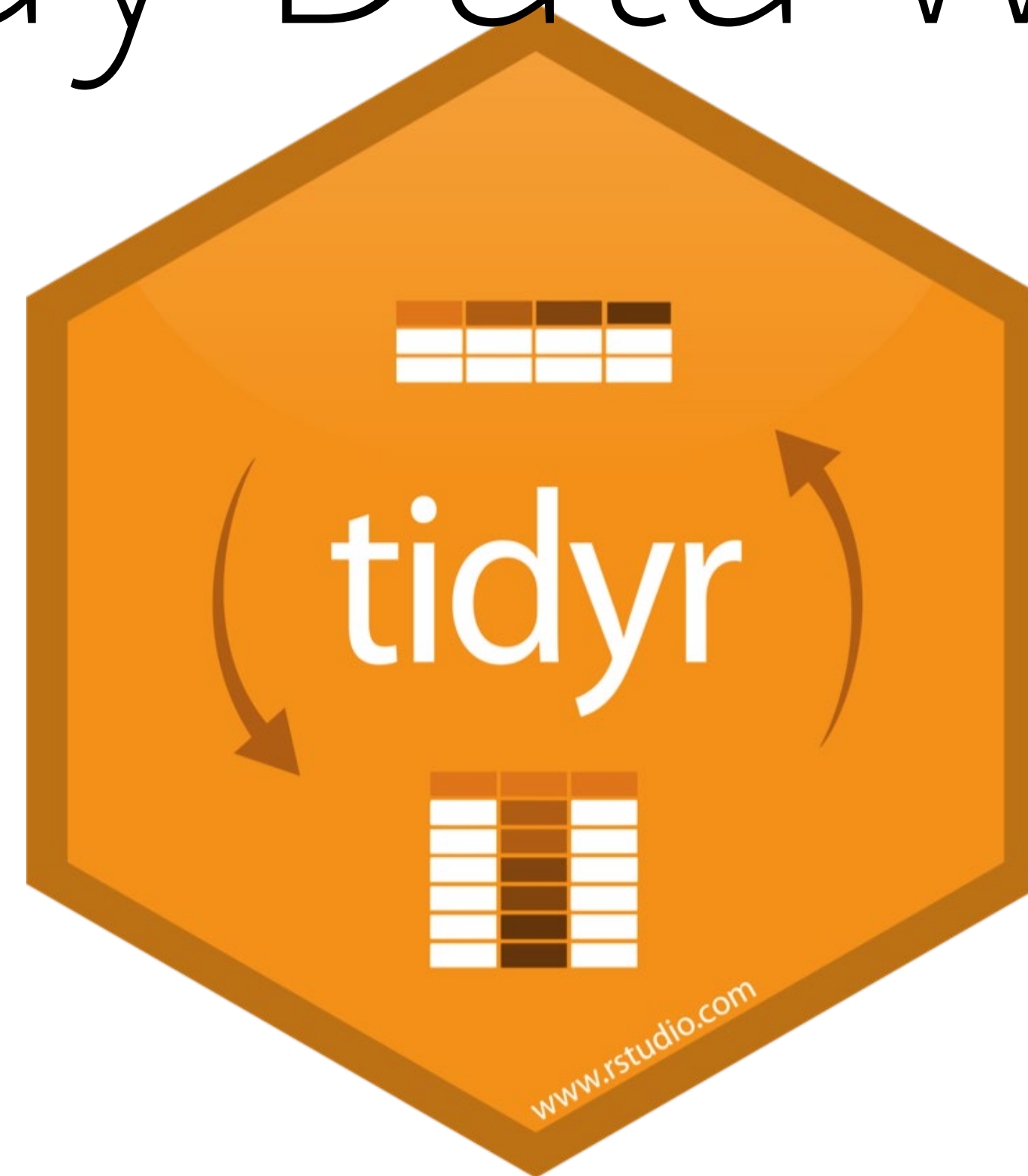


Tidy Data with



"Data comes in many formats, but R prefers just one: tidy data. "

- Garrett Grolmund

Tidy data

country	year	cases	pop
Afghanistan	1999	745	19937631
Afghanistan	2000	666	20105630
Afghanistan	2001	703	20105630
Afghanistan	2002	733	20105630
Afghanistan	2003	728	20105630
Afghanistan	2004	700	20105630

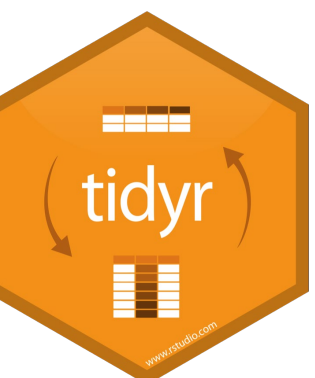
A data set is **tidy** iff:

1. Each **variable** is in its own **column**
2. Each **observation** is in its own **row**
3. Each **value** is in its own **cell**

Also see these papers, in your “other resources” folder:

Wickham, 2014: *Tidy Data*

Broman and Woo, 2017: *Data Organization in Spreadsheets*



Quiz

What are the variables in this data set?

table1

country <chr>	year <int>	cases <int>	population <int>
Afghanistan	1999	45	19987071
Afghanistan	2000	66	20505360
Brazil	1999	37	172006362
Brazil	2000	80	174504898
China	1999	212	1272915272
China	2000	213	128048583

6 rows

Tidy data

country	year	cases	pop
Afghanistan	1999	745	19937171
Afghanistan	2000	689	20195920
Afghanistan	1999	7737	17999922
Afghanistan	2000	7737	17999922
Afghanistan	2000	7737	17999922
Afghanistan	2000	7737	17999922
Afghanistan	2000	7737	17999922
Afghanistan	2000	7737	17999922
Afghanistan	2000	7737	17999922
Afghanistan	2000	7737	17999922

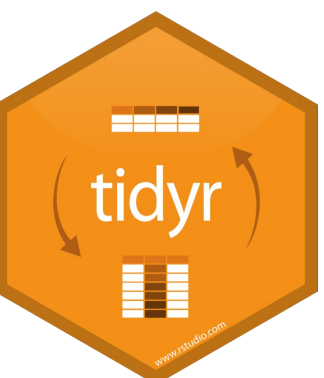
A data set is **tidy** iff:

1. Each **variable** is in its own **column**
2. Each **observation** is in its own **row**
3. Each **value** is in its own **cell**

variable: all values that measure the same underlying *attribute*

observation: all values measured on the same *unit*

value: belongs to one **variable** and one **observation**



Quiz

What are the variables in this data set?

table2

country <chr>	year <int>	type <chr>	count <int>
Afghanistan	1999	cases	745
Afghanistan	1999	population	1998701
Afghanistan	2000	cases	2666
Afghanistan	2000	population	2059530
Brazil	1999	cases	7737
Brazil	1999	population	17200632
Brazil	2000	cases	9488
Brazil	2000	population	17450488
China	1999	cases	2258
China	1999	population	12791522

1-10 of 12 rows


Previous 1 2 Next

Tidy data

country	year	cases	pop
Afghanistan	1999	745	19937631
Afghanistan	2000	666	20105520
Afghanistan	2001	703	20105520
Afghanistan	2002	733	20105520
Afghanistan	2003	2238	20105520
Afghanistan	2004	5700	20105520

A data set is **tidy** iff:

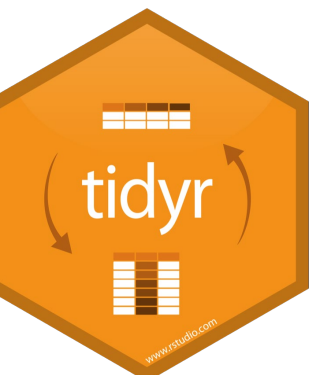
1. Each **variable** is in its own **column**
2. Each **observation** is in its own **row**
3. Each **value** is in its own **cell**



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

6 rows

```
table1$country
table1$year
table1$cases
table1$population
```



📄 ⬆ ✕

country <chr>	year <int>	type <chr>	count <int>
Afghanistan	1999	cases	745
Afghanistan	1999	deaths	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	deaths	95360
Brazil	2000	cases	737
Brazil	2000	deaths	62
Brazil	2001	cases	88
Brazil	2001	deaths	8
China	2001	cases	58
China	2001	deaths	72

1–10 of 12 rows

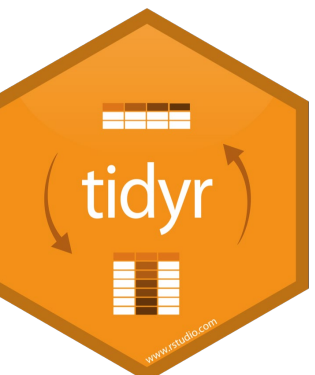
Previous 1 2 Next

```
table2$count[table2$year == 1999]
table2$year[table2$type == "cases"]
table2$count[c(1,3,5,7,9,11)]
table2$count[c(2,4,6,8,10,12)]
```


country <chr>	year <int>	cases <int>	population <int>	rate <dbl>
Afghanistan	1999	745	19987071	0.0000372741
Afghanistan	2000	2666	20595360	0.0001294466
Brazil	1999	37737	172006362	0.0002193930
Brazil	2000	80488	174504898	0.0004612363
China	1999	212258	1272915272	0.0001667495
China	2000	213766	1280428583	0.0001669488

6 rows

```
table1$cases / table1$population -> table1$rate
```



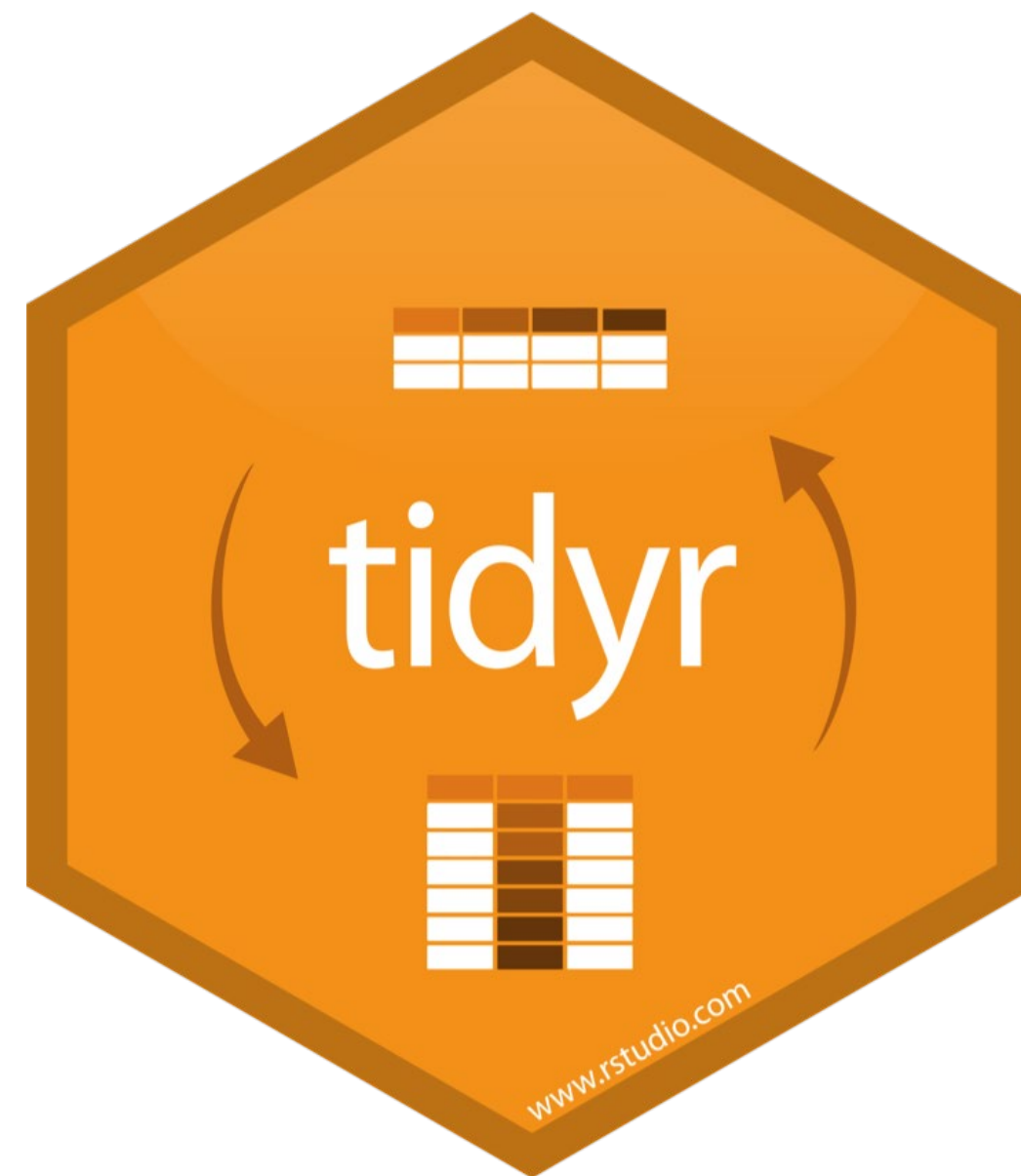
"Tidy data sets are all alike; but every messy data set is messy in its own way."

- Hadley Wickham

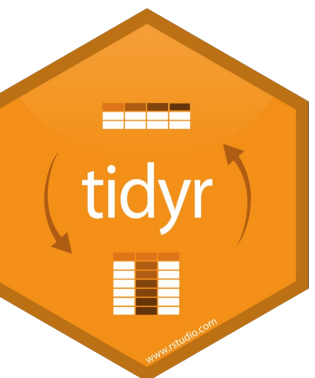
tidyr



tidyr



A package that reshapes the layout of tabular data.



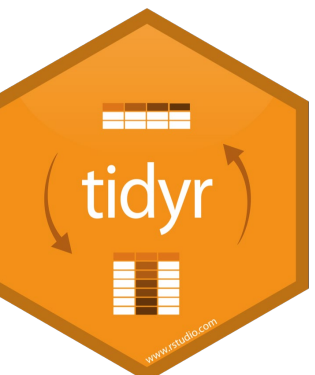
pivot_longer()



Toy data

```
03-Tidy-Data.Rmd x
1 |---
2 title: "Tidy Data"
3 output: html_notebook
4 ---
5
6 ```{r setup}
7 library(tidyverse)
8 library(babynames)
9
10 # Toy data
11 cases <- tribble(
12   ~Country, ~"2011", ~"2012", ~"2013",
13   "FR",      7000,    6900,    7000,
14   "DE",      5800,    6000,    6200,
15   "US",     15000,   14000,    13000,
16 )
17
18 pollution <- tribble(
19   ~city, ~size, ~amount,
20   "New York", "large", 23,
21   "New York", "small", 14,
22   "London", "large", 22,
23   "London", "small", 16,
24   "Beijing", "large", 121,
25   "Beijing", "small", 121,
26 )
27
28 x <- tribble(
29   ~x1, ~x2,
30   "A", 1,
31   "B", NA,
32   "C", NA,
33   "D", 3,
34   "E", NA,
35 )
```

```
cases <- tribble(
  ~Country, ~"2011", ~"2012", ~"2013",
  "FR",      7000,    6900,    7000,
  "DE",      5800,    6000,    6200,
  "US",     15000,   14000,    13000
)
```



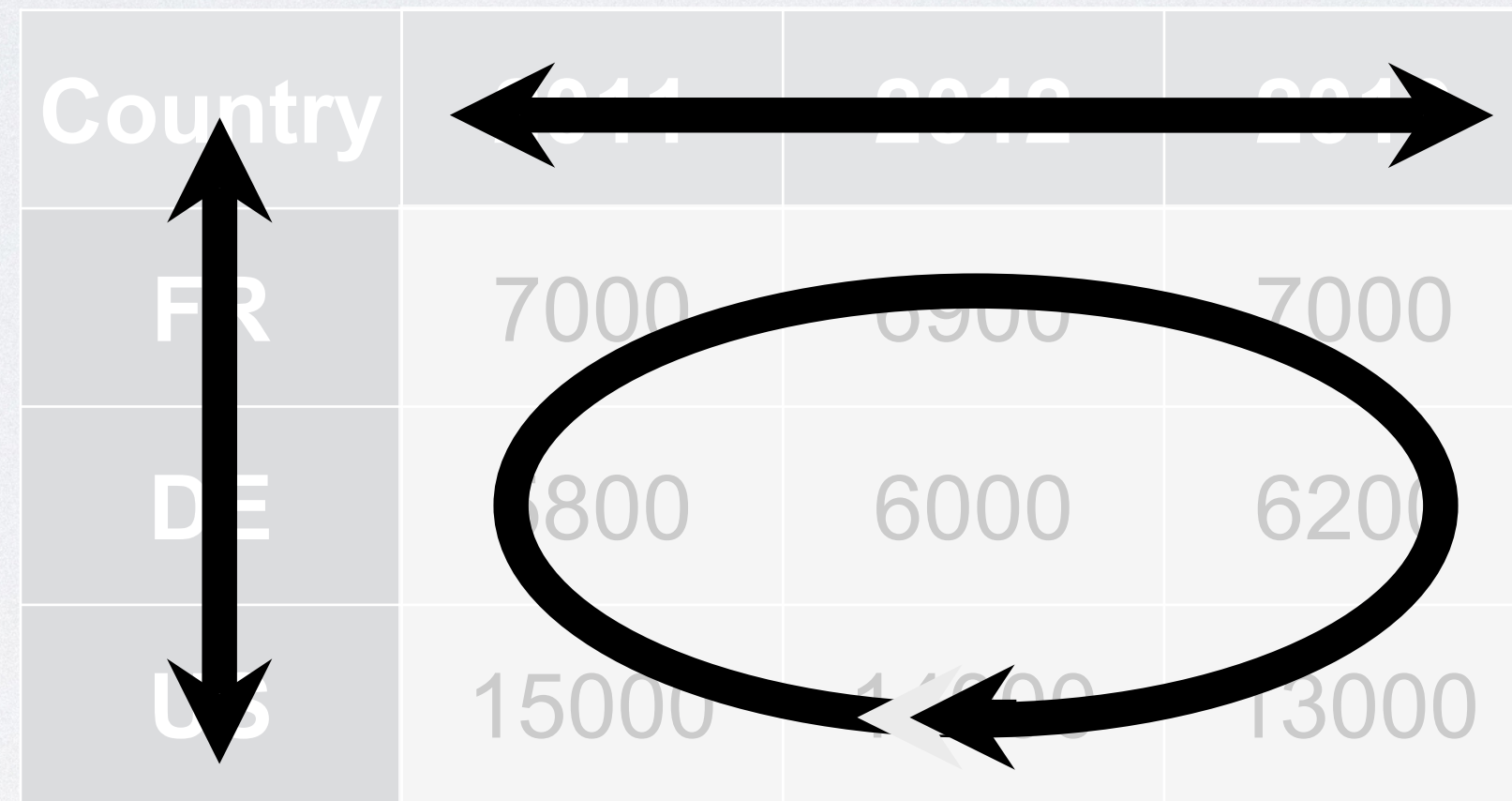
Quiz

What are the variables in cases?

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Quiz

What are the variables in cases?



The diagram shows a data table with three columns and four rows. The first column is labeled 'Country' and contains the values 'FR', 'DE', and 'US'. The second column contains the values '7000', '800', and '15000'. The third column contains the values '6900', '6000', and '11000'. The fourth column contains the values '7000', '6200', and '13000'. A vertical double-headed arrow is positioned to the left of the 'Country' column, spanning the rows for 'FR', 'DE', and 'US'. A horizontal double-headed arrow is positioned above the second and third columns, spanning the rows for 'FR', 'DE', and 'US'. An oval is drawn around the data cells in the second and third columns for the 'FR', 'DE', and 'US' rows.

Country			
FR	7000	6900	7000
DE	800	6000	6200
US	15000	11000	13000

- Country
- Year
- Count

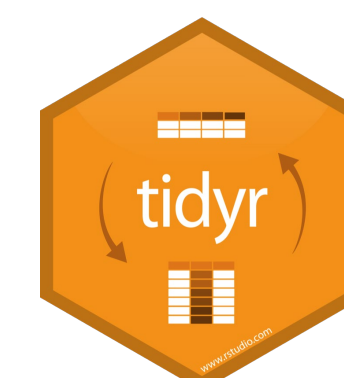
Your Turn 1

On a sheet of paper, draw how the cases data set would look if it had the same values grouped into three columns: *country*, *year*, *n*

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

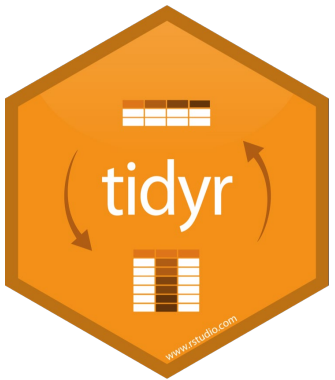
04:00

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000



Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
---------	------	---



Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

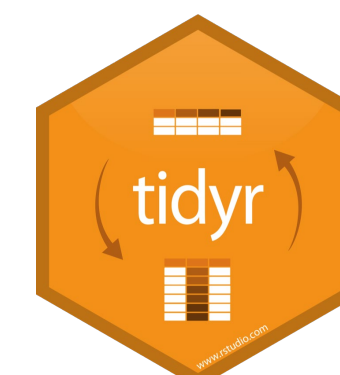
Country	Year	n
FR	2011	7000

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000



Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

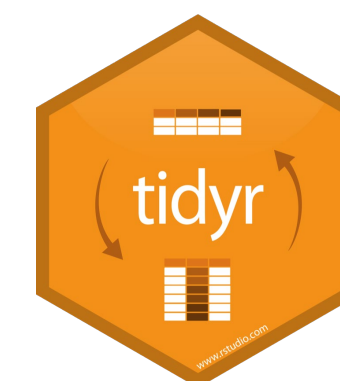
Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000

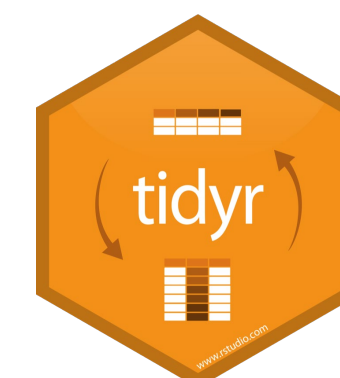
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000



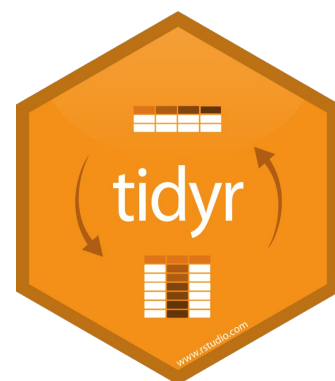
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000



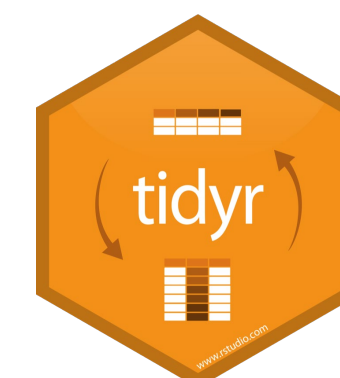
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200



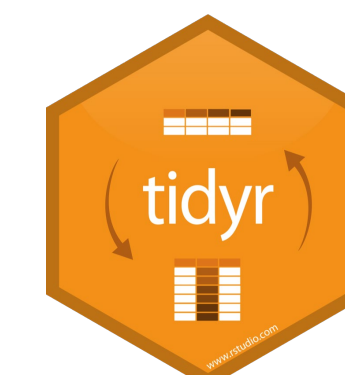
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

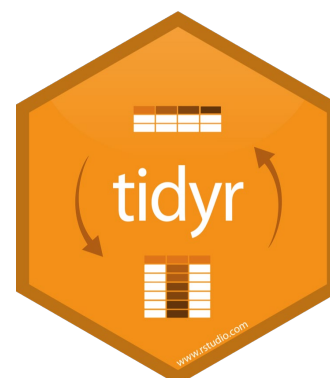
Country	Year	Revenue
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

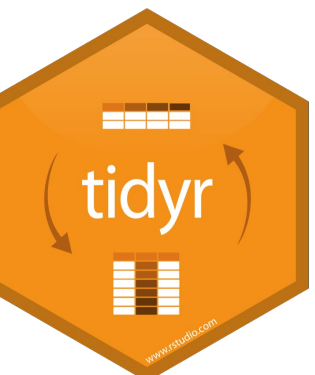
`pivot_longer()`

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

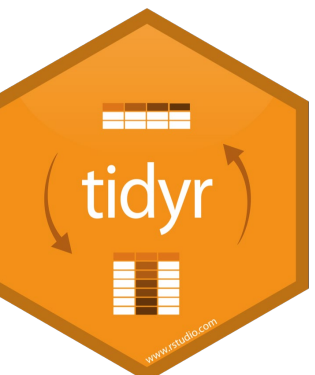
	1	2
Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



names_to (former column names)

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

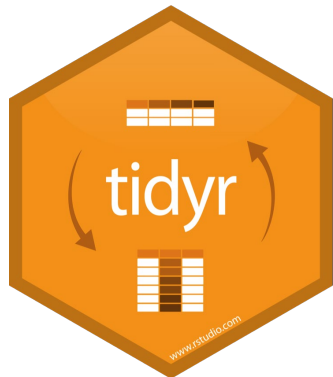
Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



values_to (former cells)

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



pivot_longer()

```
cases %>% pivot_longer(cols = 2:4, names_to = "year", values_to = "n")
```

**data frame to
reshape**

**numeric
indices of
columns to
collapse
(or names)**

**name of the
new key
column
(a character
string)**

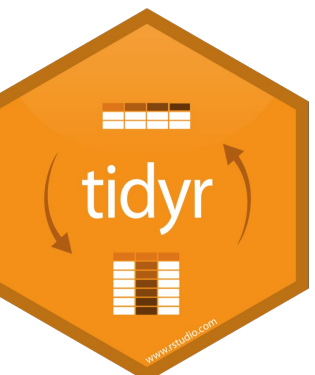
**name of the
new value
column
(a character
string)**

pivot_longer()

```
cases %>% pivot_longer(2:4, "year", "n")
```

numeric
indices

Country <chr>	2 2011 <dbl>	3 2012 <dbl>	4 2013 <dbl>
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

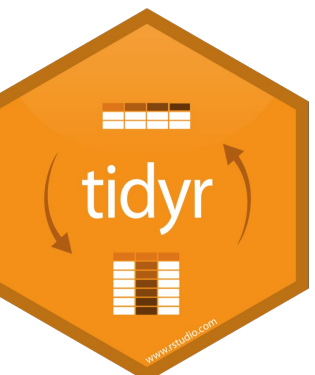


pivot_longer()

```
cases %>% pivot_longer(c("2011", "2012", "2013"), "year", "n")
```

names

	2011	2012	2013
Country <chr>	2011 <dbl>	2012 <dbl>	2013 <dbl>
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000



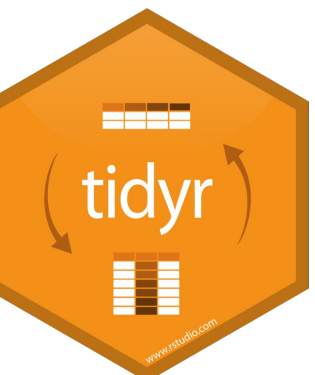
pivot_longer()

```
cases %>% pivot_longer(-Country, "year", "n")
```

Everything
except...

Not Country Not Country Not Country

Country <chr>	2011 <dbl>	2012 <dbl>	2013 <dbl>
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000



Your Turn 2

Use **pivot_longer()** to reorganize **table4a** into three columns: *country*, *year*, and *cases*.

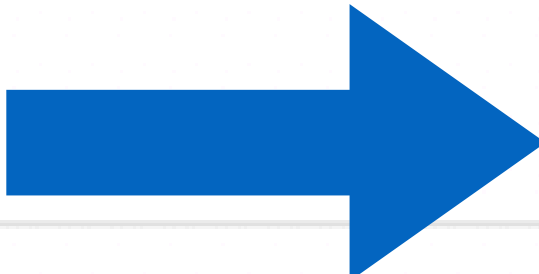
	country <chr>	1999 <int>	2000 <int>
1	Afghanistan	745	2666
2	Brazil	37737	80488
3	China	212258	213766

3 rows

03:00

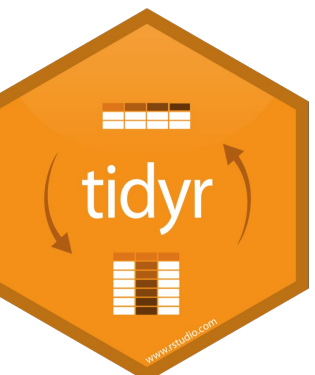

```
table4a %>%
```

```
  pivot_longer(cols = 2:3, names_to = "year", values_to = "n")
```



country <chr>	year <chr>	n <int>
Afghanistan	1999	745
Brazil	1999	37737
China	1999	212258
Afghanistan	2000	2666
Brazil	2000	80488
China	2000	213766

6 rows



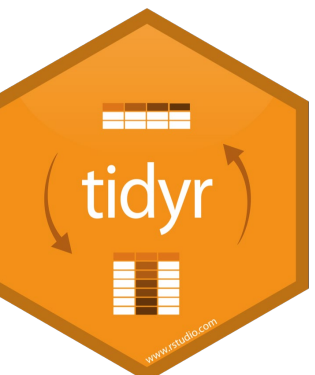
pivot_wider()



Toy data

```
03-Tidy-Data.Rmd
1 |---
2 title: "Tidy Data"
3 output: html_notebook
4 |---
5
6 ```{r setup}
7 library(tidyverse)
8 library(babynames)
9
10 # Toy data
11 cases <- tribble(
12   ~Country, ~"2011", ~
13     "FR", 7000,
14     "DE", 5800,
15     "US", 15000,
16 )
17
18 pollution <- tribble(
19   ~city, ~size, ~
20     "New York", "large",
21     "New York", "small",
22     "London", "large",
23     "London", "small",
24     "Beijing", "large",
25     "Beijing", "small",
26 )
27
28 x <- tribble(
29   ~x1, ~x2,
30     "A", 1,
31     "B", NA,
32     "C", NA,
33     "D", 3,
34     "E", NA
35 )
```

```
pollution <- tribble(
  ~city, ~size, ~amount,
  "New York", "large", 23,
  "New York", "small", 14,
  "London", "large", 22,
  "London", "small", 16,
  "Beijing", "large", 121,
  "Beijing", "small", 56
)
```



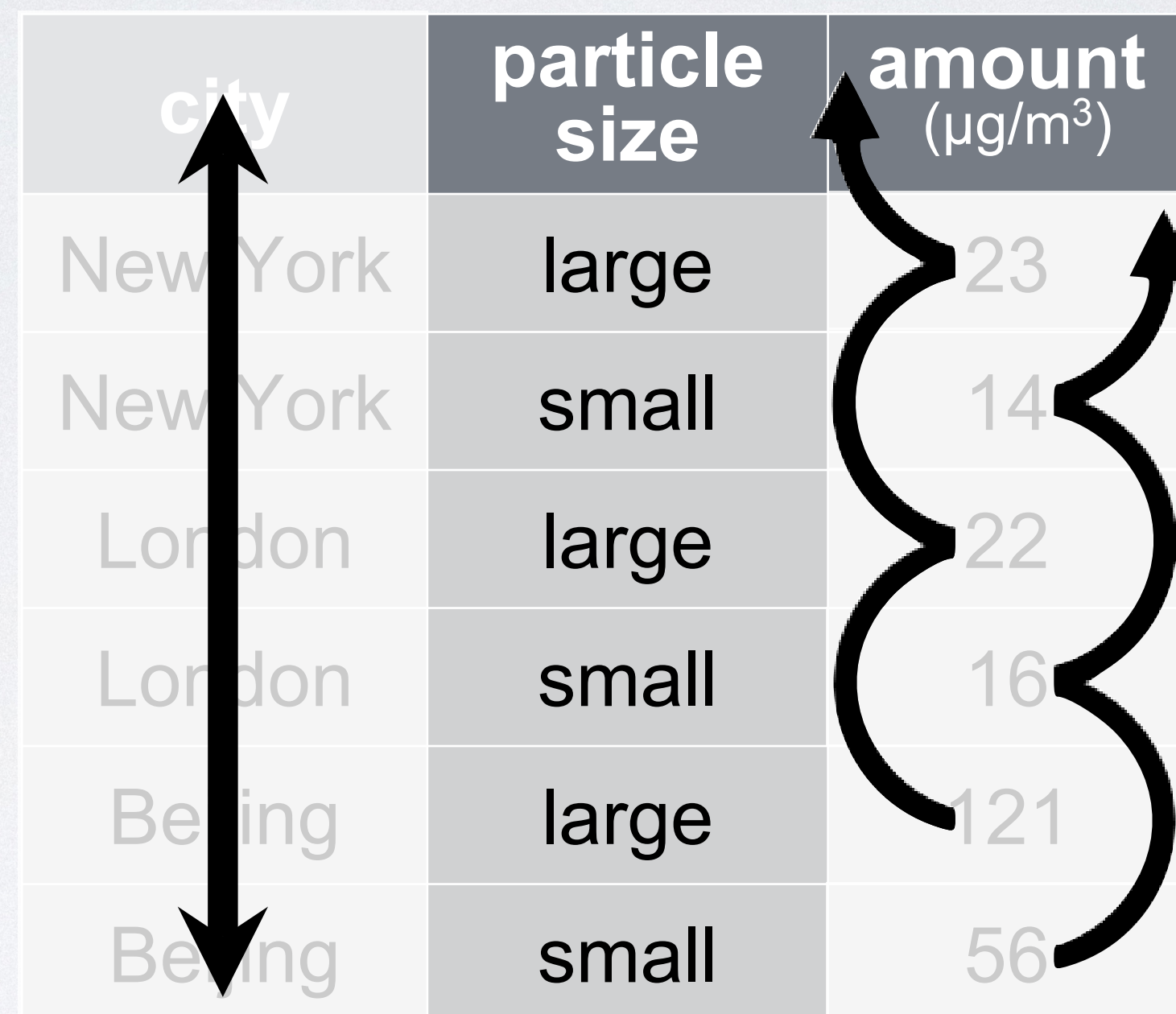
Quiz

What are the variables in pollution?

city	particle size	amount ($\mu\text{g}/\text{m}^3$)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

Quiz

What are the variables in pollution?



city	particle size	amount (µg/m³)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

- City
- Amount of large particulate
- Amount of small particulate

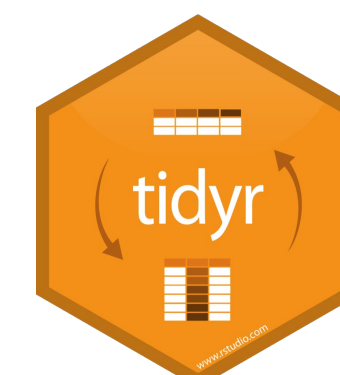
Your Turn 3

On a sheet of paper, draw how this data set would look if it had the same values grouped into three columns: *city*, *large*, *small*

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

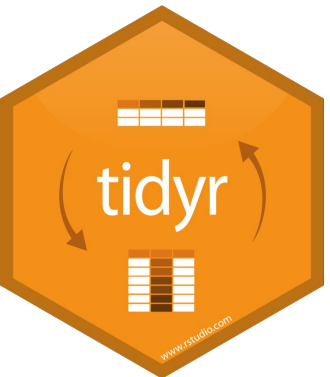
03:00

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56



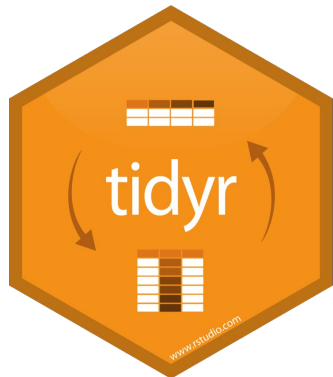
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
------	-------	-------



city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	

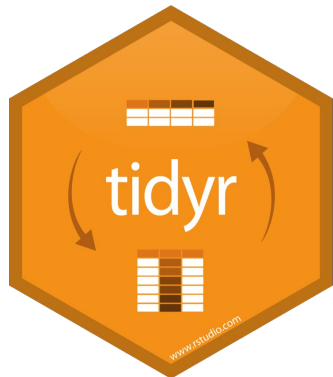


city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	



city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	16

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

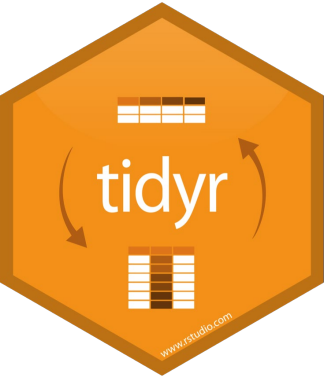
city	large	small
New York	23	14
London	22	16
Beijing	121	

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	16
Beijing	121	56

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	16
Beijing	121	56



city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

`pivot_wider()`

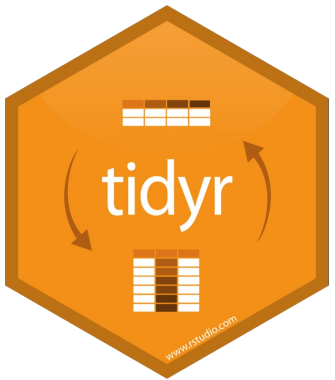
city	large	small
New York	23	14
London	22	16
Beijing	121	56

1

2

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

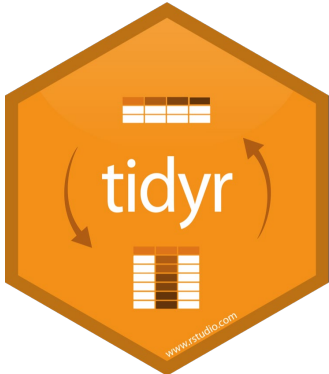
city	large	small
New York	23	14
London	22	16
Beijing	121	56



names_from (new column names)

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	16
Beijing	121	56



values_from (new cells)

city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	16
Beijing	121	56

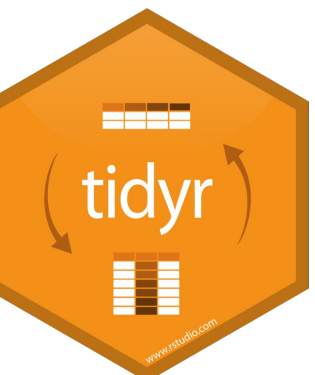
pivot_wider()

```
pollution %>% pivot_wider(names_from = size, values_from = amount)
```

**data frame to
reshape**

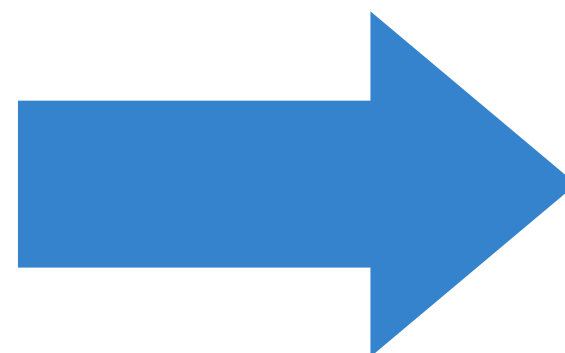
column to use for keys
(becomes new
column names)

column to use for values
(becomes new
column cells)




```
pollution %>% pivot_wider(names_from = size, values_from = amount)
```

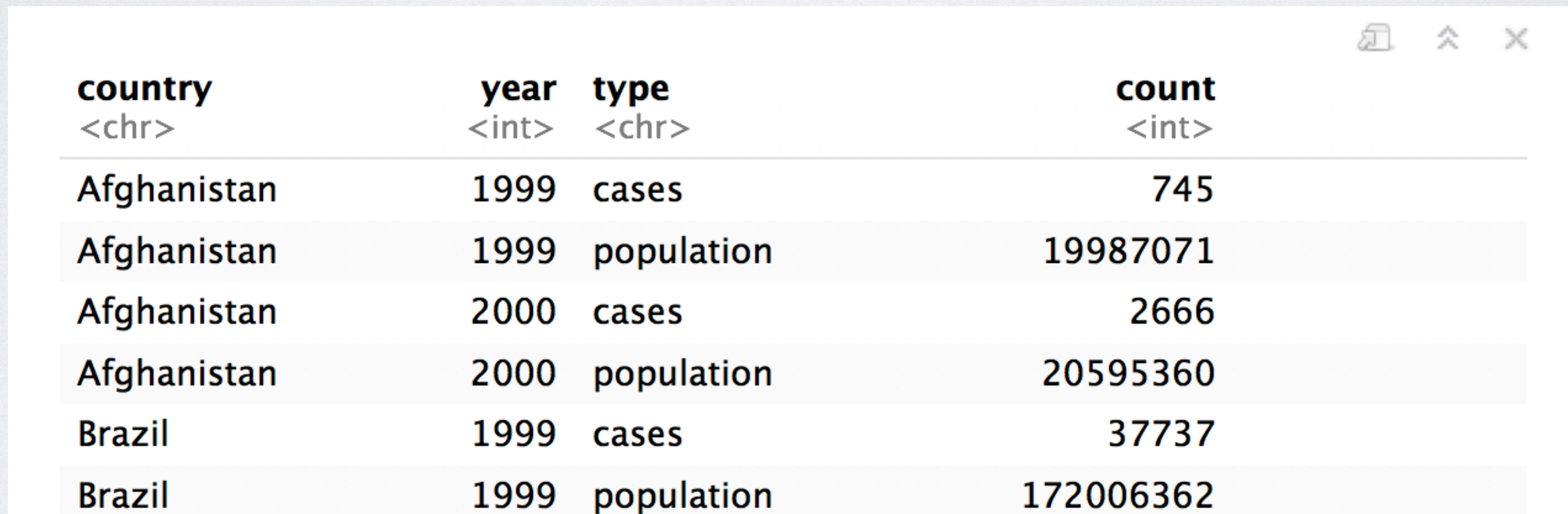
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56



city	large	small
New York	23	14
London	22	16
Beijing	121	56

Your Turn 4

Use **pivot_wider()** to reorganize **table2** into four columns: *country*, *year*, *cases*, and *population*.


A screenshot of an RStudio console window. The window has a title bar with a copy icon, an up arrow icon, and a close icon. Inside the window, a data table is displayed with four columns: 'country', 'year', 'type', and 'count'. The 'country' column has a data type of '<chr>', 'year' is '<int>', 'type' is '<chr>', and 'count' is '<int>'. The table contains six rows of data for Afghanistan and Brazil in 1999 and 2000, with 'cases' and 'population' types.

country <chr>	year <int>	type <chr>	count <int>
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362

03:00

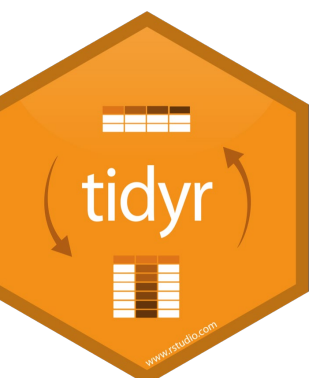

```
table2 %>%
```

```
  pivot_wider(names_from = type, values_from = count)
```



	country <chr>	year <int>	cases <int>	population <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

6 rows



Tidy Data with

