Estruturas de Dados: objetos e suas funções de manipulação

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

Introdução

Introdução

- As pessoas costumam organizar seus dados pára facilitar seu uso. Aos dados organizados chamamos **estrutura de dados**.
- Formas comuns de organizar os dados são: listas, tabelas, árvores (onde se organizam dados a depender da situação que efetivamente vier a ocorrer) etc.
- Como a linguagem S foi projetada para ser uma linguagem funcional, ela conta com estruturas de dados pré-definidas para armazenar dados que são facilmente modeladas como funções.
- Tais estruturas de dados são chamadas ***objeto***.

Objetos

- De modo simplificado, pode-se entender que o GNU R armazena *internamente* os objetos como listas de tabelas com atributos e seus respectivos valores. [1]
- Por exemplo, um objeto vetor poderia ser composto apenas pelos atributos length e class armazenados internamente como $\{values = (1, 3), length = 2; class =' numeric'\}$.

[1] Listas de tabelas são facilmente modeladas usando funções.

Tipos de dados

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 GNU R são:

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

Observações I

- A linguagem S oferece três conceitos de tipos de dados: type, mode e class.
- A especificação da linguagem S chama o tipo básico dos elementos que compõem um objeto é chamado de mode.
- Diferentemente de outras linguagens, a linguagem S trata números inteiros e números reais como um único tipo de entidade e, por isso, pertencem ao mesmo mode chamado numeric. Ou seja, em S, mode e type são equivalentes e o conceito mais usado é mode.

Observações II

1. **No dialeto do R, em oposição à especificação original da linguagem S**, números inteiros e números reais são tratados diferentemente.
2. Por isso, o type de inteiros é integer, o type de reais é double, embora o mode de ambos é numeric tal como prescreve a especificação da linguagem S. Com efeito, double é equivalente a numeric. Por isso, em R, o conceito mais usado é type.

Observações III

1. Class é um atributo de um objeto que lhe permite ser tratado igualmente a outros objetos da mesma classe, isto é, ser capaz de realizar todas as operações que os objetos daquela classe são capazes.
2. Originalmente, a noção de class existe para permitir a escrita de código segundo o paradigma de programação orientada a objetos em S.

Observações IV

1. No paradigma de programação orientada a objetos, objetos possuem atributos e métodos. Atributos são os valores das variáveis que caracterizam o objeto e métodos são funções que o objeto é capaz de executar sobre si mesmo ou objetos da mesma classe de seus atributos.
2. Em S, os dados armazenados nos objetos são seus atributos principais, além de outros atributos internos (como length, mode e class) e os métodos são escritos por meio do mecanismo de funções genéricas.[2]

[2] Mais à frente, estudaremos orientação a objetos em R.

Observações V

- Diz-se que o tipos de dados em R são dinâmicos, isto é, os tipos são definidos no momento em que as operações são efetivamente realizadas (tempo de execução). Por isso, o usuário não define os tipos de dados previamente quando escreve o código.
- Diz-se que os tipos de dados em R são fracos, ou seja, o tipo é convertido automaticamente e implicitamente pelo ambiente para conseguir realizar as operações.
- Tipos dinâmicos e fracos exigem cuidado do programador para não provocar erros de difícil percepção e solução.

Estruturas de dados mais importantes em R

1. vector - é um vetor de tipos básicos do R.
2. list - é uma lista de tipos básicos do R.
3. factor - é uma vetor com classificações de outro vetor em R.
4. matrix - é uma matriz de tipos básicos do R.
5. array - é um arranjo de tipos básicos do R.
6. dataframe - é uma lista de listas em do R.

Vectors

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: $x < -c(1, 2, 3)$
- 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 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

Lists

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

Factors

Factor

- É um vetor usado para classificar dados de outro vetor de mesmo tamanho.
- Cada dado distinto do vetor classificado é chamado de nível.
- Confira detalhes em Fatores

Matrices e Arrays

Matrix

- Definição: Uma matrix é um vetor bidimensional.
- Atribuição:
  $Z <- matrix(data = vector, nrow = integer, ncol = integer)$

Exemplo

$m <- matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9,), nrow = 3, ncol = 3)$

Array

- Array é um vetor de mais de 2 dimensões.
- $Z <- array(data = vector, dim = vector)$

Exemplo

- $n <- array(data = c(1,2,3,4,5,6,7,8,9,10,11,12), dim = c(3,2,2))$
- O resultado será 2 matrizes (que é a terceira coordenada do vetor de dimensões) 3×2.

Dataframes

Dataframe

- É uma estrutura de dados análoga a uma matriz que aceita componentes de modes distintos.
- O dataframe está para a lista assim como a matriz e o arranjo estão para o vetor.
- Basicamente, é uma lista onde cada coluna é um componente.

Dataframe (cont.)

- Cada coluna é uma lista e os diferentes valores de cada lista são chamados de levels.
- O melhor jeito de se utilizar as colunas de um dataframe é usar: nomedodataframe$nomedacoluna.
- Para usar as variáveis nas colunas do dataframe é preciso usar a função attach()

Leitura de arquivos

O ambiente R armazenará o resultado da leitura de arquivos em vectores, lists, dataframes ou qualquer outro de seus objetos. Além da função scan(), as funções abaixo servem para ler arquivos.

- read.table('nomedoarquivo', header=TRUE, sep=';')
- read.csv(file='nomedoarquivo', header=TRUE, sep=';')
- read.csv2(file='nomedoarquivo', header=TRUE, sep=';') - lê formato decimal brasileiro

Exercícios

Exercícios I

1. Faça um programa que peça dois números inteiros e ponha na tela a soma deles.

2. Faça um programa que leia vários valores em metros e o converta-os para milímetros.

3. Faça um programa que pergunte a velocidade do carro de usuário e, no caso desta velocidade ser superior a 80km/h, exiba uma mensagem dizendo que o usuário foi multado. Nesse caso, exiba o valor da multa, cobrando R $ 5,00 por km acima de 80km/h.

Exercícios II

Seja $C = 100 + 0,9Y$ uma função consumo keynesiana e o investimento agregado dado por I = 800-0,7R onde R é a taxa de juros real. Sendo o nível de preços igual $P = 3$, faça um programa que, usando o modelo IS/LM, peça os valores do gasto público e da oferta de moeda e calcule o consumo agregado, o investimento agregado, a taxa de juros de equilíbrio de curto prazo.

Exercícios III

Faça os exercícios de r-exercises a seguir:

1. Criando vetores
2. Sequências Regulares
3. Trabalhando com vetores
4. Vetores e funções
5. Operações Matriciais

Referências

Referências

- BLACK, K. R Tutorial. Department of Mathematics. University of Georgia. Disponível em R Tutorial.
- Matloff, Norman. *The Art of R Programming*. San Francisco, CA: No Starch Press, 2011. Disponível em link.
- R Core Team. R Language Definition. Disponível em R Language Definition.
- R Core Team. An Introduction to R Disponível em An Introduction to R.

Referências (cont.)

- SHORT, T. R Reference Card. Disponível em R Reference Card
- von Wangenheim, Aldo. *Aula de algoritmos*. Florianópolis: INF/UFSC. Disponível em Algoritmo
- Softblue - video que ensina a programar estruturadamente.