Importação de dados

Nessa leitura, você vai aprender o básico da importação de dados em R. O software vem com conjuntos de dados embutidos que são ótimas ferramentas para aprender como usar R e praticar a análise de dados. Você vai explorar a função **data()** e aprender como carregar conjuntos de dados de amostra RStudio. Depois, você vai ver como usar dois pacotes do tidyverse – readr e o readxl – para importar arquivos de outras fontes em R. Você vai aprender como usar o readr para ler um arquivo .csv, e como usar o readxl para ler um arquivo .xlsx.

# A função data()

A instalação padrão de R vem com um número de conjuntos de dados pré-carregados com os quais você pode praticar. Essa é uma ótima maneira de desenvolver suas habilidades em R e aprender sobre importantes funções de análise de dados. Além disso, muitos tutoriais e recursos online usam esses conjuntos de dados de amostra para ensinar conceitos de codificação em R.

Você pode usar a função **data()** para carregar esses conjuntos de dados em R. Se você executar a função data sem um argumento, o R vai exibir uma lista de todos os conjuntos de dados disponíveis.

data()

Isso inclui uma série de conjuntos de dados pré-carregados do pacote “datasets”.

Se você quiser carregar um conjunto de dados específico, é só colocar o nome dele no parêntesis da função **data()**. Por exemplo, vamos carregar o conjunto de dados *mtcars*, que contém informações sobre os carros que apareceram nas edições passadas da revista *Motor Trend*.

data(mtcars)

Ao executá-la, R vai carregar o conjunto de dados. O conjunto de dados também vai aparecer no painel Ambiente (Environment) do seu RStudio. O painel Ambiente (Environment) exibe os nomes dos objetos de dados, tais como data frames e variáveis, que você tem disponíveis no seu espaço de trabalho atual. Nessa imagem, o *mtcars* aparece na quinta linha do painel. R nos diz que ele contém 32 observações e 11 variáveis.

Agora que o conjunto de dados está carregado, você pode obter uma prévia dele no painel do console de R. É só digitar o nome dele...

mtcars

... e então pressionar ctrl (ou cmnd) e enter.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
| Mazda RX4 | 21,0 | 6 | 160,0 | 110 | 3,90 | 2,620 | 16,46 | 0 | 1 | 4 | 4 |
| Mazda Rx4 Wag | 21,0 | 6 | 160,0 | 110 | 3,90 | 2,875 | 17,02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22,8 | 4 | 108,0 | 93 | 3,85 | 2,320 | 18,62 | 1 | 0 | 4 | 1 |
| Hornet 4 Drive | 21,4 | 6 | 258,0 | 110 | 3,08 | 3,215 | 19,44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18,7 | 8 | 360,0 | 175 | 3,15 | 3,440 | 17,02 | 0 | 0 | 3 | 2 |
| Valiant | 18,1 | 6 | 225,0 | 105 | 2,76 | 3,460 | 20,22 | 0 | 0 | 3 | 1 |
| Duster 360 | 14,3 | 8 | 360,0 | 245 | 3,21 | 3,570 | 15,84 | 0 | 0 | 3 | 4 |
| Merc 250D | 24,4 | 4 | 146,7 | 62 | 3,69 | 3,190 | 20,00 | 0 | 0 | 4 | 2 |
| Merc 230 | 22,8 | 4 | 140,8 | 95 | 3,92 | 3,150 | 22,90 | 0 | 0 | 4 | 2 |
| Merc 280 | 19,2 | 6 | 167,6 | 123 | 3.2 | 3,440 | 18,30 | 0 | 0 | 4 | 4 |

Você também pode exibir o conjunto de dados clicando diretamente no nome do conjunto de dados no painel ambiente. Portanto, se você clicar em **mtcars**

no painel Ambiente, R vai executar automaticamente a função View() e exibir o conjunto de dados no visualizador de RStudio.

Tente fazer a experiência com outros conjuntos de dados na lista se você quiser um pouco mais de prática.

# readr

Além de usar os conjuntos de dados embutidos em R, também é útil importar dados de outras fontes para praticar ou analisar. O pacote readr de R é uma ótima ferramenta para a leitura de dados retangulares. Os dados retangulares são dados que se encaixam bonitinho dentro de um retângulo de linhas e colunas, com cada coluna se referindo a uma única variável e cada linha se referindo a uma única observação.

Aqui estão alguns exemplos de tipos de arquivos que armazenam dados retangulares:

* **.csv (valores separados por vírgula)**: um arquivo .csv é um arquivo de texto simples que contém uma lista de dados. Geralmente, esses arquivos usam vírgulas para separar (ou delimitar) dados, mas às vezes eles usam outros caracteres, como ponto e vírgula.
* **.tsv (valores separados por tabulação)**: um arquivo .tsv armazena uma tabela de dados em que as colunas de dados são separadas por tabulações. Por exemplo, um conjunto de dados em tabela, ou planilha de dados.
* **.fwf (arquivos com largura ajustada)**: um arquivo .fwf tem um formato específico que permite o salvamento de dados textuais de uma maneira organizada.
* **.log**: um arquivo .log é um arquivo gerado pelo computador que registra eventos dos sistemas operacionais e outros programas de software.

A base de R possui funções para ler os arquivos, mas as funções equivalentes no readr são, tipicamente, *muito* mais rápidas. Elas também produzem tibbles, que são mais fáceis de usar e ler.

O pacote readr é parte do centro do tidyverse. Então, se você já instalou o tidyverse, você tem o que precisa para começar a

trabalhar com readr. Se você ainda não instalou o tidyverse, você pode fazer isso agora.

## funções de readr

A meta de readr é fornecer um meio rápido e amigável para ler dados retangulares. O readr dá suporte a várias funções read\_. Cada função se refere a um formato de arquivo específico.

* read\_csv(): arquivos com valores separados por vírgula (.csv)
* read\_tsv(): arquivos com valores separados por tabulação
* read\_delim(): arquivos gerais delimitados
* read\_fwf(): arquivos com largura ajustada
* read\_table(): arquivos tabulares onde as colunas são separadas por um espaço em branco
* read\_log(): arquivos de log da web

Essas funções todas possuem sintaxe similar, portanto uma vez que você aprenda como usar uma delas, você pode usar esse conhecimento nas outras. Esse texto vai focar na função read\_csv(), uma vez que os arquivos .csv são as formas mais comuns de armazenamento de dados e você vai trabalhar frequentemente com eles.

Na maioria dos casos, essas funções vão funcionar automaticamente: você vai fornecer um caminho para um arquivo, executar a função, e vai obter um tibble que exibe os dados no arquivo. Por trás das câmeras, o readr analisa o arquivo como um todo e especifica como cada coluna pode ser convertida de um caractere de vetor para um tipo de dado mais apropriado.

## Ler um arquivo csv com readr

O pacote readr vem com alguns arquivos de amostra de conjuntos de dados embutidos que você pode usar como exemplos de código. Para listar esses arquivos de amostra, você pode executar a função readr\_example() sem argumentos.

|  |  |  |
| --- | --- | --- |
| readr\_example() |  |  |
| [1] “challenge.csv” | “epa78.txt” | “exemplo.log” |
| [4] “fwf-sample.txt” | “massey-rating.txt” “mtcars.csv” | |
| [7] “mtcars.csv.bz2” | “mtcars.csv.zip” |  |

O arquivo “mtcars.csv” se refere ao conjunto de dados mtcars que foi mencionado mais cedo. Vamos usar a função **read\_csv()** para ler o arquivo “mtcars.csv” , como exemplo. No parêntesis, você vai fornecer o caminho para o arquivo. Nesse caso, é o “readr\_example(“mtcars.csv”).

read\_csv(readr\_example(“mtcars.csv”))

Quando você executar a função, R vai imprimir na tela uma especificação de coluna que fornece o nome e o tipo de cada coluna.

── **Especificação da coluna** ─────────────────────────────────────────────────── cols(

mpg = col\_double(),

cyl = col\_double(),

disp = col\_double(),

hp = col\_double(),

drat = col\_double(),

wt = col\_double(),

qsec = col\_double(),

vs = col\_double(),

am = col\_double(),

gear = col\_double(),

carb = col\_double()

)

R também elabora um tibble

# O tibble: 32 x 11

mpg cyl disp hp drat wt qsec vs am gear carb

*<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>*

1 21 6 160 110 3,9 2,62 16,5 0 1 4 4

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 18,1 | 6 | 225 | 105 | 2,76 | 3,46 | 20,2 | 1 | 0 | 3 | 1 |
| 7 | 14,3 | 8 | 360 | 245 | 3,21 | 3.57 | 15,8 | 0 | 0 | 3 | 4 |
| 8 | 24,4 | 4 | 147. | 62 | 3,69 | 3,19 | 20 | 1 | 0 | 4 | 2 |
| 9 | 22,8 | 4 | 141. | 95 | 3,92 | 3,15 | 22,9 | 1 | 0 | 4 | 2 |
| 10 | 19,2 | 6 | 168. | 123 | 3,92 | 3,44 | 18,3 | 1 | 0 | 4 | 4 |

# … com mais 22 linhas

# readxl

Para importar planilhas de dados para R, você pode usar o pacote readxl. Esse pacote facilita a transferência de dados de Excel para R. O readxl oferece suporte tanto para os arquivos dos antigos formatos .xls quanto para os formatos modernos baseados em xml e .xls.

O pacote readxl é parte do tidyverse, mas não é um pacote do núcleo de tidyverse, portanto você precisa baixar o readxl em R por meio da função library().

library(readxl)

## Ler um arquivo .csv com readxl

Do mesmo jeito que o pacote readr, readxl vem com alguns arquivos de amostra de conjuntos de dados embutidos que você pode usar para praticar. Você pode executar o código readxl\_example() para ver a lista.

Você pode usar a função **read\_excel()** para ler um arquivo de planilha da mesma forma que você usou a função read\_csv() para ler um arquivo .csv. O código para ler o arquivo de exemplo “type-me.xlsx” inclui o caminho para o arquivo nos parênteses da função.

read\_excel(readxl\_example(“type-me.xlsx”))

Você pode usar a função [excel\_sheets()](https://readxl.tidyverse.org/reference/excel_sheets.html) para listar os nomes de cada planilha individualmente.

excel\_sheets(readxl\_example(“type-me.xlsx”))

[1] “logical\_coercion” “numeric\_coercion” “date\_coercion” “text\_coercion”

Você pode usar a função [excel\_sheets()](https://readxl.tidyverse.org/reference/excel_sheets.html) para listar os nomes de cada planilha individualmente.

excel\_sheets(readxl\_example(“type-me.xlsx”))

1. “logical\_coercion” “numeric\_coercion” “date\_coercion” “text\_coercion”

Você também pode especificar uma planilha por nome ou número. É só digitar “sheet =” seguido pelo nome ou número da planilha. Por exemplo, você pode usar a planilha chamada “numeric\_coercion” da lista acima.

read\_excel(readxl\_example(“type-me.xlsx”), sheet =“numeric\_coercion”)

Quando você executa a função, R retorna um tibble da planilha.

# O tibble: 7 x 2

`maybe numeric?` explanation

|  |  |
| --- | --- |
| <chr> | <chr> |
| 1 NA | “vazio” |
| 2 TRUE | “booleano verdadeiro” |
| 3 FALSE | “booleano falso” |
| 4 40534 | “datahora” |
| 5 123456 | “a string \”123456\”” |
| 6 123456 | “o número 123456” |
| 7 repolho | “\”repolho\”” |

## Recursos

* + Se você quer aprender como usar as funções do readr para trabalhar com arquivos mais complexos, dê uma olhada no [capítulo “Importação de dados” do livro R](https://r4ds.had.co.nz/data-import.html) [for Data Science](https://r4ds.had.co.nz/data-import.html). Ele explora alguns dos problemas mais comuns que você pode encontrar quando ler os arquivos, e como usar readr para lidar com esses problemas.
  + A entrada do [readxl](https://readxl.tidyverse.org/)  na documentação de tidyverse dá uma boa noção das funções básicas de readxl, fornece uma explicação detalhada de como o pacote funciona e os conceitos de codificação por trás dele, e oferece links para outros recursos úteis.
  + O pacote “datasets” de R contém muitos conjunto de dados úteis pré-carregados. Verifique a lista de conjuntos de dados no site [R Datasets](https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html) [Package.](https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html)  Ela inclui links para descrições detalhadas de cada conjunto de dados.