Tidying data

Data Science in a Box datasciencebox.org

sales.xlsx

```
clientes <- readxl::read xlsx("sales data/sales.xlsx", sheet = 'clientes')</pre>
clientes
## # A tibble: 2 × 4
    id cliente item 1 item 2
##
                                 item 3
##
         <dbl> <chr> <chr>
                                 <chr>
## 1
            1 pao leite
                                   banana
## 2
            2 leite papel higienico <NA>
precos <- readxl::read_xlsx("sales_data/sales.xlsx", sheet = 'precos')</pre>
precos
## # A tibble: 5 × 2
## item
                   price
                   <dbl>
## <chr>
## 1 abacate
                     2
## 2 banana
                     0.5
## 3 pao
                     1.5
## 4 leite
## 5 papel higienico
```

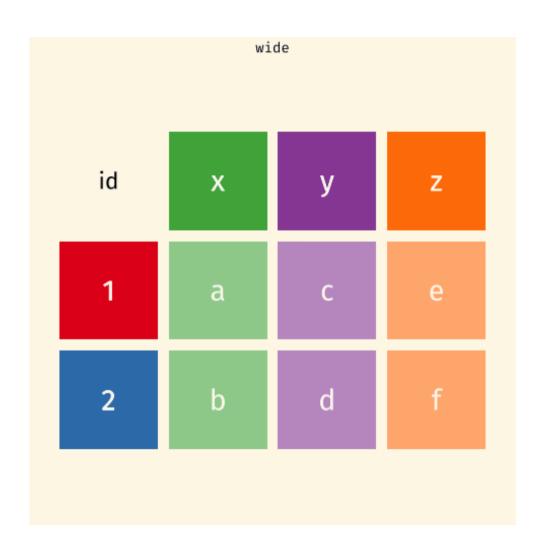
clientes

Temos...

Queremos...

```
## # A tibble: 6 × 3
     id_cliente item_no item
##
         <dbl> <chr>
##
                       <chr>
## 1
              1 item_1
                       pao
## 2
              1 item 2
                       leite
## 3
              1 item_3
                       banana
## 4
              2 item_1 leite
## 5
              2 item_2 papel higienico
## 6
              2 item_3
                       <NA>
```

O objetivo



Wide vs. long

wide

mais colunas / variáveis

long

mais linhas / observações

```
## # A tibble: 6 × 3
     id_cliente item_no item
          <dbl> <chr>
##
                        <chr>
## 1
              1 item 1
                        pao
## 2
              1 item 2
                        leite
## 3
              1 item 3
                       banana
              2 item_1
## 4
                       leite
## 5
              2 item_2 papel higienico
## 6
              2 item 3
                       <NA>
```

data

```
pivot_longer(
    data,
    cols,
    names_to = "name",
    values_to = "value"
)
```

- data
- cols: colunas a transpor para formato long

```
pivot_longer(
  data,
  cols,
  names_to = "name",
  values_to = "value"
)
```

- data
- cols: colunas a transpor para formato long
- names_to: nome da variável que vai receber os nomes das colunas a transpor para long, como valores

```
pivot_longer(
  data,
  cols,
  names_to = "name",
  values_to = "value"
)
```

- data
- cols: colunas a transpor para formato long
- names_to: nome da variável que vai receber os nomes das colunas a transpor para long, como valores
- values_to: nome da variável que vai receber os valores atualmente dispersos por várias colunas (string)

```
pivot_longer(
  data,
  cols,
  names_to = "name",
  values_to = "value"
)
```

clientes \rightarrow compras

```
compras <- clientes %>%
  pivot_longer(
    cols = item_1:item_3,  # variables item_1 to item_3
    names_to = "item_no",  # column names -> new column called item_no
    values_to = "item"  # values in columns -> new column called item
    )

compras
```

```
## # A tibble: 6 × 3
## id_cliente item_no item
        <dbl> <chr> <chr>
##
## 1
            1 item 1 pao
## 2
            1 item_2 leite
## 3
            1 item 3 banana
## 4
            2 item 1 leite
## 5
            2 item_2 papel higienico
## 6
            2 item_3
                     < NA>
```

Exemplo da importância de dados tidy

Várias operações de transformação requerem-no (e.g. *join* - prox. aula)

precos

```
compras %>%
  left_join(precos)
```

```
## # A tibble: 6 × 4
     id_cliente item_no item
                                       price
         <dbl> <chr>
                                       <dbl>
##
                       <chr>
              1 item_1 pao
                                         1.5
## 1
## 2
              1 item 2 leite
                                         0.5
## 3
              1 item 3
                       banana
## 4
             2 item 1 leite
## 5
             2 item 2
                       papel higienico
## 6
              2 item 3
                       <NA>
                                        NA
```

$compras \rightarrow clientes$

- data
- names_f rom: variável em formato long a transpor para nomes de novas colunas
- values_from:
 variável em formato
 long que contêm
 valores a dispersar por
 várias colunas no
 formato wide

```
compras %>%
  pivot_wider(
    names_from = item_no,
    values_from = item
)
```