

Comandos básicos do R: Guia de bolso

Lucas C. Germano

2022-05-15

Contents

Sobre este livro

Sejam bem-vindos!

O objetivo deste livro é disponibilizar para consulta anotações de códigos R de forma prática e rápida. Não há explicações aprofundadas nem se pretende esgotar as possibilidades do conteúdo apresentado, assim, esta documentação deve ser utilizada somente como um guia rápido, pois não passa de um conjunto de rascunhos apreendidos no dia-a-dia da manipulação de dados e na apresentação de resultados. O conteúdo poderá ser baixado nos formatos **.pdf** ou **epub**, mas a proposta é que o conteúdo seja dinâmico, com atualizações semanais. A estrutura de construção está disponível no GitHub.

Críticas, sugestões ou contribuições de código e conteúdo podem ser enviadas para lucascgermano@gmail.com. Ficarei muito feliz, qualquer que seja o motivo do contato.

Chapter 1

Leitura de arquivos de texto

1.1 Definição de diretório de trabalho

Comando	Definição
setwd() Ctrl + Shift + h	Define diretório de trabalho. Abre janela de navegação para definir diretório.
file.choose()	Abre janela de navegação e ao selecionar o arquivo, ele retorna o caminho (diretório). Pode-se usar também dentro do comando, como em <code>read.csv2(file = file.choose())</code> .
No RStudio: Session > Setting Working Directory Inserir aspas ' ' + Tab entre elas	Equivalente a Ctrl + Shift + h Navegação que pode servir para explorar caminhos.

1.2 Arquivos .csv

1.2.1 read.csv2()

.csv = Arquivos separados por vírgula (,)

.csv2 = Arquivos separados por ponto e vírgula (;)

Os argumentos das funções são os mesmos, por isso o exemplo será dado somente para .csv2 (mais usado)

```
dados <- read.csv2(file = 'dados/dados.csv')
head(dados, 5)           # Exibir as 5 primeiras linhas dos dados.
```

```
##      X      data code_mn      muni  faixa casos obitos masc fem  ano mes semana
## 1 1 2020-01-01  353070 Mogi Guaçu 30 a 39      1      0      0      1 2020      1      1
## 2 2 2020-01-20  353070 Mogi Guaçu 50 a 59      1      0      1      0 2020      1      3
## 3 3 2020-01-29  352380      Itobi 30 a 39      1      0      1      0 2020      1      5
## 4 4 2020-01-30  353050      Mococa 70 a 79      1      0      0      1 2020      1      5
## 5 5 2020-02-02  353080 Mogi Mirim 40 a 49      1      0      0      1 2020      2      5
##      pop
## 1 150713
## 2 150713
## 3      7830
## 4 68788
## 5 92715
```

Argumentos principais

Os argumentos são os mesmos da função `read.table()`.

Argumento	Definição
<code>file</code>	Nome do arquivo que será lido, contendo o caminho do diretório.
<code>header</code>	Logical. Indica se o arquivo contém os nomes das colunas na primeira linha
<code>sep</code>	Tipo de separador de campo. Default é = “;”.
<code>dec</code>	Tipo de separador de decimal. Default é = “.”.
<code>nrows</code>	Integer. Número máximo de linhas a serem lidas.
<code>skip</code>	Integer. Número de linhas que serão puladas antes de iniciar a leitura dos dados.
<code>fill</code>	Logical. Se TRUE, caso as linhas tenham comprimento desigual, são adicionados campos em branco.
<code>blank.lines.skip</code>	Logical. Se TRUE linhas vazias serão ignoradas.

Argumento	Definição
stringsAsFactors	Logical. Se TRUE os vetores character serão convertidos para factors. Se houver distorção dos caracteres, utilizar FALSE para serem mantidos os caracteres originais, sem conversão.
fileEncoding	Character string. Define o encoding que será usado. Ex. fileEncoding = “UTF-8” ou “Latin-1” ou “ISO-8859-1”.
skipNull	Logical. Se TRUE os nulos (NA) devem ser ignorados.
colClasses	character. Um vetor de classes referentes as colunas. Valores possíveis são NA (default, quando type.convert é usado), “NULL” (quando a coluna é pulada), um vetor atômico de classes(logical, integer, numeric, complex, character, raw), or “factor”, “Date” or “POSIXct”.

1.2.2 readr::read_csv2()

Exemplo 1

```
dados <- readr::read_csv2(file = 'dados/dados.csv', # Caminho e arquivo
                          col_select = c(2,4:7),   # Seleção de colunas de forma numérica (é in
                          guess_max = 1000,         # Máximo de linhas utilizadas para adivinha
                          skip_empty_rows = TRUE)   # Pular linhas vazias
head(dados, 5)                                     # Exibir as 5 primeiras linhas dos dados.
```

```
## # A tibble: 5 x 5
##   data      muni      faixa  casos obitos
##   <date>    <chr>    <chr>   <dbl> <dbl>
## 1 2020-01-01 Mogi Guaçu 30 a 39     1     0
## 2 2020-01-20 Mogi Guaçu 50 a 59     1     0
## 3 2020-01-29 Itobi     30 a 39     1     0
## 4 2020-01-30 Mococa    70 a 79     1     0
## 5 2020-02-02 Mogi Mirim 40 a 49     1     0
```

Exemplo 2

```

dados <- readr::read_csv2('dados/dados.csv',           # Caminho e arquivo
                          guess_max = 1000,           # Máximo de linhas uti
                          skip_empty_rows = TRUE,     # Pular linhas vazias
                          skip = 1,                  # Pular primeira linha
                          col_names = c('a','b','c','d','e'), # Definir nomes das co
                          col_select = c('a','b','c','d','e')) # Selecionar colunas
head(dados, 5)                                       # Exibir as 5 primeira

```

```

## # A tibble: 5 x 5
##       a b      c d      e
##   <dbl> <date>   <dbl> <chr>   <chr>
## 1     1 2020-01-01 353070 Mogi Guaçu 30 a 39
## 2     2 2020-01-20 353070 Mogi Guaçu 50 a 59
## 3     3 2020-01-29 352380 Itobi      30 a 39
## 4     4 2020-01-30 353050 Mococa     70 a 79
## 5     5 2020-02-02 353080 Mogi Mirim 40 a 49

```

Argumentos principais

Argumento	Definição
file	Nome do arquivo que será lido, contendo o caminho do diretório (admite http). Arquivos terminados em .gz, .bz2, .xz, ou .zip serão automaticamente descomprimidos.
col_names	TRUE ou FALSE ou um vetor tipo caracter com nomes das colunas. Se TRUE, a primeira linha será usada para nomear as colunas. Se FALSE, nomes das colunas serão gerados automaticamente (X1, X2, X3 etc). Se col_names for um vetor com nomes, os valores serão usados como nomes das colunas, mas a primeira linha será considerada no banco (nomes errados), assim, pode-se usar o argumento renomeando as colunas, mas fazendo a leitura sem considerar a primeira linha, com [-1,] ou skip = 1. Colunas sem nome (NA) receberão nomes fictícios.

Argumento	Definição
<code>col_types</code>	Se for NULL, todos as classes de coluna serão imputadas a partir do máximo de linhas lidas (<code>guess_max</code>) intercaladas por todo o arquivo. Se a imputação falhar, você precisará aumentar o <code>guess_max</code> ou fornecer os tipos corretos você mesmo. As especificações de coluna criadas por <code>list()</code> ou <code>cols()</code> devem conter uma especificação de coluna para cada coluna. Se você quiser ler apenas um subconjunto das colunas, use <code>cols_only()</code> . Para compactar um vetor com as classes, usar as letras <code>c</code> = character, <code>i</code> = integer, <code>n</code> = number, <code>d</code> = double, <code>l</code> = logical, <code>f</code> = factor, <code>D</code> = date, <code>T</code> = date time, <code>t</code> = time, <code>?</code> = guess, <code>_</code> or <code>-</code> = skip; Por padrão, a leitura de um arquivo sem uma especificação de coluna imprimirá uma mensagem mostrando o que o leitor adivinhou. Para remover esta mensagem, defina <code>show_col_types = FALSE</code> ou defina <code>'options(readr.show_col_types = FALSE)</code> .
<code>col_select</code>	Colunas a serem incluídas nos resultados, equivale a <code>dplyr::select()</code> para se referir às colunas pelo nome. Use <code>c()</code> ou <code>list()</code> para usar mais de uma expressão de seleção. Embora esse uso seja menos comum, <code>col_select</code> também aceita um índice de coluna numérica.
<code>locale</code>	A localidade controla os padrões que variam de lugar para lugar. A localidade padrão é centrada nos EUA (como R), mas você pode usar <code>locale()</code> para criar sua própria localidade que controla coisas como o fuso horário padrão, codificação, marca decimal, marca grande e nomes de dia/mês.

