

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

2024-03-27

Índice

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 ?? 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 ?? 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 Objetos

Na apresentação dessa parte do material, trouxemos uma citação que, em parte, dizia:

Everything that exists is an object.

Um objeto é simplesmente um nome que guarda um valor ou código. Não há como ser mais direto: tudo que existe no R é um *objeto*, inclusive as funções que veremos no capítulo seguinte. Nesse capítulo, veremos com detalhe os objetos que são designados à armazenar dados. Antes, no entanto, vamos dar um passo para trás e explicar o que são dados.

2.1 Dados

Segundo a Oxford Languages, dados são

fatos e estatísticas coletadas de forma conjunta para referência ou análise.

Na prática, dados nos mostram informações sobre determinado indivíduo ou situação que procuramos descrever, seja uma pessoa, instituição, comportamento, condição geográfica, etc. O número de horas que você dormiu essa noite é um dado. A lista que relata quem é ou não calvo na sua família é uma *coleção* de dados. A expectativa, hoje, de quanto será a inflação acumulada nos próximos 12 meses é um dado. A variação percentual do Produto Interno Bruto (PIB) real no último trimestre é um dado. A lista que mostra a sequência de variações do PIB real nos últimos dez trimestres é uma *série temporal*, isto é, dados em sequência ao longo do tempo.

2.1.1 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 ?? apresenta os que podem aparecer com maior frequência.

Tabela 2.1: 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 ?? apresenta as formas mais comuns de organização.

Tabela 2.2: 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

Um escalar é um elemento único, que pode ser de qualquer tipo. Ou seja, a representação elementar de um dado se dá através de um escalar! Por exemplo, o tipo sanguíneo de determinada pessoa, representado pelos caracteres “AB”, é um escalar do tipo texto. Você pode pensar no escalar como um dado organizado em 1 linha e 1 coluna.

Por sua vez, dados retangulares são àqueles cuja organização ocorre em i linhas e j colunas, tal que $i, j \in \mathbb{N}$ e $i > 1$ ou $j > 1$. As formas retangulares mais comuns são *vetores*, *matrizes* e *tabelas de dados*. Uma matriz é uma forma de organização de dados *numéricos* em em i linhas

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.

e j colunas. Quando uma matriz possui i linhas e 1 coluna *ou* 1 linha e j colunas, chamamos de vetor-coluna e vetor-linha, respectivamente; em muitos casos, chamamos apenas de *vetor*. Assim, o vetor é um caso especial de matriz unidimensional. As tabelas de dados, por outro lado, possuem i linhas e j colunas, tal que $i > 1$ e $j > 1$. Além disso, aceitam todos os tipos de dado – por exemplo, numéricos, de textos ou lógicos – em qualquer que seja a combinação de linha e coluna.

Por sua vez, dados não-retangulares se referem a toda organização de dados que não seja feita em linhas e colunas relacionadas entre si. A forma mais comum é a lista. Observe um exemplo abaixo: cada característica pode ser entendida como um elemento de uma lista. Apesar de pertencerem a mesma estrutura, os elementos não se comunicam entre si.

Gênero	Jorge Masculino	Laís Feminino	Matheus Masculino	Laura Feminino	Nathália Feminino
Idade	Jorge 18	Laís 23	Matheus 22	Laura 21	Nathália 21
Altura (cm)	Jorge 180	Laís 170	Matheus 170	Laura 175	Nathália 168
Peso (kg)	Jorge 76	Laís 65	Matheus 70	Laura 68	Nathália 66

Nesse caso em específico, conseguimos fazer a transição para uma tabela (forma retangular) pois todos os elementos são características das mesmas pessoas. Em uma tabela de dados, automaticamente temos uma relação entre os dados: cada linha contém características de uma unidade específica.

Tabela 2.4

Nome	Gênero	Idade	Altura (cm)	Peso (kg)
Jorge	Masculino	18	180	76
Laís	Feminino	23	170	65
Matheus	Masculino	22	175	70
Laura	Feminino	21	181	68
Nathália	Feminino	21	168	66

2.2 Estruturas de Dados no R

Na seção anterior, vimos os conceitos de *tipo* e *forma*. Tenha em mente que são duas definições que existem independentemente de qualquer linguagem de programação – elas versam sobre *dados* de forma geral.

Por outro lado, agora veremos o conceito e alguns exemplos de *estrutura de dados* para o R. **A estrutura de dados é a forma pela qual o R classificará um objeto em relação ao tipo e a forma dos dados que contém.** Existe uma estrutura de dados para cada combinação de tipo e forma? **Não.** Compreender as principais estruturas disponíveis no R requer vê-las como uma combinação de

- (a) *algum* formato de dados
- (b) o fato de conterem um único ou vários tipos de dados

2.2.1 Criando e armazenando objetos na memória

Antes de conhecê-las, no entanto, vamos entender melhor os comandos para criar e armazenar *qualquer* objeto (seja ele para armazenar dados, como nesse capítulo, ou para criar funções, que serão vistas no próximo) na memória do R.

Para *criar* e *armazenar* um objeto, sempre escreveremos inicialmente seu nome (escolhido por você), seguido de um dos *operadores de atribuição* (ou *assignment operators*, como são conhecidos) e, por fim, o objeto propriamente dito com as informações de nosso interesse. O principal operador de atribuição para se criar objetos é `<-`. Outro operador que é comumente utilizado para cumprir a mesma tarefa é `=`. Ainda que exista uma leve diferença entre ambos, ao longo dos cursos será possível utilizar o operador de sua preferência. Por ser o ideal, utilizaremos `<-` no restante do material.

```
nome_do_objeto <- >objeto com informações<
nome do objeto = >objeto com informações<
```

No parágrafo anterior, observe que está escrito ‘*criar e armazenar*’. Nós poderíamos simplesmente criar um objeto, sem armazená-lo na memória do R. Nesse caso, não teríamos o nome do objeto disponível na aba **Environment** (ou seja, ele não seria armazenado no nosso ambiente de trabalho) e seria bem mais complicado registrar todas as mudanças que viermos a fazer nele. Aconteceria apenas a ocorrência de uma única saída no Console com a estrutura do objeto criado (de forma semelhante ao que fizemos no capítulo anterior) – o quê não tem grande utilidade para nós, exceto caso você queira verificar a estrutura do objeto antes de realmente armazená-lo.