII que, concatenados, formam os identificadores dos objetos e das operações.

Estes código deve ser armazenado em um arquivo de texto com a extensão `.R`, sendo sua execução realizada pela invocação do interpretador de comandos do ambiente R.

A sintaxe

- A sintaxe da linguagem é construída para dar suporte tanto à programação procedural quanto à programação funcional.
- O suporte à programação procedural é dado por uma sintaxe fortemente inspirada pela linguagem C
- Para facilitar a programação funcional, são oferecidas construtos sintáticos (funções) para tornar o código uma composição de funções e expressões.

O Ambiente Estatístico R

As implementações *S* e *S-plus*

- A implementação completa da linguagem S se materializou no ambiente estatístico S definido em [Becker and Chambers (1978)].
- A evolução do ambiente estatístico S levou ao atual ambiente *S-plus* cuja linguagem de script foi definida em [Becker, Chambers e Wilks (1988)].
- A linguagem S passou por várias modificações ao longo do tempo, sendo sua especificação atual definida em [Chambers (1998)].
- Atualmente, a linguagem S suporta vários paradigmas de programação, tais como, estruturado, procedural, funcional e orientado a objetos com implementação dada pelo ambiente

O ambiente estatístico R

Em 1991, os professores de estatística da Universidade de Auckland (Austrália) Ross Ihaka e Robert Gentleman começaram a implementar a linguagem S por meio do desenvolvimento de um ambiente denominado R cuja primeira versão foi publicada em 1993.

Em 1995, R foi disponibilizado como um projeto de software livre da Free Software Foundation, ficando também conhecido como GNU S.

R pode ser entendido como uma versão de linha de comando e de código aberto do ambiente estatístico ***S-plus***

O dialeto R

R implementa um dialeto para a linguagem S diferente do originalmente implementado no ambiente *S-plus*.¹

O dialeto R tem como base a especificação da linguagem S exposta em [Becker, Chambers e Wilks (1988)] com uma semântica modificada para se assemelhar à linguagem de programação Scheme.

¹Tais diferenças estão relacionadas [aqui](#).

O dialeto R (cont.)

Os dados em R são armazenados por meio de uma estrutura denominada **objeto** que pode assumir diversos tipos para os quais estão definidas uma larga gama de métodos estatísticos definidos na própria linguagem.

Aos métodos implementados no pacote base do R foram acrescentados pela comunidade de pesquisa em estatística milhares de pacotes de funções implementando os principais métodos estatísticos e tipos de gráficos.

A estrutura do código-fonte R

A estrutura do código-fonte R I

As ideias comunicadas pela linguagem R são chamadas de objeto os quais se implementam como regiões de memória onde estão armazenados os dados.

Os objetos são denotados no código-fonte por identificadores formados por caracteres ASCII. Os identificadores são eles mesmos objetos R.

A estrutura do código-fonte R II

Os objetos são relacionados por meio de operadores que representam operações realizadas sobre eles para formar expressões ou funções matemáticas.

As operações que podem ser realizadas com um objeto são definidas pelo tipo de dado armazenado no objeto. Em R, este tipo de dado é chamado de **modo**.

O Modelo de Dados de R

Dados e seus tipos no mundo concreto

- Mundo concreto os dados se apresentam em diversos tipos (por exemplo, número, texto etc).
- Os tipos de dados caracterizam as operações que podem ser realizadas sobre eles. Por exemplo, a adição só pode ser realizada entre números, mas não entre caracteres. Por isso, números e caracteres são tipos de dados diferentes.

Tipos de dados: definição

Tipo de dado é a especificação da forma como o computador armazenará um dado na memória em conjunto com as operações podem ser realizadas com ele. Cada linguagem de programação define os tipos de dados que ela é capaz de manipular.

Exemplos de tipos de dados

- Verdadeiro ou Falso (chamado de dado booleano ou lógico)
- número inteiro
- número real
- número complexo
- caracter

Tipos básicos de Dados da linguagem S

Os tipos de dados da linguagem S mais comumente usados para desenvolvimento de aplicações estatísticas no R são:

- 1 logical;
- 2 integer;
- 3 character;
- 4 double;
- 5 complex;
- 6 list, e
- 7 S4

Objetos

- Os dados do mundo real normalmente são organizados pelas pessoas de alguma forma. Por exemplo, em listas, tabelas etc. Tais formas de organizar os dados são chamadas de **estruturas de dados**.
- As regiões de memória onde se armazenam as estruturas de dados que R manipula são chamadas de **objetos**.
- De modo simplificado, pode-se entender que o R armazena os objetos como listas de tabelas com atributos e seus respectivos valores.
- Os objetos manipulados pelos usuários de R possuem um dos tipos mencionados anteriormente.

Principais objetos de R

- vector, e
- list

Vectors

- O índice inicial de um vetor é igual a 1.
- Podem ter dimensão igual ou superior a 1.
- Todos os elementos de um vetor são do mesmo tipo de dado.
- Para criar um vetor chamado x de três coordenadas contendo os inteiros 1, 2 e 3, escreve-se: $x \leftarrow c(1, 2, 3)$
- Cada termo do vetor x pode ser chamado por $x[i]$.

Numeric Vector

- Vetores numéricos são aqueles cujas coordenadas são todas integer, double ou complex.
- operadores aritméticos - Adição (+), Subtração (-), Multiplicação (*), Potenciação (** ou $\hat{}$), Resto da Divisão (%%)
- As operações aritméticas em um vetor em R são feitas elemento a elemento tal como em uma operação no \mathbb{R}^3 .
- Confira em [Operações aritméticas](#)

Logical Vector

- TRUE, FALSE, NA (not available), Nan (not a number).
- $>$, $>=$, $<$, $<=$, $==$, $!=$
- $c \ \& \ b = c \ e \ b$
- $c|b - c \text{ ou } b$

Character Vector

- Caracteres e cadeias de caracteres (strings) São colocados entre aspas simples ou duplas.
- Caracteres reservados podem ser acionados adicionando uma barra anteriormente, isto é: % .
- barra seguido de n,t ou designa, respectivamente, nova linha, tab e backspace.

Algumas funções muito usadas com vetores

- aritméticas - `sum(x)`, `prod(x)`, `abs(x)`
- ordenação - `sort(x)`, `order()` e `sort.list()` .
- junção de caracteres - `paste()`
- geração de sequencias - `seq()` e `:` - `1:30` ou `seq(from,to,by,length)`
- replicação de objetos - `rep()` -
- `is.na(x)` - testa se o valor de `x` está disponível.
- `mode(x)` - diz o tipo básico de dado (modo) de `x`.

Algumas funções estatísticas muito usadas

- `min(x)`, `max(x)`, `range(x)`, `length(x)`, `mean(x)`, `var(x)`, `hist()`
- `rnorm(x)` - geração de números aleatórios normais
- `data()` - base de dados de teste do R
- `lsfit()` - regressão por mínimos quadrados
- `ls.diag()` - teste de diagnóstico da regressão
- `summary()` - estatísticas descritivas do objeto.
- `arima()` - Metodologia Box e Jenkins

List

- List é uma conjunto ordenado de objetos denominados componentes.
- A lista é como um vetor, mas pode ter componentes de tipos básicos (modos) distintos.
- `lst <- list('casa', 3, c(4,5,3))`
- `lst[1] = casa`
- `lst1 <- list(name = 'Pedro', esposa = 'Maria', filho = 'Carlos')`
- Veja [Concatenação de listas](#)

- Introdução
- Conceitos Fundamentais
- A linguagem S
- O Ambiente Estatístico R
- A estrutura do código-fonte R
- O Modelo de Dados de R
- O Modelo de Execução de R**
- Entrada e saída de dados
- Referências

O Modelo de Execução de R

O Modelo de Execução de R

- O código fonte R é formado pela composição de operações sobre os objetos.
- Tal composição é chamada de expressão.
- Toda expressão possui um valor que é retornado ao usuário.
- O cálculo se encerra ao final da avaliação de todas as expressões.²
- Observe que avaliar expressões compostas é equivalente a calcular o valor de funções compostas.

²Mais detalhes podem ser vistos em [Avaliação de Expressões](#)

O Modelo de Execução de R (cont.)

- Este modelo de execução foi adotado pelos implementadores de R para reproduzir o modelo de execução da linguagem [Scheme](#).
- A vantagem deste modelo de execução é que ele permite a prova matemática da correção do código escrito e, por isso, é especialmente interessante para aplicações científicas.
- A formalização matemática deste modelo de computação é chamada de [cálculo lambda](#) e é devida ao matemático Alonzo Church.

O Modelo de Execução de R (cont.)

- O cálculo lambda se mostra equivalente ao modelo de computação tradicional devido a Alan Turing e desenvolvido por von Neumann.
- Este modelo de computação é o fundamento do paradigma de programação funcional.
- Diz-se que as linguagens que o adotam dão suporte tal paradigma.

Escopo de variáveis, ambientes e funções

As variáveis em R só podem ser manipuladas dentro do ambiente em que são definidas. Isto é chamado de escopo léxico ou estático.

Como R é uma linguagem R funcional, pode-se criar dinamicamente funções que servem para definir escopos de execução, superando a possível limitação que o escopo léxico poderia apresentar.

Função

Uma função em R é um conjunto de expressões colocadas dentro de um ambiente e que podem ser chamadas à execução, retornando valor.

Listagem 1: funções

```
y <- f(x){  
    return x*x  
}
```

Entrada e saída de dados

Funções de Entrada (ou Leitura) de Dados

- Como vimos, R manipula os objetos por meio de funções em virtude de a linguagem S ser eminentemente funcional ou orientada a expressões.
- Porém, por simplicidade, R também disponibiliza atalhos sintáticos não puramente funcionais.
- Por exemplo, a atribuição de variáveis é feita por meio de uma função `assign` a qual, como vimos, utiliza como atalho sintático o símbolo `< -`.

Pacotes de funções

- As funções do ambiente R estão agrupadas em objetos maiores chamados pacotes.
- O pacote básico que compõe a instalação do R é chamado *base*.
- Os usuários do R podem criar suas próprias funções bem como pacotes para manipular seus dados. Isto será visto mais adiante.

Leitura de dados I

- A função `scan()` é a base da leitura dados em R.
- A sintaxe da função `scan()` é dada a seguir:

```
scan(file= "", what= double(), nmax= -1, sep= "")
```


Leitura de dados II

- file é o nome do arquivo a ser lido. O padrão é o teclado.
- what é o tipo de dado a ser lido. O padrão é um vector de qualquer tipo.
- nmax é o número parâmetros a serem lidos. O padrão é o valor -1 que significa terminar com o pressionamento da tecla ENTER.
- sep é o separador entre as entradas de dados. O padrão é espaço.

Leitura de dados: exemplo

A seguir mostramos o uso da função `scan()` para leitura dois números reais diretamente do teclado.

```
scan(what= double(), nmax= 2)
```

Note que, como sempre, quando a leitura é feita seguindo o padrão, não é necessário indicar os parâmetros da função.

Funções de Saída (ou Escrita) de dados

- A função `print()` é a base da escrita de dados em tela no R.
- A sintaxe da função `print()` é dada a seguir:




```
print(x, digits = NULL)
```

Onde:




- `x` é o objeto a ser impresso.
- `digits` é o número de algarismos significativos que devem ser impressos.

Referências

Referências

-  Becker, Richard A. (2015), A Brief History of S, Murray Hill, New Jersey: AT&T Bell Laboratories. Disponível em [link](#).
-  Becker, R.A.; Chambers, J.M. (1978). S: An Interactive Environment for Data Analysis and Graphics. Bell Laboratories
-  Richard A. Becker, John M. Chambers and Allan R. Wilks (1988), The New S Language. Chapman & Hall, New York. Normalmente conhecido como “Blue Book”.

Referências (cont.)

-  John M. Chambers (1998) *Programming with Data*. Springer, New York. Normalmente, chamado de “Green Book”.
-  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](#).