Worksheet 4 Tidy Data

The tidyverse package has several data sets built in. In this worksheet, we will look at several versions of the same data, set up in different formats with the aim of organising the tables into a specified ‘tidy’ format.

The example below shows the same data organised in different ways. Each dataset shows the same values of four variables country, year, population, and cases, but each dataset organises the values in a different way. The data sets are part of the tidyverse package, table 1 is organised correctly, just type:

table1

A tibble: 6 × 4

country year cases population

<chr> <int> <int> <int>

1 Afghanistan 1999 745 19987071

2 Afghanistan 2000 2666 20595360

3 Brazil 1999 37737 172006362

4 Brazil 2000 80488 174504898

5 China 1999 212258 1272915272

6 China 2000 213766 1280428583

**gather() gather makes wide tables longer and narrower**

A common problem is a dataset where some of the column names are not names of variables, but values of a variable. For table 4a, he numbers in the table represent a variable called cases

table4a

A tibble: 3 × 3

country `1999` `2000`

<chr> <int> <int>

1 Afghanistan 745 2666

2 Brazil 37737 80488

3 China 212258 213766

Note that the column names 1999 and 2000 represent values of the year variable, and each row represents two observations, not one.

To tidy a dataset like this, we need to gather those columns into a new pair of variables. To describe that operation we need three parameters:

* The set of columns that represent values, not variables. In this example, those are the columns 1999 and 2000.
* The name of the variable whose values form the column names. I call that the *key*, and here it is year.
* The name of the variable whose values are spread over the cells. I call that *value*, and here it’s the number of cases.

Together those parameters generate the call to gather():

table4a %>%

gather(`1999`, `2000`, key = "year", value = "cases")

A tibble: 6 × 3

country year cases

<chr> <chr> <int>

1 Afghanistan 1999 745

2 Brazil 1999 37737

3 China 1999 212258

4 Afghanistan 2000 2666

5 Brazil 2000 80488

6 China 2000 213766

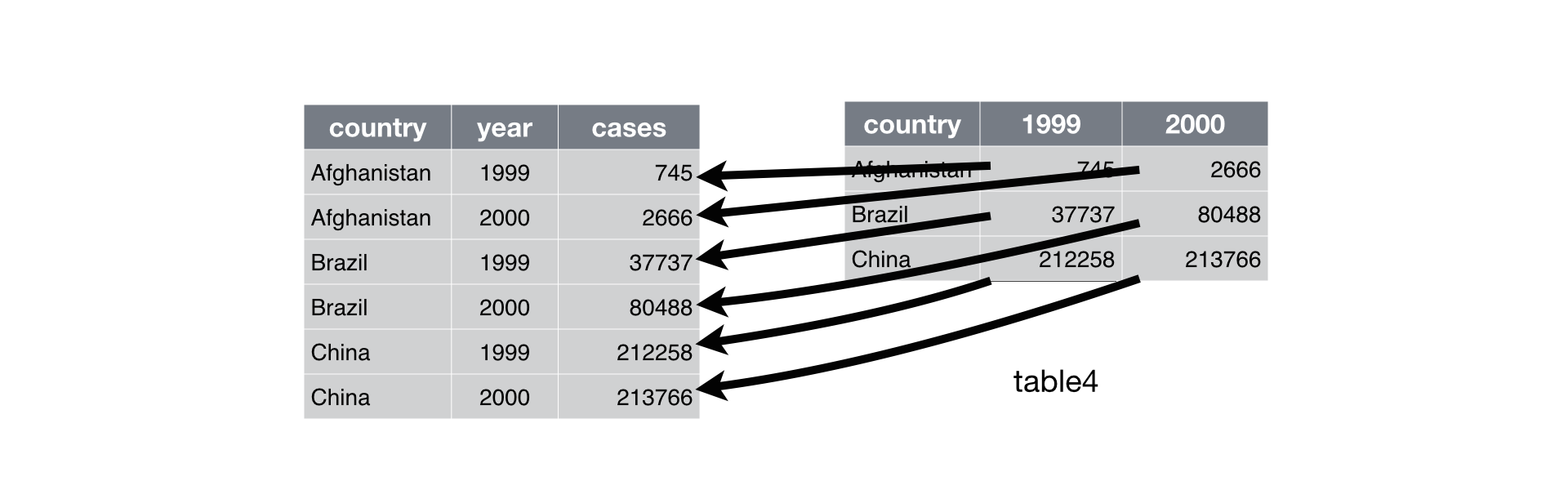
Looking at the command

gather(`1999`, `2000`, key = "year", value = "cases")

Note that:

* since there are only two columns to gather (1999 and 2000) we list the columns individually.
* “1999” and “2000” are non-syntactic names (because they don’t start with a letter) so we have to surround them in backticks. To refresh your memory of the other ways to select columns, see select()

In the final result, the gathered columns are dropped, and we get new key and value columns. Otherwise, the relationships between the original variables are preserved. Visually, this is shown in the figure below.



We can use gather() to tidy table4b in a similar fashion. The only difference is the variable stored in the cell values (this time it is population).

**Exercise:** use the gather() command to put table4b into the correct format.

To combine the tidied versions of table4a and table4b into a single tibble, we need to use left\_join()

tidy4a <- table4a %>%

gather(`1999`, `2000`, key = "year", value = "cases")

tidy4b <- table4b %>%

gather(`1999`, `2000`, key = "year", value = "population")

left\_join(tidy4a, tidy4b)

The commands above produce the following output:

Joining, by = c("country", "year")

A tibble: 6 × 4

country year cases population

<chr> <chr> <int> <int>

1 Afghanistan 1999 745 19987071

2 Brazil 1999 37737 172006362

3 China 1999 212258 1272915272

4 Afghanistan 2000 2666 20595360

5 Brazil 2000 80488 174504898

6 China 2000 213766 1280428583

**spread() makes long tables shorter and wider**

Spreading is the opposite of gathering. You use it when an observation is scattered across multiple rows. For example, take table2: an observation is a country in a year, but each observation is spread across two rows.

table2

A tibble: 12 × 4

country year type count

<chr> <int> <chr> <int>

1 Afghanistan 1999 cases 745

2 Afghanistan 1999 population 19987071

3 Afghanistan 2000 cases 2666

4 Afghanistan 2000 population 20595360

5 Brazil 1999 cases 37737

6 Brazil 1999 population 172006362

... with 6 more rows

To tidy this up, we first analyse the representation in similar way to gather(). This time, however, we only need two parameters:

* the column that contains variable names, the *key* column. Here, it’s type.
* the column that contains values forms multiple variables, the *value* column. Here it’s count.

Once we’ve figured that out, we can use spread(), as shown below.

spread(table2, key = type, value = count)

A tibble: 6 × 4

country year cases population

\* <chr> <int> <int> <int>

1 Afghanistan 1999 745 19987071

2 Afghanistan 2000 2666 20595360

3 Brazil 1999 37737 172006362

4 Brazil 2000 80488 174504898

5 China 1999 212258 1272915272

6 China 2000 213766 1280428583

