Curso de R básico

Sergio Pedro R Oliveira

2022-05-22

Table of Contents

# 1 Objetivo

Estudo dirigido de básico de linguagem R.

# 2 Referência

Videoaulas de ‘curso R para iniciantes’.

# 3 Aula 01 - Introdução

* **R** é uma linguagem estatística e gráfica.
* É uma linguagem com um foco bem definido.
* Muito usada para **Big Data** e **Machine Learning**.
* Linguagem de codigo aberto.
* Muitos pacotes a disposição.
* Ambiente de desenvolvimento: **RStudio**.

# 4 Aula 02 - Instalação R e RStudio

* **R**  
  <https://cran.r-project.org/>
* **RStudio**(IDE)  
  <https://www.rstudio.com/>

# 5 Aula 03 - Conhecendo o R e o RStudio parte 1

## 5.1 Configuração do **layout** do **RStudio**

* Alterando o *layout* do **RStudio**:  
  + opção **Tools**
  + **Global Options**
  + **Pane Layout**  
    Reorganizar o *layout* das janelas do RStudio para melhor se adaptar ao usuário.

## 5.2 Inserir **comentarios**

* Inserir **comentarios**, utilizar ‘**#**’.

## 5.3 Compilar codigo

* Para executar um script:  
  + Deixar o cursor sobre a linha que deve ser executada.
  + Selecionar o codigo que deve ser executado.  
    Obs.: Atalho para executar ‘CRTL + Enter’.

## 5.4 Limpar a tela do console

* Comando que limpa a tela do console:  
  + **CTRL** + **L**

## 5.5 Atribuições

(Tipagem automatica/dinâmica)

* Atribuições de **variaveis** usar o sinal ‘**->**’ ou ‘**<-**’.
* Atribuição de **funções** utilizar o sinal ‘**=**’.

## 5.6 Tipagem de variaveis

* Tipagem das variaveis automatica/dinâmica.
* Erro ao fazer operações com variaveis de tipos distintos.

# 6 Aula 04 - Conhecendo R e o RStudio parte 2

## 6.1 **Packages** - Bibliotecas/Pacotes

* São bibliotecas/pacotes com funções que podem ser baixados e instalados.
* Os pacotes/bibliotecas são instalados no sistema.
* As bibliotecas são baixadas com o comando no **R**:  
  + Sintaxe:  
    **install**.**packages** (“*nome\_do\_pacote*”)
* Para usar os pacotes é preciso declarar eles no script (chamar o pacote):  
  + Sintaxe:  
    **library**(“*nome\_do\_pacote*”)
* Baixado e declarado o pacote/biblioteca no script é possivel agora usar as funções desse pacote.

## 6.2 Pesquisa de função - função Help

* Help ‘**?**’, usado para pesquisar uma função na documentação do **R**.
* O resultado da pesquisa aparece no *Help*.
* Sintaxe:  
  **?***c*
* Caso o *Help* não ajude a descobrir a função, outro modo de pesquisar é recorrer ao site:  
  <https://www.rdocumentation.org/>  
  + Um site destinado a pesquisa e informações sobre a documentação do **R**.
  + Retorna informações sobre a função, estrutura, pacote a qual ela faz parte, etc.

## 6.3 **summary** - Resumo dos dados

* A função **summary**() retorna o resumo de variaveis.
* O retorno depende do argumento (se for um vetor, uma lista, um data.table).
* O retorno da função, no geral, retorna diversos metodos aplicados aos dados, tais como:  
  + valor mínimo
  + 1º quantil
  + valor da mediana
  + valor da media
  + 3º quantil
  + valor máximo
* Sintaxe:  
  **summary**(*variavel*)

## 6.4 Descobrir a classe de uma variavel - **class**

* A função “**class**()” retorna a que classe do objeto do argemunto pertence.
* Basicamente diz se o objeto é numerico, string, vetor, lista, data.frame, matriz, …
* Sintaxe:  
  **class**(*argumento*)

## 6.5 Descobrir a estrutura de uma variavel - **str**

* A função “**str**()” retorna a estrutura do objeto do argumento.
* Retorna diversos dados, entre eles:  
  + A classe do objeto.
  + Tamanho do objeto.
  + A lista, ou vertor, dos campos com o tipo e tamanho.
* Sintaxe:  
  **str**(*argumento*)

## 6.6 Vetor

* Vetores são variaves que aguardam diversos valores de mesmo tipo.
* No **R** o vetor é criado a partir da função “*c*()”.
* *Strings* e *caracteres* dentro do vetor devem ter seus valores entre aspas (““).
* Sintaxe (exemplo):  
  **c**(10,5,8,…)

## 6.7 Lista

* Listas sao os objetos R que contem elementos de diferentes tipos (diferente do vetor), como \_ numeros, strings, vetores e outra lista dentro dela.
* Uma lista tambem pode conter uma matriz ou uma funcao como seus elementos.
* *Strings* e *caracteres* dentro da lista devem ter seus valores entre aspas (““).
* A lista e criada usando a funcao “*list*()”.
* Sintaxe (exemplo):  
  **list**(azul, 10, c(5,8,9), …)

# 7 Aula 05 - Operadores

## 7.1 Operadores básicos

## Operador Simbolo  
## 1 Soma +  
## 2 Subtração -  
## 3 Divisão /  
## 4 Multiplicação \*  
## 5 Potencia \*\* ou ^  
## 6 Raiz sqrt(numero)  
## 7 Exponencial exp(numero)  
## 8 Log na base n log(numero,n)

## 7.2 Operadores lógicos

## Operador\_logico Simbolos  
## 1 Igual ==  
## 2 Diferente !=  
## 3 Maior que >  
## 4 Maior ou igual >=  
## 5 Menor que <  
## 6 Menor ou igual <=  
## 7 E &  
## 8 OU |  
## 9 Negação !

* A resposta do sistema para uma operação logica é *TRUE* (verdadeiro) ou *FALSE* (falso). Ou em termos numericos 1 (verdadeiro), 0 (falso).
* No caso da negação (NOT), ele vem antes da operação como um todo.  
  + Exemplo:

# 8 Aula 06 á 09 - Tipo de dados

## 8.1 Tipos de dados

### 8.1.1 Numerico - **numeric**

* O **R** converte automaticamente variaveis númericas para tipo numeric.
* São variaveis contando números tanto inteiros (*int*), quanto *float*.
* Variaveis do tipo numeric podem realizar operações matemáticas.
* É possivel converter um número qualquer (*numeric*) para um número inteiro (*int*) utilizando a função uma função:  
  + **as**.**integer**(*variavel*)
* Observações:
  + A função “**as**.**interger**(*variavel*)” faz truncamente. CUIDADO!!!
  + Para fazer arredondamento, usar a função “**round**(*variavel*, *qtd\_decimal*)”.

### 8.1.2 Caractere e String - **character**

* O **R** entende como variaveis do tipo *character* (caracter/string) todo dado que tiver entre aspas (““).
* Mesmo números se tiverem entre aspas, serão lidos como *character*.
* Tipo *character* não faz operações matemáticas.
* Tipo *character* faz operações lógicas (igual á, diferente de, …)  
  + Espaços em branco fazem diferença para operações lógicas.

### 8.1.3 Fatores - **factor**

* Factor é o tipo categoria/enumerado.
* O tipo **factor** não se trata de numeros, mas sim de categorias classificar determinado registro.
* O tipo **factor** não se comporta como numeros, logo não é possivel fazer operações matematicas nele. Porem aceita operações lógicas.
* Pacara tipar uma variavel (ou determinado vetor) como **factor** basta usar a função **as**.**factor**(*argumento*).
* No caso da conversão de vetores para tipo factor dos elementos:
  + Elementos iguais serão considerados “niveis”.
  + A função **summary**(*vetor*), numera o numero de elementos no mesmo “nivel”.

### 8.1.4 Lógico

* É possivel salvar dentro de uma variavel uma operação do tipo lógica.
* O que é salvo é o resultado da operação, os valores “*TRUE*” ou “*FALSE*”.
* O tipo lógico são palavras reservadas *TRUE* e *FALSE*. Se utiliza-las dentro de aspas elas se convertem em tipo character (“*TRUE*” e “*FALSE*”).
* Se converter um tipo lógico para numerico *TRUE* assume o valor 1 e *FALSE* o valor 0.
* Sintaxe (exemplo):  
  *L* <- *variavel\_1* < *variavel\_2*

## 8.2 Descobrir e converter tipos

### 8.2.1 as.tipo

* As funções começadas por “**as**.”, seguidas do “*tipo*” e a “*variavel*” como argumento, servem para converter variaveis para outros tipos.
* Sintaxe (exemplo):  
  **as**.**factor**(*variavel*)  
  + converte a variavel de um tipo qualquer para uma variavel do tipo factor.

### 8.2.2 is.tipo

* As funções começadas por “**is**.”, seguidas por “*tipo*” e a “*variavel*” como argumento, servem para descobrir/confirmar o tipo da variavel.
* O retorno é “*TRUE*” ou “*FALSE*”.
* Sintaxe (exemplo):  
  **is**.**factor**(*variavel*)  
  + Retorna “*TRUE*” se verdadeiro, o tipo da variavel for factor.
  + Ou retorna “*FALSE*” se falso, o tipo da variavel não for factor.

# 9 Aula 10 -

# 10 Aula 11 -

# 11 Aula 12 -

# 12 Aula 13 -

# 13 Aula 14 -

# 14 Aula 15 -

# 15 Aula 16 -

# 16 Aula 17 -

# 17 Andamento dos Estudos

## 17.1 Assunto em andamento

Atualmente estou escrevendo Aula 10.