



A estrutura de dados tibble

Prof. Walmes Zeviani

walmes@ufpr.br

Laboratório de Estatística e Geoinformação
Departamento de Estatística
Universidade Federal do Paraná

A detailed architectural line drawing of a classical building facade. The drawing shows a prominent pediment supported by a series of columns. The text 'UNIVERSIDADE DO PARANÁ' is visible on the frieze below the pediment. The drawing is rendered in a light, sketchy style, typical of architectural blueprints or conceptual drawings.

Um overview do tibble

Motivação

- ▶ O `data.frame` é a estrutura nativa (primitiva) para representar tabelas de dados.
- ▶ O `tibble` é uma reimplementação da estrutura com melhorias.
 - ▶ Método `print` mais enxuto e informativo.
 - ▶ Mais consistente para seleção e modificação de conteúdo.
 - ▶ Mais fácil conversão de outros formatos para `tibble`.
 - ▶ Colunas/cédulas podem representar objetos mais complexos.
- ▶ Documentação:
 - ▶ <https://tibble.tidyverse.org/>.
 - ▶ <https://r4ds.had.co.nz/tibbles.html>.
 - ▶ <https://cran.r-project.org/package=tibble>

A ficha técnica

tibble: Simple Data Frames

Provides a 'tbl_df' class (the 'tibble') that provides stricter checking and better formatting than the traditional data frame.

Version: 2.1.1
Depends: R (≥ 3.1.0)
Imports: [cli](#) (≥ 1.0.1), [crayon](#) (≥ 1.3.4), [fansj](#) (≥ 0.4.0), methods, [pillar](#) (≥ 1.3.1), [pkgconfig](#) (≥ 2.0.2), [rlang](#) (≥ 0.3.1), utils
Suggests: [bench](#) (≥ 1.0.1), [covr](#) (≥ 3.2.1), [dplyr](#) (≥ 0.7.8), [htmltools](#) (≥ 0.3.6), [import](#) (≥ 1.1.0), [knitr](#) (≥ 1.21), [mockr](#) (≥ 0.1), [nycflights13](#) (≥ 1.0.0), [rmarkdown](#) (≥ 1.11), [testthat](#) (≥ 2.0.1), [withr](#) (≥ 2.1.2)
Published: 2019-03-16
Author: Kirill Müller [aut, cre], Hadley Wickham [aut], Romain Francois [ctb], Jennifer Bryan [ctb], RStudio [cph]
Maintainer: Kirill Müller <krlmlr+r at mailbox.org>
BugReports: <https://github.com/tidyverse/tibble/issues>
License: [MIT](#) + file [LICENSE](#)
URL: <http://tibble.tidyverse.org/>, <https://github.com/tidyverse/tibble>
NeedsCompilation: yes
Materials: [README](#) [NEWS](#)
CRAN checks: [tibble results](#)

Figura 1. Ficha técnica do tibble. Fonte: .

Formas de criar um tibble

Criação por especificação de colunas (*column-by-column*)

```
library(tidyverse) 1  
# packageVersion("tibble") 2
```

```
tb <- tibble(x = 1:3, y = letters[1:length(x)], z = x^2) 1  
tb 2
```

```
## # A tibble: 3 x 3  
##       x y       z  
##   <int> <chr> <dbl>  
## 1     1 a         1  
## 2     2 b         4  
## 3     3 c         9
```

Formas de criar um tibble

Criação por especificação de linhas (row-by-row)

```
tribble(  
  ~x, ~y, ~z,  
  "a", 2, 3.6,  
  "b", 1, 8.5  
)
```

1
2
3
4
5

```
## # A tibble: 2 x 3  
##   x         y         z  
##   <chr> <dbl> <dbl>  
## 1 a         2     3.6  
## 2 b         1     8.5
```

Formas de criar um tibble

Coerção de vetores nomeados

```
notas <- c("André" = 6, "Larissa" = 9, "Mariana" = 8, "Tobias" = 3) 1
notas <- notas %>% 2
  enframe(name = "aluno", value = "nota") 3
notas 4
```

```
## # A tibble: 4 x 2
##   aluno   nota
##   <chr> <dbl>
## 1 André     6
## 2 Larissa   9
## 3 Mariana   8
## 4 Tobias     3
```

Formas de criar um tibble

Coerção de matrizes

```
matrix(1:12, ncol = 3) %>%  
  as_tibble()
```

1
2

```
## # A tibble: 4 x 3  
##       V1     V2     V3  
##   <int> <int> <int>  
## 1     1     5     9  
## 2     2     6    10  
## 3     3     7    11  
## 4     4     8    12
```


Formas de criar um tibble

Coerção de data.frames

```
iris %>%  
  as_tibble()
```

```
## # A tibble: 150 x 5  
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>  
## 1         5.1         3.5           1.4           0.2 setosa  
## 2         4.9         3             1.4           0.2 setosa  
## 3         4.7         3.2           1.3           0.2 setosa  
## 4         4.6         3.1           1.5           0.2 setosa  
## 5         5           3.6           1.4           0.2 setosa  
## 6         5.4         3.9           1.7           0.4 setosa  
## 7         4.6         3.4           1.4           0.3 setosa  
## 8         5           3.4           1.5           0.2 setosa  
## 9         4.4         2.9           1.4           0.2 setosa  
## 10        4.9         3.1           1.5           0.1 setosa  
## # ... with 140 more rows
```

1
2

Formas de criar um tibble

Usando objetos de classe diferente

```
# lv <- iris$Species %>% unique() 1
# lv <- iris %$% Species %>% unique() 2
lv <- iris %>% distinct(Species) %>% pull() 3
vals <- by(data = iris[, -5], INDICES = iris$Species, FUN = as.matrix) 4
tibble(Species = lv, Data = vals) 5
```

```
## # A tibble: 3 x 2
##   Species Data
##   <fct>   <list>
## 1 setosa   <dbl [50 x 4]>
## 2 versicolor <dbl [50 x 4]>
## 3 virginica <dbl [50 x 4]>
```

Métodos disponíveis para a classe

```
tb <- as_tibble(iris)
class(tb)
```

1
2

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

```
methods(class = "tbl")
```

1

```
## [1] as.tbl  format  fortify  glimpse  print  tbl_sum
## see '?methods' for accessing help and source code
```

```
methods(class = "tbl_df")
```

1

```
## [1] all.equal      anti_join      arrange_      arrange
## [5] as_data_frame  as.data.frame  as_tibble     auto_copy
## [9] distinct_     distinct      filter_       filter
## [13] format         fortify        full_join     inner_join
## [17] left_join      mutate_       mutate        print
## [21] right_join     row.names<-   semi_join     slice_
## [25] slice          summarise_    summarise     [
## [29] [             $             type_sum
## see '?methods' for accessing help and source code
```

Operações com tibble

Seleção

```
tb <- iris[1:4, ] %>%  
  as_tibble()
```

```
# Resulta em vetor.
```

```
tb$Petal.Length
```

```
tb[["Petal.Length"]]
```

```
tb[[3]]
```

```
# Resulta em `tibble`!
```

```
tb[, 3]
```

```
tb[, "Petal.Length"]
```

```
tb[, c(3:4)]
```

```
tb[1, ]
```

1

2

3

4

5

6

7

8

9

10

11

12

13

Operações com tibble

Adicionar colunas

```
notas <- c("André" = 6, "Larissa" = 9, "Mariana" = 8, "Tobias" = 3) 1
notas <- notas %>% enframe(name = "aluno", value = "nota")           2
notas <- notas %>%                                                 3
  add_column(faltas = c(12, 8, 0, 18))                               4
notas                                                                5
                                                                    6
```

```
## # A tibble: 4 x 3
##   aluno   nota faltas
##   <chr> <dbl> <dbl>
## 1 André     6     12
## 2 Larissa   9      8
## 3 Mariana   8      0
## 4 Tobias     3     18
```

Operações com tibble

Adicionar linhas

```
notas <- notas %>%  
  add_row(aluno = c("Roberto", "Sabrina"),  
          nota = c(8, 7))  
notas
```

1
2
3
4

```
## # A tibble: 6 x 3  
##   aluno   nota faltas  
##   <chr> <dbl> <dbl>  
## 1 André     6     12  
## 2 Larissa   9      8  
## 3 Mariana   8      0  
## 4 Tobias     3     18  
## 5 Roberto   8     NA  
## 6 Sabrina   7     NA
```



Exercícios para usar o tibble

Criar um tibble a partir:

1. do objeto `precip`.
2. do objeto `cars`.
3. do objeto `mtcars`.
4. do objeto `anscombe` com 3 colunas: grupo, x e y.
5. do objeto `WorldPhones`.
6. do objeto `HairEyeColor`.
7. dos dados exibidos nos dois *donet plot* em <http://www.vizwiz.com/2016/06/data-scientists-need-alteryx.html> com 6 linhas e 3 colunas: tarefa, tempo, enfado.
8. dos dados disponíveis em <http://leg.ufpr.br/~walmes/data/bib1.txt> usando `tribble()`.