

Dados arrumados (tidy data)

Conceito

Tidy data é um princípio de organização de dados que ajuda a simplificar a análise de dados. A motivação desse conceito é que, no mundo real, lidamos com dados em diversos formatos e estruturas, o que pode dificultar a análise. Ao colocar o princípio de dados arrumados em prática, aceleramos o processo de análise e facilitamos a comunicação dos resultados.

O conceito foi proposto por Hadley Wickham em 2014 e é baseado em três princípios:

1. Cada **observação** forma uma **linha**.
2. Cada **variável** forma uma **coluna**.
3. Cada **valor** forma uma **célula**.

A Figura 1 ilustra esses princípios.

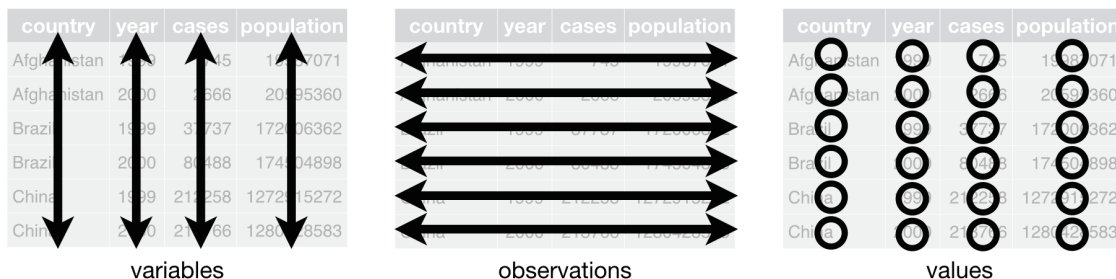


Figura 1: As seguintes três regras tornam um conjunto de dados tidy: variáveis são colunas, observações são linhas e valores são células. Fonte: R for Data Science.

Vamos analisar o primeiro princípio: cada observação forma uma linha. Note que estamos ligando conceitos de duas áreas do conhecimento distintas: a estatística e a computação. Na estatística, uma unidade amostral é um elemento da população que está sendo estudada. Por exemplo, se estamos estudando a altura de alunos de uma escola, cada aluno é uma unidade

amostral. Na computação, uma linha é um registro de um banco de dados ou uma linha de um arquivo de texto. O primeiro princípio do tidy data estabelece que cada unidade amostral deve formar uma linha. Ou seja, se temos, por exemplo, uma tabela no Excel, cada linha (1, 2, 3, ...) dessa tabela deve representar uma unidade amostral.

Exemplo que segue o primeiro princípio:

```
dados <- tribble(
  ~id, ~altura, ~peso,
  1, 1.70, 70,
  2, 1.80, 80,
  3, 1.60, 60
)
knitr::kable(dados)
```

id	altura	peso
1	1.7	70
2	1.8	80
3	1.6	60

Exemplo que não segue o primeiro princípio:

```
dados <- tribble(
  ~id, ~tipo_medida, ~valor,
  1, "altura", 1.70,
  1, "peso", 70,
  2, "altura", 1.80,
  2, "peso", 80,
  3, "altura", 1.60,
  3, "peso", 60
)
knitr::kable(dados)
```

id	tipo_medida	valor
1	altura	1.7
1	peso	70.0
2	altura	1.8
2	peso	80.0
3	altura	1.6
3	peso	60.0

id	tipo_medida	valor
----	-------------	-------

No exemplo acima, a tabela não segue o primeiro princípio do tidy data, pois cada unidade amostral (cada pessoa) não forma uma linha. Em vez disso, cada unidade amostral é dividida em duas linhas, uma para a altura e outra para o peso. Para seguir o primeiro princípio do tidy data, a tabela deveria ser organizada de forma que cada unidade amostral formasse uma linha.

A segunda tabela está no formato que chamamos de “longo” (ou “long format”). Esse formato é comum em bases de dados que não seguem o primeiro princípio do tidy data, mas existem situações em que esse formato é útil (veremos mais adiante).

Para transformar a base de dados que não segue o primeiro princípio do tidy data em uma base de dados que segue o primeiro princípio do tidy data, podemos usar a função `pivot_wider()` do pacote `tidyr`. Essa função transforma a base de dados de longo para largo. Veja o exemplo a seguir:

```
dados <- tribble(
  ~id, ~tipo_medida, ~valor,
  1, "altura", 1.70,
  1, "peso", 70,
  2, "altura", 1.80,
  2, "peso", 80,
  3, "altura", 1.60,
  3, "peso", 60
)

dados_largo <- dados |>
  pivot_wider(names_from = tipo_medida, values_from = valor)

knitr::kable(dados_largo)
```

id	altura	peso
1	1.7	70
2	1.8	80
3	1.6	60

O segundo princípio é que cada variável deve formar uma coluna. Nesse caso, o conceito estatístico de variável é o de variável aleatória, que é uma característica da observação que pode ser medida por algum instrumento. Por exemplo, no exemplo da escola, se estamos estudando

a altura e o peso de alunos, a altura e o peso são variáveis. Já o conceito computacional é o da coluna, que é uma coluna de uma tabela de banco de dados ou de um arquivo de texto. O segundo princípio do tidy data estabelece que cada variável deve formar uma coluna. Ou seja, se temos, por exemplo, uma tabela no Excel, cada coluna (A, B, C, ...) dessa tabela deve representar uma variável.

Exemplo que não segue o segundo princípio:

```
dados <- tribble(
  ~tipo, ~alice, ~bruno, ~carla, ~daniel,
  "altura", 1.70, 1.80, 1.60, 1.75,
  "peso", 70, 80, 60, 75
)
knitr::kable(dados)
```

tipo	alice	bruno	carla	daniel
altura	1.7	1.8	1.6	1.75
peso	70.0	80.0	60.0	75.00

Nesse exemplo, a tabela não segue o segundo princípio do tidy data, pois cada variável (altura e peso) não formam colunas. Em vez disso, cada variável é dividida em quatro colunas, uma para cada pessoa. Vamos arrumar essa tabela para que ela siga o segundo princípio do tidy data. Primeiro, deixamos essa base no formato longo, usando a função `pivot_longer()`:

```
dados_longo <- dados |>
  pivot_longer(cols = -tipo, names_to = "nome", values_to = "valor")
knitr::kable(dados_longo)
```

tipo	nome	valor
altura	alice	1.70
altura	bruno	1.80
altura	carla	1.60
altura	daniel	1.75
peso	alice	70.00
peso	bruno	80.00
peso	carla	60.00
peso	daniel	75.00

Essa base, no entanto, ainda não está arrumada, pois está no formato largo. Para arrumá-la, usamos a função `pivot_wider()`, que vimos anteriormente:

```
dados_arrumados <- dados_longo |>
  pivot_wider(names_from = tipo, values_from = valor)

knitr::kable(dados_arrumados)
```

nome	altura	peso
alice	1.70	70
bruno	1.80	80
carla	1.60	60
daniel	1.75	75

Finalmente, o terceiro princípio é que cada valor deve formar uma célula. Nesse caso, o conceito estatístico de valor é o de valor da variável, que é a medida da variável para uma unidade amostral. Por exemplo, se estamos estudando a altura de alunos de uma escola, o valor da variável altura para um aluno é a altura desse aluno. Já o conceito computacional é o da célula, que é uma célula de uma tabela de banco de dados ou de um arquivo de texto (o cruzamento entre uma linha e uma coluna). O terceiro princípio do tidy data estabelece que cada valor deve formar uma célula. Ou seja, se temos, por exemplo, uma tabela no Excel, cada célula (A1, A2, B1, B2, ...) dessa tabela deve representar um valor.

Exemplo que não segue o terceiro princípio:

```
dados <- tribble(
  ~id, ~altura_peso,
  1, "1.70 / 70",
  2, "1.80 / 80",
  3, "1.60 / 60",
  4, "1.75 / 75"
)

knitr::kable(dados)
```

id	altura_peso
1	1.70 / 70
2	1.80 / 80
3	1.60 / 60
4	1.75 / 75

Nesse exemplo, a tabela não segue o terceiro princípio do tidy data, pois cada valor (altura e peso) não formam células. Em vez disso, os valores são agrupados em uma única célula, que

contém a altura e o peso separados por uma barra. Vamos arrumar essa tabela para que ela siga o terceiro princípio do tidy data. Para isso, dividimos a coluna `altura_peso` em duas colunas, `altura` e `peso`, usando a função `separate()`:

```
dados <- tribble(
  ~id, ~altura_peso,
  1, "1.70 / 70",
  2, "1.80 / 80",
  3, "1.60 / 60",
  4, "1.75 / 75"
)

dados_arrumados <- dados |>
  separate(altura_peso, into = c("altura", "peso"), sep = " / ")

knitr::kable(dados_arrumados)
```

id	altura	peso
1	1.70	70
2	1.80	80
3	1.60	60
4	1.75	75

Até agora, vimos os três princípios do tidy data e como arrumar uma base de dados que não segue esses princípios. No entanto, é importante ressaltar que nem sempre é muito claro se uma base de dados segue ou não os princípios do tidy data. Por exemplo, a base de dados pode estar no formato longo, mas isso não significa necessariamente que ela não segue os princípios do tidy data. Por exemplo, se estamos estudando a evolução da altura e do peso de alunos ao longo do tempo:

```
dados <- tribble(
  ~id, ~mes, ~altura, ~peso,
  "1", 1, 1.70, 70,
  "1", 2, 1.72, 75,
  "1", 3, 1.80, 80,
  "2", 1, 1.80, 80,
  "2", 2, 1.87, 85,
  "2", 3, 1.90, 90,
  "3", 1, 1.60, 60,
  "3", 2, 1.64, 65,
  "3", 3, 1.70, 70
```

```
)
```

```
knitr::kable(dados)
```

id	mes	altura	peso
1	1	1.70	70
1	2	1.72	75
1	3	1.80	80
2	1	1.80	80
2	2	1.87	85
2	3	1.90	90
3	1	1.60	60
3	2	1.64	65
3	3	1.70	70

Nesse caso, é mais fácil organizar esses dados no formato longo, com uma coluna para a altura e outra para o peso, e uma linha para cada observação (altura e peso de um aluno em um determinado momento). Nesse caso, na prática, estamos *definindo* que a observação é a combinação de aluno e período, ainda que cada unidade amostral seja um aluno. Assim, a base de dados está arrumada, mesmo estando no formato longo.

Esse formato, inclusive, é o preferido para fazer visualizações de dados com o pacote `ggplot2`. Por exemplo:

```
dados |>
  ggplot(aes(x = mes, y = altura, color = id)) +
  geom_line() +
  labs(title = "Evolução da altura dos alunos ao longo do tempo")
```

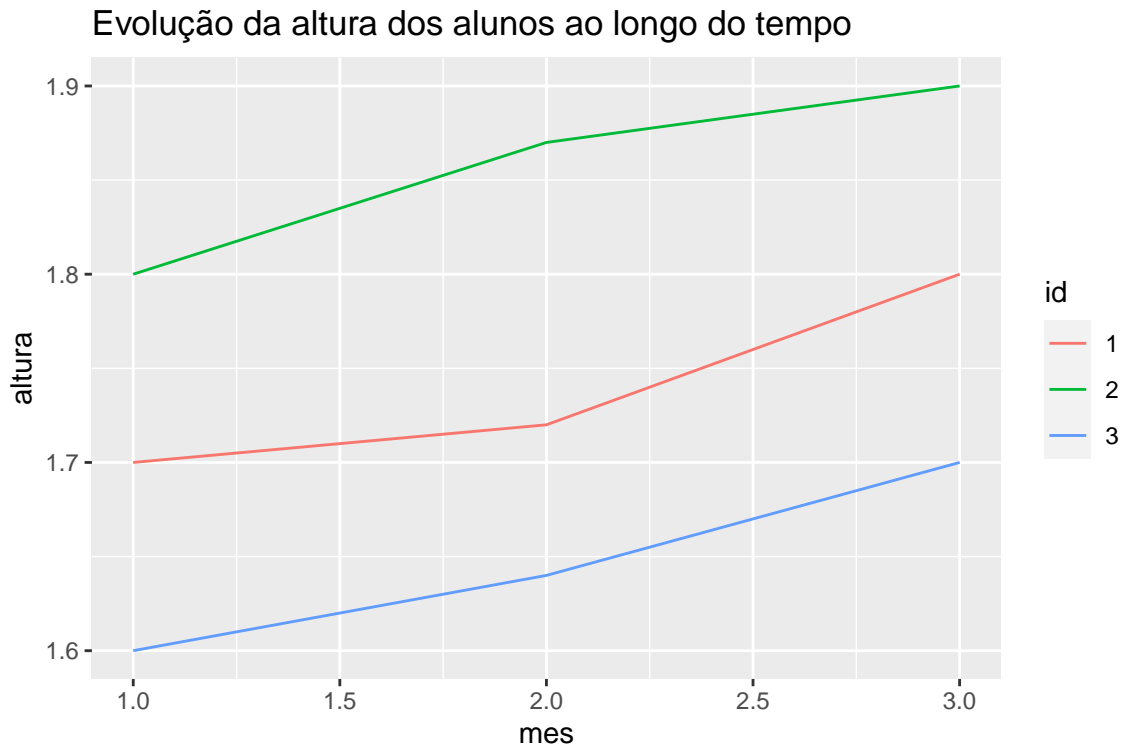


Figura 2: Evolução da altura dos alunos ao longo do tempo. Cada linha representa um aluno.

Ou seja, o princípio do tidy data pode ser adaptado dependendo dos **objetivos de uma análise**. Para cada análise, é importante definir o que é uma observação, o que é uma variável e o que é um valor, de acordo com o contexto do problema. Isso é o que torna o tidy data um conceito tão poderoso e flexível.

Aplicações no Direito

No Direito, tipicamente as bases de dados são organizadas de forma que cada linha representa um processo judicial e cada coluna representa uma variável relacionada ao processo. Por exemplo, podemos ter uma base de dados com as seguintes colunas: **número do processo**, **data de ajuizamento**, **valor da causa**, **tipo de ação**, **juiz responsável**, entre outras. Nesse caso, cada linha representa um processo judicial e cada coluna representa uma variável relacionada ao processo.

Também é possível trabalhar com bases cuja observação não é o processo judicial. Por exemplo, um processo pode ter vários recursos. Nesse caso, podemos colocar cada recurso em uma linha e cada variável relacionada ao recurso em uma coluna. Por exemplo, podemos ter uma base de dados com as seguintes colunas: **número do processo**, **número do recurso**, **data**

de interposição, resultado, relator, entre outras. Nesse caso, cada linha representa um recurso e cada coluna representa uma variável relacionada ao recurso.

(row)	processo	cdacordao	assunto	orgao_julgador	data_julgamento	data_publicacao
1	00019966620178260542	13514786	Roubo Majorado	4ª Câmara de Direito Criminal	2020-04-29	2020-04-29
2	00392749320158260050	11468647	Tráfico de Drogas e Condutas Afins	1ª Câmara de Direito Criminal	2018-05-14	2018-05-21
3	15014595720208260617	15170314	Tráfico de Drogas e Condutas Afins	4ª Câmara de Direito Criminal	2021-11-09	2021-11-09
4	01102558420148260050	10030219	Roubo Majorado	2ª Câmara de Direito Criminal	2016-11-28	2016-12-01
5	15003921120228260545	16355137	Tráfico de Drogas e Condutas Afins	5ª Câmara de Direito Criminal	2022-12-19	2022-12-19
6	00696408120168260050	13053601	Roubo Majorado	11ª Câmara de Direito Criminal	2019-11-06	2019-11-06
7	00008981220208260390	16801854	Tráfico de Drogas e Condutas Afins	12ª Câmara de Direito Criminal	2023-05-30	2023-05-30
8	15005520820198260556	15088031	Tráfico de Drogas e Condutas Afins	12ª Câmara de Direito Criminal	2021-10-07	2021-10-07
9	00018238420198260279	14439211	DIREITO PROCESSUAL PENAL	12ª Câmara de Direito Criminal	2021-03-10	2021-03-10
10	00149412120148260565	15853957	Uso de documento falso	10ª Câmara de Direito Criminal	2022-07-14	2022-07-14

Outro exemplo de caso em que a observação não é o processo é quando trabalhamos com as tabelas de partes ou de movimentações do processo. Nesse caso, cada linha representa uma parte ou uma movimentação do processo e cada coluna representa uma variável relacionada à parte ou à movimentação.

(row)	processo	id_parte	nome	parte	papel
1	1500336-38.2018.8.26.0247	1	RODNEI DANIEL	Apelante	Apelante
2	1500336-38.2018.8.26.0247	1	Miriam Aparecida Silva	Apelante	Apelante
3	1500336-38.2018.8.26.0247	2	Ministério Público do Estado de São Paulo	Apelado	Apelado
4	0058091-55.2008.8.26.0050	1	Andre Henrique dos Santos	Apelante	Apelante
5	0058091-55.2008.8.26.0050	1	Marcos Paulo Ramos Rodrigues Farnezi	Apelante	Apelante
6	0058091-55.2008.8.26.0050	2	Grece Cristina da Silva	Apelante	Apelante
7	0058091-55.2008.8.26.0050	2	Leandro Pinheiro Deksny	Apelante	Apelante
8	0058091-55.2008.8.26.0050	3	Regina Celia Strelec	Apelante	Apelante
9	0058091-55.2008.8.26.0050	3	Samuel Honorato da Trindade	Apelante	Apelante
10	0058091-55.2008.8.26.0050	4	Ministério Público do Estado de São Paulo	Apelado	Apelado
11	0047421-18.2012.8.26.0114	1	Abner Joel de Oliveira	Apelante	Apelante
12	0047421-18.2012.8.26.0114	1	Luis Ricardo Bernardes dos Santos	Apelante	Apelante
13	0047421-18.2012.8.26.0114	2	Laércio Bentini Júnior	Apelante	Apelante
14	0047421-18.2012.8.26.0114	2	Rodolpho Pettena Filho	Apelante	Apelante
15	0047421-18.2012.8.26.0114	3	Rodrigo Henrique Teodoro	Apelante	Apelante
16	0047421-18.2012.8.26.0114	3	Noadir Marques da Silva Junior	Apelante	Apelante
17	0047421-18.2012.8.26.0114	4	Alef Clederson Delbue	Apelante	Apelante
18	0047421-18.2012.8.26.0114	4	Viviane Salles Rocha Moreno	Apelante	Apelante
19	0047421-18.2012.8.26.0114	5	Ministério Público do Estado de São Paulo	Apelado	Apelado