

Data reshaping

The gather function



These slides are licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 4.0
International License.

Structure of the `gather` function

`gather` takes columns and turns them into rows.

```
name_of_dataset %>%  
  gather(  
    column_one,  
    column_two,  
    key = "category",  
    value = "value"  
)
```

```
name_of_dataset %>%  
  gather(  
    column_one:column_three,  
    key = "category",  
    value = "value"  
)
```

- The first set of inputs are the columns you wish to turn into rows.
- The `key =` input is the name you will give to the new column that will contain the names of the gathered columns.
- The `value =` input is the name you will give to the new column that will contain the values of the gathered columns.

Untidy data example

Problem: Untidy data frame stored in `table4a` and `table4b`

table4a

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

table4b

country	1999	2000
Afghanistan	19987071	20595360
Brazil	172006362	174504898
China	1272915272	1280428583

Untidy data example

Goal: Use `gather` to transform `table4a` and `table4b` back to this:

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

gather schematic

gather takes columns and turns them into rows.

country	year	cases	country	1999	2000
Afghanistan	1999	745	Afghanistan	745	2666
Afghanistan	2000	2666	Brazil	37737	80488
Brazil	1999	37737	China	212258	213766
Brazil	2000	80488			
China	1999	212258			
China	2000	213766			

table4

The diagram illustrates the 'gather' operation by showing two tables side-by-side. The first table, on the left, has columns for 'country', 'year', and 'cases'. The second table, on the right, has columns for 'country', '1999', and '2000'. Arrows point from the 'cases' column of the first table to the '1999' and '2000' columns of the second table, indicating that the 'cases' values are being moved into new columns based on the 'year'.

Source: Figure 12.2 in *R for Data Science* by Garrett Grolemund and Hadley Wickham.

gather example

```
tidy4a <- table4a %>%  
  gather(  
    `1999`:`2000`,  
    key = "year",  
    value = "cases"  
)
```

```
tidy4b <- table4b %>%  
  gather(  
    `1999`:`2000`,  
    key = "year",  
    value = "population"  
)
```

gather example

```
tidy4a <- table4a %>%
  gather(
    `1999`:`2000`,
    key = "year",
    value = "cases"
  )
```

country	year	cases
Afghanistan	1999	745
Brazil	1999	37737
China	1999	212258
Afghanistan	2000	2666
Brazil	2000	80488
China	2000	213766

```
tidy4b <- table4b %>%
  gather(
    `1999`:`2000`,
    key = "year",
    value = "population"
  )
```

country	year	population
Afghanistan	1999	19987071
Brazil	1999	172006362
China	1999	1272915272
Afghanistan	2000	20595360
Brazil	2000	174504898
China	2000	1280428583

gather example

To fully restore `table1`, we use the `left_join` function from the `dplyr` package:

```
left_join(tidy4a, tidy4b)
```

country	year	cases	population
Afghanistan	1999	745	19987071
Brazil	1999	37737	172006362
China	1999	212258	1272915272
Afghanistan	2000	2666	20595360
Brazil	2000	80488	174504898
China	2000	213766	1280428583

Credits

License

Creative Commons Attribution-NonCommerical-ShareAlike 4.0 International

Acknowledgments

Content adapted from *R for Data Science* by Garrett Grolemund and Hadley Wickham, [chapter 12](#), made available under the CC BY-NC-ND 3.0 license.