

Curso de R básico

Sergio Pedro R Oliveira

2022-05-23

Contents

1	Objetivo	2
2	Referência	2
3	Aula 01 - Introdução	3
4	Aula 02 - Instalação R e RStudio	4
5	Aula 03 - Conhecendo o R e o RStudio parte 1	5
5.1	Configuração do layout do RStudio	5
5.2	Inserir comentarios	5
5.3	Compilar código	5
5.4	Limpar a tela do console	5
5.5	Atribuições	5
5.6	Tipagem de variáveis	5
6	Aula 04 - Conhecendo R e o RStudio parte 2	6
6.1	Packages - Bibliotecas/Pacotes	6
6.2	Pesquisa de função - função Help	7
6.3	summary - Resumo dos dados	7
6.4	Descobrir a classe de uma variável - class	7
6.5	Descobrir a estrutura de uma variável - str	7
7	Aula 05 - Operadores	9
7.1	Operadores básicos	9
7.2	Operadores lógicos	9
8	Aula 06 á 09 - Tipo de dados	10
8.1	Tipos de dados	10
8.2	Descobrir e converter tipos	12
9	Aula 10 e 11 - Vetores e Listas	13
9.1	Vetor	13
9.2	Lista	13
10	Aula 12 - Matrizes e Data.frames	14
11	Aula 13 -	15
12	Aula 14 -	16
13	Aula 15 -	17

14 Aula 16 -	18
15 Aula 17 -	19
16 Andamento dos Estudos	20
16.1 Assunto em andamento	20

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 código 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 ‘CTRL + 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 possível 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 variáveis.
- 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 vetor, dos campos com o tipo e tamanho.
- Sintaxe:
str(*argumento*)

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:
!5 > 4

8 Aula 06 á 09 - Tipo de dados

8.1 Tipos de dados

8.1.1 Numérico - `numeric`

- O **R** converte automaticamente variáveis numéricas para tipo `numeric`.
- São variáveis contando números tanto inteiros (*int*), quanto *float*.
- Variáveis do tipo `numeric` podem realizar operações matemáticas.
- É possível converter um número qualquer (*numeric*) para um número inteiro (*int*) utilizando a função uma função:

- `as.integer(variavel)`

- Observações:
 - A função “`as.integer(variavel)`” faz truncamento. CUIDADO!!!
 - Para fazer arredondamento, usar a função “`round(variavel, qtd_decimal)`”.

8.1.2 Caractere e String - `character`

- O **R** entende como variáveis 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 números, mas sim de categorias classificar determinado registro.
- O tipo **factor** não se comporta como números, logo não é possível fazer operações matemáticas nele. Porém aceita operações lógicas.
- Para tipar uma variável (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 “níveis”.
 - A função `summary(vetor)`, numera o número de elementos no mesmo “nível”.

8.1.4 Lógico

- É possível salvar dentro de uma variável 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 numérico *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 variáveis para outros tipos.
- Sintaxe (exemplo):
as.factor(*variavel*)
 - converte a variável de um tipo qualquer para uma variável 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 variável.
- O retorno é “*TRUE*” ou “*FALSE*”.
- Sintaxe (exemplo):
is.factor(*variavel*)
 - Retorna “*TRUE*” se verdadeiro, o tipo da variável for factor.
 - Ou retorna “*FALSE*” se falso, o tipo da variável não for factor.

9 Aula 10 e 11 - Vetores e Listas

9.1 Vetor

- Vetores são variáveis 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 (“”).
- *length* indica o tamanho do vetor, o número de elementos dentro do vetor.
- Caso seja inserido um dado do tipo *character* para um vetor do tipo *numérico*, ele converte todo o vetor em tipo *character* (todos os elementos em *character*).
- Sintaxe (exemplo):
`c(10,5,8,...)`
- Acessando um valor dentro de um vetor, uso a notação:
`nome_do_vetor [posição]`

9.2 Lista

- Listas são os objetos R que contêm elementos de diferentes tipos (diferente do vetor), como `_` números, strings, vetores e outra lista dentro dela.
- Uma lista também pode conter uma matriz ou uma função como seus elementos.
- *Strings* e *caracteres* dentro da lista devem ter seus valores entre aspas (“”).
- A lista é criada usando a função “*list()*”.
- Sintaxe (exemplo):
`list(azul, 10, c(5,8,9), ...)`
- Acessando um valor dentro de uma lista.
 - Notação:
`nome_da_lista [[posição]]`
 - Caso tenha uma lista ou vetor dentro de algum elemento:
`nome_da_lista [[posição]][posição_dentro_da_lista_ou_vetor_do_elemento]`

10 Aula 12 - Matrizes e Data.frames

11 Aula 13 -

12 Aula 14 -

13 Aula 15 -

14 Aula 16 -

15 Aula 17 -

16 Andamento dos Estudos

16.1 Assunto em andamento

Atualmente estou escrevendo Aula 11.