

Lidando com Tabelas de dados (data frames)

60

Data frames em R

Operações básicas com Data frames

Operação	Descrição
<code>cbind(x,y)</code>	Adiciona a coluna y ao data frame x
<code>rbind(x,y)</code>	Adiciona a linha y ao data frame x
<code>x[a,b]</code>	Acesso o elemento da linha a, coluna b
<code>x[,b]</code>	Acesso todos elementos da coluna b
<code>x[a,]</code>	Acesso todos elementos da linha a
<code>x[2:4, 5:8]</code>	Acesso os valores das linhas 2 a 4 nas colunas 5 a 8
<code>x\$y</code>	Acessa a coluna y do data frame x
<code>x\$y[n]</code>	Acessa o elemento n da coluna y no data frame x
<code>x\$y["condição"]</code>	Acessa os valores que obedecem à condição da coluna y no data frame x
<code>x\$y <- a</code>	Atualiza (ou cria) a coluna y com os valores de a
<code>x\$y[n] <- a</code>	Atualiza o elemento n da coluna y com os valores de a
<code>x\$y <- NULL</code>	Elimina a coluna y
<code>x[-n]</code>	Exclui a coluna de posição n

DICA DE OURO!
A função `names()` extrai os nomes das colunas

Alcoforado, 2021

61

Data frames em R

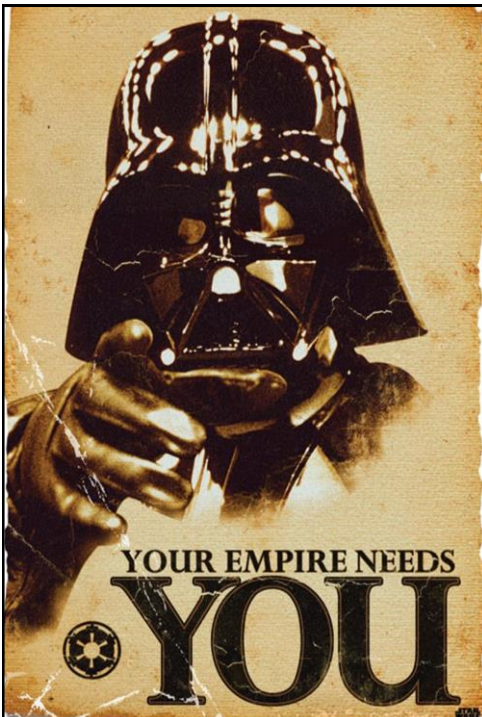
Exemplo

```
> dados <- cbind(dados, Tipo = c("PCR", "ELISA", "PCR"))
> dados <- rbind(dados, data.frame(paciente_id = 4,
  teste_resultado = "Negativo",
  teste_valor = 95,
  teste_data = "2003-05-12",
  Tipo = "PCR"))
> dados$teste_data[2]
[1] "2004-04-10"
```

```
> dados$teste_valor[dados$Tipo == "PCR"]
[1] 70 220 95
> dados[dados$Tipo == "PCR", ]
> dados$PCR <- ifelse(dados$Tipo ==
  "PCR", "Sim", "Não")
```

```
> print(dados)
  paciente_id teste_resultado teste_valor teste_data  Tipo
1           1      Negativo          70 2000-05-10   PCR
2           2      Positivo          80 2004-04-10  ELISA
3           3      Positivo         220 2005-03-10   PCR
4           4      Negativo          95 2003-05-12   PCR
```

62



Agora é com você!

63

Data frames em R

Operações encadeadas com pipe (|> ou %>%)

Implementa escrita lógica do código aprimorando a interpretação e reprodutibilidade

Exemplo:

Obtenha o desvio padrão dos valores dos testes abaixo de 200 padrão.
Arredonde para nenhuma casa decimal

Opção 1 (sem pipe)

```
> round(sd(dados$teste_valor[dados$teste_valor<200]),0)
[1] 7
```

Opção 2 (com pipe)

```
> dados$teste_valor[dados$teste_valor<200] |> sd() |> round(0)
[1] 7
```



DICA DE OURO!

Apenas versões atualizadas do R reconhecem o pipe no formato |>

```
> print(dados)
  paciente_id teste_resultado teste_valor teste_data
1           1      Negativo          70 2000-05-10
2           2      Positivo          80 2004-04-10
3           3      Positivo         220 2005-03-10
```

Alcoforado, 2021

64

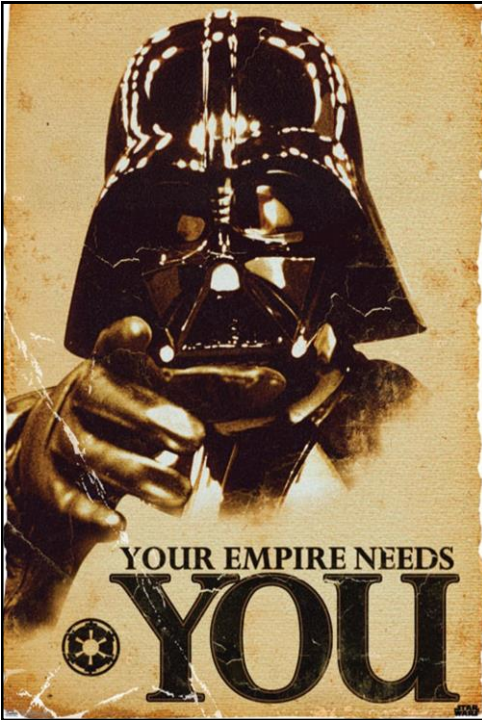
Data frames em R

Caso o padrão não atenda, podemos definir o tipo de dado que o data frame recebe com as funções "as":

```
> dados <- data.frame(paciente_id = c(1:3),
  teste_resultado = as.factor(c("Negativo", "Positivo", "Positivo")),
  teste_valor = c(70, 80, 220),
  teste_data = as.Date(c("2000-05-10", "2004-04-10", "2005-03-10"), "%Y-%m-%d")
)
```

```
> str(dados)
'data.frame':  3 obs. of  4 variables:
 $ paciente_id   : int  1 2 3
 $ teste_resultado: Factor w/ 2 levels "Negativo","Positivo": 1 2 2
 $ teste_valor   : num  70 80 220
 $ teste_data    : Date, format: "2000-05-10" "2004-04-10" "2005-03-10"
```

65



Agora é com você!

66

Data frames em R

Variáveis ordinais

A função `factor()` também permite atribuir uma ordem às variáveis nominais, tornando-as variáveis ordinais. Isso é feito definindo o parâmetro de pedido como `TRUE` e atribuindo um vetor com a hierarquia de nível desejada aos níveis de argumento.

Exemplo: Considere um vetor “escolaridade” com as categorias “Fundamental”, “Médio” e “Superior”.

```
> escolaridade <- c("Fundamental", "Superior", "Superior", "Médio", "Fundamental")
> str(escolaridade)
chr [1:5] "Fundamental" "Superior" "Superior" "Médio" "Fundamental"

> escolaridade <- factor(escolaridade,
  order = TRUE,
  levels = c("Fundamental", "Médio", "Superior"))
> str(escolaridade)
Ord.factor w/ 3 levels "Fundamental"<..: 1 3 3 2 1
```

67

Data frames em R

Você pode usar a função `as.Date()` para converter dados de caracteres em datas. O formato padrão é `aaaa-mm-dd`, e pode ser definido ou manipulado:

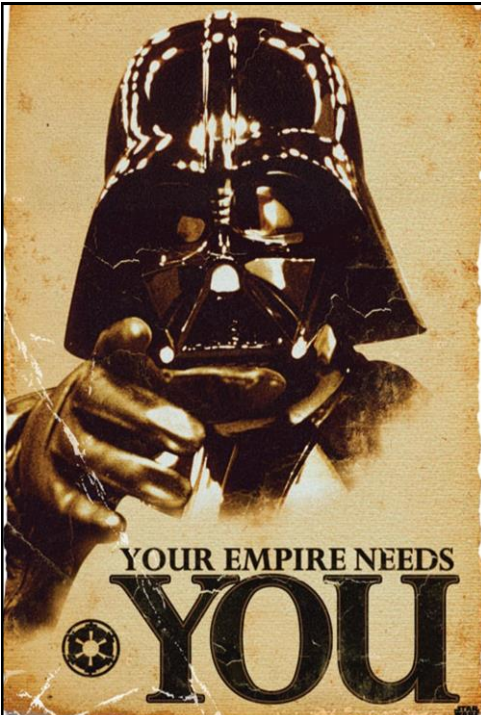
Símbolo	Entrada	Resultado
%d	Dia como número (0-31)	11324
%a	Dia da semana abreviado	Mon
%A	Dia da semana não abreviado	Monday
%m	Mês (00-12)	00-12
%b	Mês abreviado	Jan
%B	Mês não abreviado	January
%y	Ano com 2 dígitos	7
%Y	Ano com 4 dígitos	2007

As datas são armazenadas como o número de dias desde 01/01/1970, com valores negativos para datas anteriores (Hora Unix)

DICA DE OURO!

A função `format()` permite extrair o valor de dia, mês ou ano de um campo de data. Ex: `format(x, %d)` extrai o dia do campo de data x

68



Agora é com você!

69

Data frames em R

Operações básicas com texto (**character**)

Função	Descrição
<code>substr(x, start=a, stop=b)</code>	Extraí (ou substituí) partes de um vetor de caracteres <i>x</i> , iniciando no caracter <i>a</i> e parando em <i>b</i> .
<code>strsplit(x, a)</code>	Divide os elementos do vetor de caracteres <i>x</i> no marcador <i>a</i> .
<code>paste(..., sep="a")</code>	Concatena as <i>strings</i> com marcador <i>a</i> .
<code>toupper(x)</code>	Transforma em caixa alta
<code>tolower(x)</code>	Transforma em caixa baixa

PRA SABER MAIS!

As funções `grep()` e `sub()` permitem trabalhar com expressões regulares REGEX (<https://stat.ethz.ch/R-manual/R-devel/library/base/html/grep.html>)

Alcoforado, 2021

70

Data frames em R

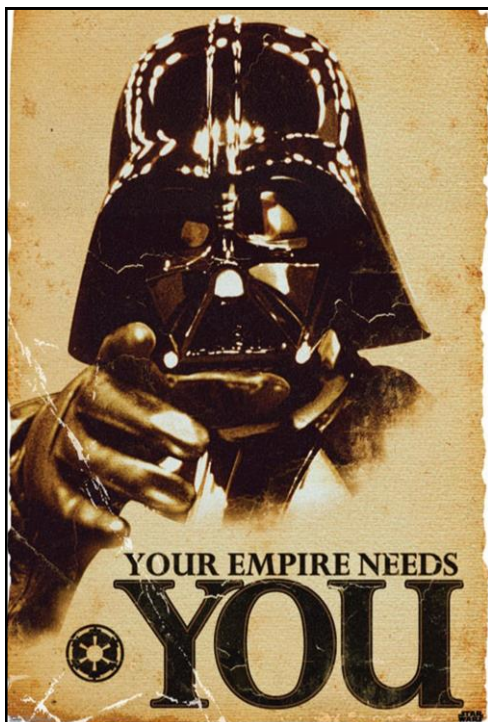
Exemplo

```
> substr(dados$teste_data, start=1, stop=4)
[1] "2000" "2004" "2005"
> strsplit(dados$teste_data, "-")
[[1]]
[1] "2000" "05" "10"
[[2]]
[1] "2004" "04" "10"
[[3]]
[1] "2005" "03" "10"
> typeof(x)
[1] "list"
> x[[1]][2]
[1] "05"
```

```
> paste("Paciente", dados$paciente_id, sep=": ")
[1] "Paciente: 1" "Paciente: 2" "Paciente: 3"
> toupper(dados$teste_resultado)
[1] "NEGATIVO" "POSITIVO" "POSITIVO"
> tolower(dados$teste_resultado)
[1] "negativo" "positivo" "positivo"
```

```
> print(dados)
  paciente_id teste_resultado teste_valor teste_data
1           1      Negativo          70 2000-05-10
2           2      Positivo          80 2004-04-10
3           3      Positivo         220 2005-03-10
```

71



Agora é com você!

72



Universidade de Brasília
Faculdade de Ciências da Saúde
Departamento de Saúde Coletiva

Obrigado

Ivan Zimmermann
ivan.zimmermann@unb.br



73

73