Enap

Aula 2 - Interagindo com o R

Marcelo Prudente e Rafael Giacomin

8 de março de 2018

Enap

Estrutura de Dados

Introdução

Enap

- As estruturas de dados no **R** podem ser diferenciadas por sua dimensão e pelo seu conteúdo (heterogêneo ou homogêneo)

Dimensões	Homogêneo	Heterogêneo
1	Atomic Vector	List
2	Matrix	Data Frame
Nd	Array	

Vetores Atômicos

Enap

- Os vetores atômicos são as estruturas mais simples de dados, com uma dimensão e um tipo.
- Há três tipos de armazenamento de objetos (os vetores assumem um desses tipos):
  - Numérico - double (contínua) ou integer (discreto)
  - Lógicos
  - Texto

Exemplo 1: criando vetores

Enap

- Os vetores são criados por meio dos sinais de atribuição <- ou =. Semprem assumem um tipo de armazenamento.
- Podemos criar vetores por meio da função **combine**, *c(arg1, arg2, . . . , arg_n)*.

```
# Numérico contínuo
num_dbl <- c(1, 2.8, 4.5, 6.3)
# Numérico discreto
num_int = c(1L, 3L, 5L, 7L)
# Lógico
log_vt <- c(TRUE, FALSE, TRUE, FALSE)
# Texto
chr_vt <- c("Jõao", "Zezinha", "Itamar", "Cristiane" )
```

Exemplo 1: criando vetores

Enap

- **Atenção**:
  - os valores decimais são separados por **.**
  - o uso do **L** ao lado dos números os torna discretos
  - o uso das aspas **“”** marca a criação de textos
  - os valores lógicos **TRUE** ou **FALSE** são escritos em maiúsculo. Numericamente, correspondem a 1 e 0.

Exemplo 1: testando a natureza dos vetores

Enap

- Podemos testar se os vetores assumem os tipos esperados:

```
length(num_dbl)
is.vector(num_dbl)
is.numeric(num_dbl)
is.integer(num_dbl)
is.atomic(num_dbl)
is.double(num_dbl)
```

- Vamos testar a natureza dos outros vetores criados?

Exercício 1

Enap

Exercício

- Crie seus próprios vetores:
  - numérico, contínuos e discretos
  - lógico
  - character

factor

Enap

- Os fatores são vetores gravados como números inteiros para representar variáveis categórigas.
  - podem ser variáveis contínuas transformadas em categorias
  - ou variáveis de texto transformadas em categorias.
- Para tanto, utiliza-se o comando as.factor()

```
letras = c("a", "b", "a", "c", "c", "d", "e", "e", "e", "e")
as.factor(letras)
```

```
##  [1] a b a c c d e e e e
## Levels: a b c d e
```

- As categorias tem muita utilidade nos modelos estatísticos e na criação de gráficos

Comandos matemáticos no R

Enap

- O **R** opera como uma calculadora.

Símbolo	Operação
+	Soma
-	Subtração
/	Divisão
*	Multiplicação
^	Potência
sqrt	Raiz Quadrada
log	Logarítmo
sum	Soma
max	Máximo
min	Mínimo
mean	Média

Exercício 2: operações matemáticas

Enap

Exercício:

- Realize algumas operações matemáticas entre números e vetores.
  - Utilize as operações descritas na tabela.
- Crie um vetor numérico e obtenha algumas estatísticas.
- É póssível realizar operações matemáticas com vetores de texto? E com vetores lógicos?

Operadores Lógicos

Enap

- As operações lógicas são cruciais para a manipulação dos dados.
- É possível utilizar o seguinte conjunto de operadores lógicos:

Símbolo	Descrição
<	Menor
<=	Menor ou igual a
>	Maior
>=	Maior ou igual a
==	Exatamente Igual
!=	Não é igual
x | y	x ou y
x & y	x e y

Exercício 3: operações lógicas

Enap

Exercício:

- Suponha que x representa o mês de nascimento de um indivíduo.

```
x<- c(1, 3, 1, 4, 5, 4, 8, 10, 11, 9, 7, 9)
mes_inicial <- c(1, 1, 1, 2, 6, 6, 6, 8, 8, 8, 8, 8)
mes_final <- c(12, 12, 12, 12, 10, 10, 10, 10, 9, 9, 9, 9)
```

- *Verifique*:
  - quantos casos de x são iguais ao mês inicial e quantos ao final?
  - em quais casos o valor de x é maior que o mês incial e final?
  - em quais casos não são iguais?

Operações com Conjuntos

Enap

- Ainda, é possível realizar operações com conjuntos

Símbolo	Descrição
union(x, y)	União
intersect(x, y)	Intercessão
setdiff(x, y)	Diferença
setequal(x, y)	Igualdade
is.element(el, set)	É elemento
%in%	Pertence

Exercício 4: operações com conjuntos

Enap

Dado:

```
x <- 1:15
y <- seq(5, 55, 2.5)
z <- 12:27
```

Determine:

- quais elementos comuns entre x e y?
- qual a união entre y e z?
- qual a intercessão entre x e z?

Sequências e repetições

Enap

- Uma forma importante de criar dados é por meio de sequências e repetições. Além do já conhecido **: (ex., 1:10)**, há funções que refinam essa lógica:

```
# Criando outras sequências
seq(from = 1, to = 10, by = 2)
# De modo mais breve
seq(0, 100, 0.333)

# Você pode especificar um valor
n <- 1000
q <- 2.5
seq(1, n, q)
```

Sequências e repetições

Enap

```r
# Repetindo argumentos
rep("Olá", 2)

# Você pode criar um vetor com nomes
frutas <- c("laranja", "banana", "manga", "maracujá",
            "mamão", "ameixa")
# Repetir frutas 15 vezes
rep(frutas, 15)
# Repetir cada fruta duas vezes
rep(frutas, each = 2)

# Repetir cada argumento de modo distinto
rep(frutas, c(1, 2, 3, 4, 5, 6))
rep(frutas, 1:6) # mesma coisa
rep(frutas, rep(c(2,3), 3)) # mesma coisa
```

Matrizes

Matrizes

Enap

- As matrizes se diferencia dos vetores atômicos por ter o atributo dimensão **dim()**.

Tente executar os comandos abaixo. O que você observa?

```
# Matriz
matriz_a<- matrix(1:12, nrow = 4, ncol = 3, )

# Vetor transformado em matriz (adicionar dimensão)
vetor_b<- 1:12
dim(vetor_b)<- c(4, 3)
```

Matrizes

Enap

- Como a matriz tem duas dimensões, o comando **length()** se generaliza em:
  - **ncol()** - número de colunas
  - **nrow()** - número de linhas
- Do mesmo modo, o comando **names()** se generaliza em:
  - **rownames()**
  - **colnames()**

Por exemplo:

```
rownames(matriz_a) <- c("linha_1", "linha_2", "linha_3",
                        "linha_4")
colnames(matriz_a) <- c("coluna_A", "coluna_B", "coluna_c")
```

Listas

Listas

Enap

- As listas se diferenciam dos vetores atômicos, pois seus elementos podem ser de qualquer tipo, inclusive listas ou matrizes.
- Para criar lista, utiliza-se o comando **list()**

```
lista<- list(x = 1:5,
             "Curso de R",
             c(TRUE, FALSE),
             c(1.5, 1.6, 1.7, 1.8),
             matriz = matrix(1:12, 3, 4),
             lista2 =list(letters, LETTERS))
str(lista)
```

Data Frame

Data Frame

Enap

- Forma mais comum de encontrar dados no **R**.
- É um tipo de lista com vetores de comprimento iguais e duas dimensões, do mesmo modo que as matrizes.

```
# Criando um data.frame na mão
df <- data.frame(nome =c("Josias", "Joaldo", "Josefa",
                         "Josie", "Josimar"),
                 idade = c(42, 28, 34, 27, 55),
                 tem_filhos= c(TRUE, FALSE, FALSE,
                               TRUE, TRUE))
print(df)
```

Data Frame

Enap

```r
# Estrututa do data.frame
str(df)

# Nomes das colunas
names(df); colnames(df)

# Qual o número de linhas?
nrow(df)

# Número de colunas
ncols(df)
```

Data Frame: acessando elementos com o $

Enap

- É possível acessar os elementos do data.frame por meio do **$**.

```
df$idade
```

- Com isso, é fácil extrair a média da coluna idade

```
mean(df$idade)
```

Data Frame: acessando seus elementos com os []

Enap

- Outra forma de acessar os elementos dos objetos (listas, vetores, data.frames ou matrizes), é por meio dos **[]**.
  - Para o uso **[,]**, o lado direito da vírgula é linha e o esquerdo coluna.

```
df[ , ] # todo data.frame
df[1, ] # linha 1
df[ , 2] # coluna 2
df[ 1, 2]
```

Data Frame: adicionando elementos

Enap

- Do mesmo modo que é possível extrair informações de um data.frame, é possível adicionar.
- É possível, então, criar novas colunas:

```
df$qtd_filhos <- c(3, 4, 1, 3, 6)
df[, 5] <- 5
df[, "seis"] <- 6
```

- O uso do **$** extrai ou inclui exatamente o nome da variável.
- Já os **[]** extraem ou incluim a coluna a que se referem.

Do Data.Frame para Tibble

Tibble: um data.frame melhorado

Enap

- O tibble apresenta um formato moderno para o data.frame.
- É parte do *Tidyverse*.
- Traz as seguintes vantagens:
  - Facilita a visualisação dos dados
  - Facilita a criação dos dados
  - Facilita a reciclagem de vetores (?!)

Tibble: visualisando os dados

Enap

- É possível compelir um data.frame comum à classe tibble.

```
# Qual a classe atual do "df"
class(df)
# Transformar "df" em tibble
library(tidyverse)
# Compelir o data.frame para tibble
df <- as_tibble(df)
df
```

Tibble: criando dados reciclando vetores

Enap

- É possível criar dados com o tiblle de forma mais fácil:

```
# Tente criar esse data frame
df <- data.frame(nomes = c("João", "Pedro", "Maria", "Érica"),
                 idade = c(18, 25, 49, 38),
                 idade_q = idade^2,
                 idade_media = mean(idade))
```

Tibble: criando dados reciclando vetores

Enap

- É possível criar dados com o tiblle de forma mais fácil:

```
# Tente criar esse data frame
df <- data.frame(nomes = c("João", "Pedro", "Maria", "Érica"),
                 idade = c(18, 25, 49, 38),
                 idade_q = idade^2,
                 idade_media = mean(idade))

# Com o tibble - compare
dfti <- tibble(nomes = c("João", "Pedro", "Maria", "Érica"),
                 idade = c(18, 25, 49, 38),
                 idade_q = idade^2,
                 idade_media = mean(idade) )
```

Exercício 5

Enap

Exercício

- Crie um data.frame com 15 linhas e 5 colunas em que:
  - a primeira coluna seja uma sequência de 1 a 15;
  - a segunda seja uma coluna de letras maiúsculas;
  - a terceira compreenda os números 1, 2 e 3;
  - a quarta seja a a multiplicação da soma da coluna 1 e 3 por 5;
  - a quinta seja um vetor lógico que indique se os valores da quarta coluna sejam maiores do que sua média.