reshaping data with `tidyr`

Emily Malcolm-White

The goal of `tidyr` is to help you create tidy data.

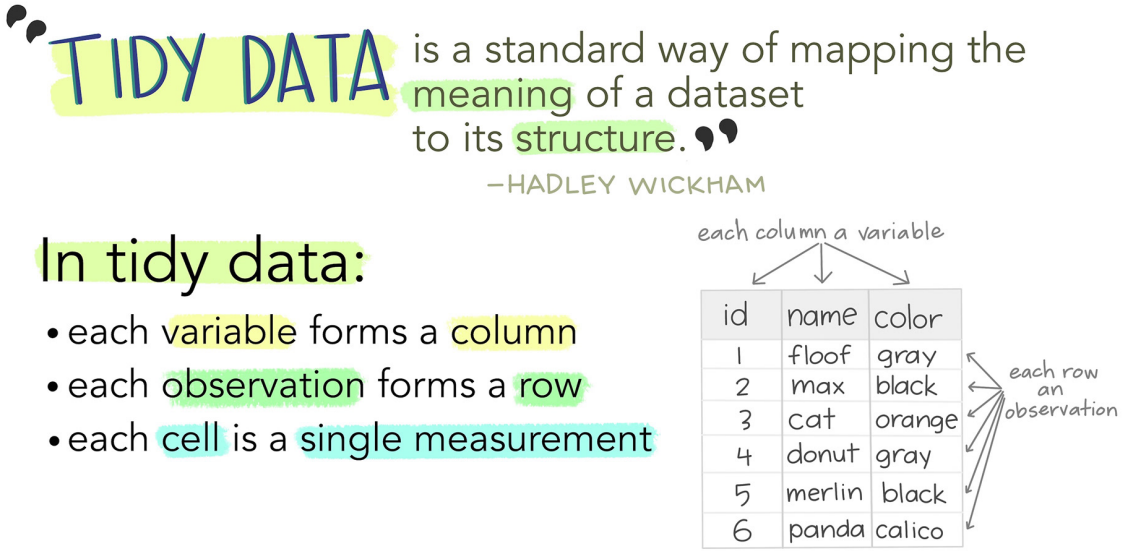
Reshaping with Pivoting – Why?

Data frames are often described as wide or long.

Wide when a row has more than one observation, and the units of observation are on one row each

Long when a row has only one observation, but the units of observation are repeated down the column

Credit: datasciencebook.ca



Wickham, H. (2014). Tidy Data. Journal of Statistical Software *59* (10). DOI: 10.18637/jss.v059.i10

Figure 1: Illustrations from the Openscapes blog Tidy Data for reproducibility, efficiency, and collaboration by Julia Lowndes and Allison Horst

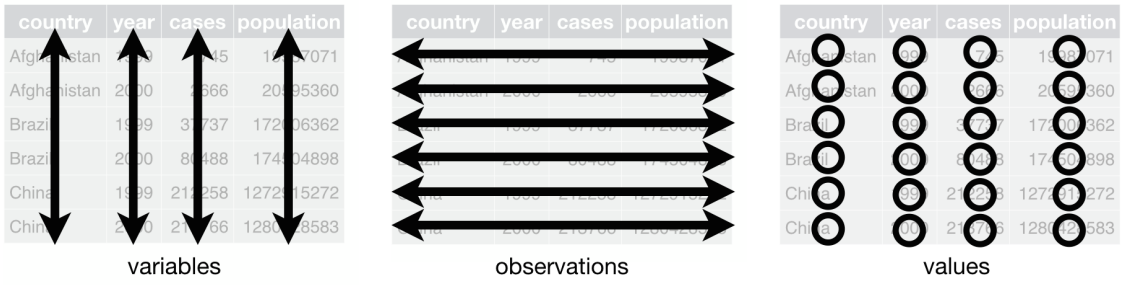


Figure 2: https://r4ds.hadley.nz/data-tidy

`canlang` dataset

```
#LOAD PACKAGES
library(tidyverse)

#LOAD DATA
lang_wide <- read.csv("https://raw.githubusercontent.com/UBC-DSCI/introduction-to-datascie
```

Pivot Longer

Credit: datasciencebook.ca

Credit: datasciencebook.ca

```
lang_mother_tidy <- pivot_longer(lang_wide,
  cols = Toronto:Edmonton,
  names_to = "region",
  values_to = "mother_tongue"
)

lang_mother_tidy
```

```
# A tibble: 1,070 x 4
   category                                language         region mother_tongue
   <chr>                                   <chr>            <chr>          <int>
 1 Aboriginal languages                    Aboriginal lang~ Toron~            80
 2 Aboriginal languages                    Aboriginal lang~ Montr~            30
 3 Aboriginal languages                    Aboriginal lang~ Vanco~            70
 4 Aboriginal languages                    Aboriginal lang~ Calga~            20
 5 Aboriginal languages                    Aboriginal lang~ Edmon~            25
 6 Non-Official & Non-Aboriginal languages Afrikaans        Toron~           985
 7 Non-Official & Non-Aboriginal languages Afrikaans        Montr~            90
 8 Non-Official & Non-Aboriginal languages Afrikaans        Vanco~          1435
 9 Non-Official & Non-Aboriginal languages Afrikaans        Calga~           960
10 Non-Official & Non-Aboriginal languages Afrikaans        Edmon~           575
# i 1,060 more rows
```

The data above is now tidy because all three criteria for tidy data have now been met:

- All the variables (category, language, region and mother_tongue) are now their own columns in the data frame.
- Each observation, (i.e., each language in a region) is in a single row.
- Each value is a single cell, i.e., its row, column position in the data frame is not shared with another value.

Pivot Wider

```
lang_long <- read.csv("https://raw.githubusercontent.com/UBC-DSCI/introduction-to-datascie
```

Credit: datasciencebook.ca

Credit: datasciencebook.ca

```
lang_home_tidy <- pivot_wider(lang_long,
  names_from = type,
  values_from = count
)
lang_home_tidy
```

```
# A tibble: 1,070 x 5
   region    category                          language most_at_home most_at_work
   <chr>     <chr>                             <chr>           <int>        <int>
 1 Montréal  Aboriginal languages               Aborigi~           15            0
 2 Toronto   Aboriginal languages               Aborigi~           50            0
 3 Calgary   Aboriginal languages               Aborigi~            5            0
 4 Edmonton  Aboriginal languages               Aborigi~           10            0
 5 Vancouver Aboriginal languages               Aborigi~           15            0
 6 Montréal  Non-Official & Non-Aboriginal l~ Afrikaa~           10            0
 7 Toronto   Non-Official & Non-Aboriginal l~ Afrikaa~          265            0
 8 Calgary   Non-Official & Non-Aboriginal l~ Afrikaa~          505           15
 9 Edmonton  Non-Official & Non-Aboriginal l~ Afrikaa~          300            0
10 Vancouver Non-Official & Non-Aboriginal l~ Afrikaa~          520           10
# i 1,060 more rows
```

gif

Gapminder

```
library(gapminder)
data("gapminder")
```

Let's say we'd like to look at `LifeExp` over time for all the countries in Asia in our dataset.

```
# Create a dataset called asia with the data we need
asia <- gapminder %>%
  filter(continent == "Asia") %>%
  select(country, year, lifeExp)
```

We can create a wide version of our table, where each row is a country and each column a year, with values of `lifeExp` in each cell of the table.

```
lifeExp_wide <- asia %>%
  pivot_wider(names_from = "year",  #<1>
              names_prefix = "yr", #<2>
              values_from = "lifeExp")
lifeExp_wide
```

① use pivot_wider to go from long to wide format
② Adds the pre-fix "yr" to all the column names – it's a good idea to avoid column names that start with a number.

```
# A tibble: 33 x 13
   country yr1952 yr1957 yr1962 yr1967 yr1972 yr1977 yr1982 yr1987 yr1992 yr1997
   <fct>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
 1 Afghan~   28.8   30.3   32.0   34.0   36.1   38.4   39.9   40.8   41.7   41.8
 2 Bahrain   50.9   53.8   56.9   59.9   63.3   65.6   69.1   70.8   72.6   73.9
 3 Bangla~   37.5   39.3   41.2   43.5   45.3   46.9   50.0   52.8   56.0   59.4
 4 Cambod~   39.4   41.4   43.4   45.4   40.3   31.2   51.0   53.9   55.8   56.5
 5 China     44     50.5   44.5   58.4   63.1   64.0   65.5   67.3   68.7   70.4
 6 Hong K~   61.0   64.8   67.6   70     72     73.6   75.4   76.2   77.6   80
 7 India     37.4   40.2   43.6   47.2   50.7   54.2   56.6   58.6   60.2   61.8
 8 Indone~   37.5   39.9   42.5   46.0   49.2   52.7   56.2   60.1   62.7   66.0
 9 Iran      44.9   47.2   49.3   52.5   55.2   57.7   59.6   63.0   65.7   68.0
10 Iraq      45.3   48.4   51.5   54.5   57.0   60.4   62.0   65.0   59.5   58.8
# i 23 more rows
# i 2 more variables: yr2002 <dbl>, yr2007 <dbl>
```

External Resources

- R for Data Science, Data Tidying