

R e RStudio para Iniciantes

**Material de Apoio para Cursos Quantitativos do Instituto de Economia da
Universidade Federal do Rio de Janeiro (IE/UFRJ)**

GPEQ/UFRJ

2024-04-11

Índice

1	Objects	3
1.0.1	Tipo & Forma	3
2	Primeiros passos	5
2.1	Operadores Aritméticos	5
2.2	Operadores Lógicos	6
2.3	Possíveis complicações	7

1 Objects

[...]

1.0.1 Tipo & Forma

Vamos nos aprofundar um pouco mais. Ao lidar formalmente com dados, **devemos ter mente que eles são compostos por uma ou mais variáveis e seus valores**. Uma *variável* é uma *dimensão ou propriedade que descreve uma unidade de observação* (por exemplo, uma pessoa) e normalmente pode assumir valores diferentes. Por outro lado, os *valores* são as *instâncias concretas que uma variável atribui a cada unidade de observação e são ainda caracterizados por seu intervalo* (por exemplo, valores categóricos versus valores contínuos) e seu *tipo* (por exemplo, valores lógicos, numéricos ou de caracteres). Estaremos interessados no *tipo* dos dados. A Tabela 1.1 apresenta os que podem aparecer com maior frequência.

Tabela 1.1: Tipos mais comuns de dados

Tipo	Serve para representar...	Exemplo
Númerico	números do tipo <i>integer</i> (inteiro) ou <i>double</i> (reais)	1, 3.2, 0.89
Texto (<i>string</i>)	caracteres (letras, palavras ou setenças)	“Ana jogou bola”
Lógico	valores verdade do tipo lógico (valores booleanos)	TRUE, FALSE, NA
Tempo	datas e horas	14/04/1999

Voltando ao primeiro exemplo, uma pessoa pode ser descrita pelas variáveis *nome*, *número de horas dormidas* e *se dormiu ou não mais de oito horas*. Os valores correspondentes a essas variáveis seriam do tipo texto (por exemplo, “Pedro”), numéricos (número de horas) e lógicos (TRUE ou FALSE, definido em função do tempo descansado¹). **Note a diferença entre dado e valor**. O número 10 é um valor, sem significado. Por outro lado, “10 horas dormidas” é um dado, caracterizado pelo valor 10 e pela variável “horas dormidas”.

Outro aspecto importante sobre os dados está em sua forma, ou seja, como os dados podem ser organizados. A Tabela 1.2 apresenta as formas mais comuns de organização.

¹Se o número de horas que a pessoa descansou for maior do que 8, então a variável deverá apresentar valor igual a TRUE – ou seja, é verdade que a pessoa dormiu mais de 8 horas. Caso contrário, FALSE.

Tabela 1.2: Formas pelas quais os dados podem ser organizados

Formato	Os dados se apresentam como...	Exemplo
Escalar	elementos individuais	“AB”, 4, TRUE
Retangular	dados organizados em i linhas e j colunas	Vetores e Tabelas de Dados
Não-retangular	junção de uma ou mais estruturas de dados	Listas

[...]

2 Primeiros passos

Partes deste capítulo são baseadas na seção 3.2 ‘R como calculadora’ do livro *Ciência de Dados em R*, feito pelo Curso-R. De qualquer modo, eventuais erros são inteiramente de nossa responsabilidade.

Como vimos nos capítulos anteriores, o papel do **Console** no R é interpretar os nossos comandos à luz da linguagem. Ele avalia o código que o passamos e devolve a saída correspondente — se tudo der certo — ou uma mensagem de erro — se o seu código tiver algum problema. Essa operação é chamada de **avaliar**, **executar** ou **rodar** o código. Para que seu código seja executado diretamente no Console, escreva-o e, na sequência, aperte **Enter**. A outra forma de executar uma expressão é escrever o código em um *script* no **Editor**, deixar o cursor em cima da linha e usar o atalho **Ctrl + Enter**. Assim, o comando é enviado para o Console, onde é diretamente executado.

Nesse capítulo, você *rodará* suas primeiras linhas de código com intuito de realizar operações aritméticas como *adição*, *subtração*, *multiplicação* e *divisão*, além de comparações lógicas simples. O objetivo aqui não é te ensinar matemática básica, mas te preparar para a execução de linhas de código mais avançadas. É a forma mais fácil de um iniciante ganhar familiaridade e experiência prática com o R.

2.1 Operadores Aritméticos

De agora em diante, cada região sombreada de cinza representa código, ao passo que seu resultado estará exposto logo na sequência. Vamos começar com um exemplo simples:

```
1 + 1
```

```
[1] 2
```

Nesse caso, o nosso comando foi o código `1 + 1` e a saída foi o valor 2. Como você pode reproduzir esse comando no RStudio? Inicialmente, copie o que está escrito acima ao clicar no símbolo de prancheta no canto superior direito da região sombreada. Na sequência, cole no Editor de Código e aperte **Ctrl + Enter** (ou então no Console, pressionando apenas **Enter**). Observe abaixo!

Tente agora jogar no Console a expressão: $2 * 2 - (4 + 4) / 2$. Deu zero? Pronto! Você já é capaz de pedir ao R para fazer *qualquer uma das quatro operações aritméticas básicas*. Repare que as operações e suas precedências são mantidas como na matemática, ou seja, divisão e multiplicação são calculadas antes da adição e subtração, além de os parênteses ditarem a ordem na qual serão realizadas. A seguir, apresentamos a Tabela 2.1 resumindo como fazer as principais operações no R.

Tabela 2.1: Operadores matemáticos do R

Operação	Operador	Exemplo	Resultado
Adição	+	$1 + 1$	2.00
Subtração	-	$4 - 2$	2.00
Multiplicação	*	$2 * 3$	6.00
Divisão	/	$5 / 3$	1.67
Potenciação	^	$4 ^ 2$	16.00
Resto da Divisão	%%	$5 \% \% 3$	2.00
Parte Inteira da Divisão	%%/%	$5 \% / \% 3$	1.00

2.2 Operadores Lógicos

O R permite também testar comparações lógicas. Os valores lógicos básicos em R são `TRUE` (ou apenas `T`) e `FALSE` (ou apenas `F`). Por exemplo, podemos pedir ao R que nos diga se é verdadeiro que 5 é menor do que 3. Como a resposta é obviamente negativa, ele retornará `FALSE`, nos dizendo que a proposição que fizemos é falsa.

```
5 < 3
```

```
[1] FALSE
```

Abaixo, introduzimos a Tabela 2.2 com outros operadores lógicos da linguagem.

Tabela 2.2: Operadores lógicos do R

Operação	Operador	Exemplo	Resultado
Maior que	>	$2 > 1$	TRUE
Maior ou igual que	>=	$2 >= 2$	TRUE
Menor que	<	$2 < 3$	TRUE

Tabela 2.2: Operadores lógicos do R

Operação	Operador	Exemplo	Resultado
Menor ou igual que	<code><=</code>	<code>5 <= 3</code>	FALSE
Igual à	<code>==</code>	<code>4 == 4</code>	TRUE
Diferente de	<code>!=</code>	<code>5 != 3</code>	TRUE
<code>x e y</code>	<code>&</code>	<code>x <- c(1, 4, NA, 8) x[!is.na(x) & x > 5]</code>	8
<code>x ou y</code>	<code> </code>	<code>x <- c(1, 4, NA, 8) x[!is.na(x) x > 5]</code>	1, 4, 8

2.3 Possíveis complicações

Se você digitar um comando incompleto, como `5 +`, e apertar **Enter**, o R mostrará um `+`, o que não tem nada a ver com a adição da matemática. Isso significa que o R está esperando você enviar **mais** algum código para completar o seu comando. Termine o seu comando ou aperte **Esc** para recomeçar.

```
5 -
+
+ 5
```

```
[1] 0
```

Se você digitar um comando que o R não reconhece, ele retornará uma mensagem de erro. **Não entre em pânico.** Ele só está te avisando que não conseguiu interpretar o comando.

```
5 % 2
```

```
Error: <text>:1:3: unexpected input
1: 5 % 2
   ^
```

Você pode digitar outro comando normalmente em seguida.

```
5 ^ 2
```

```
[1] 25
```