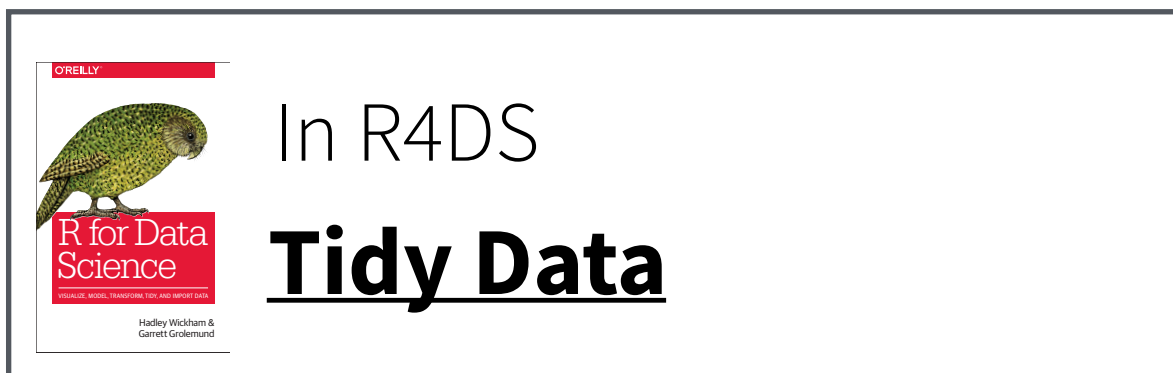
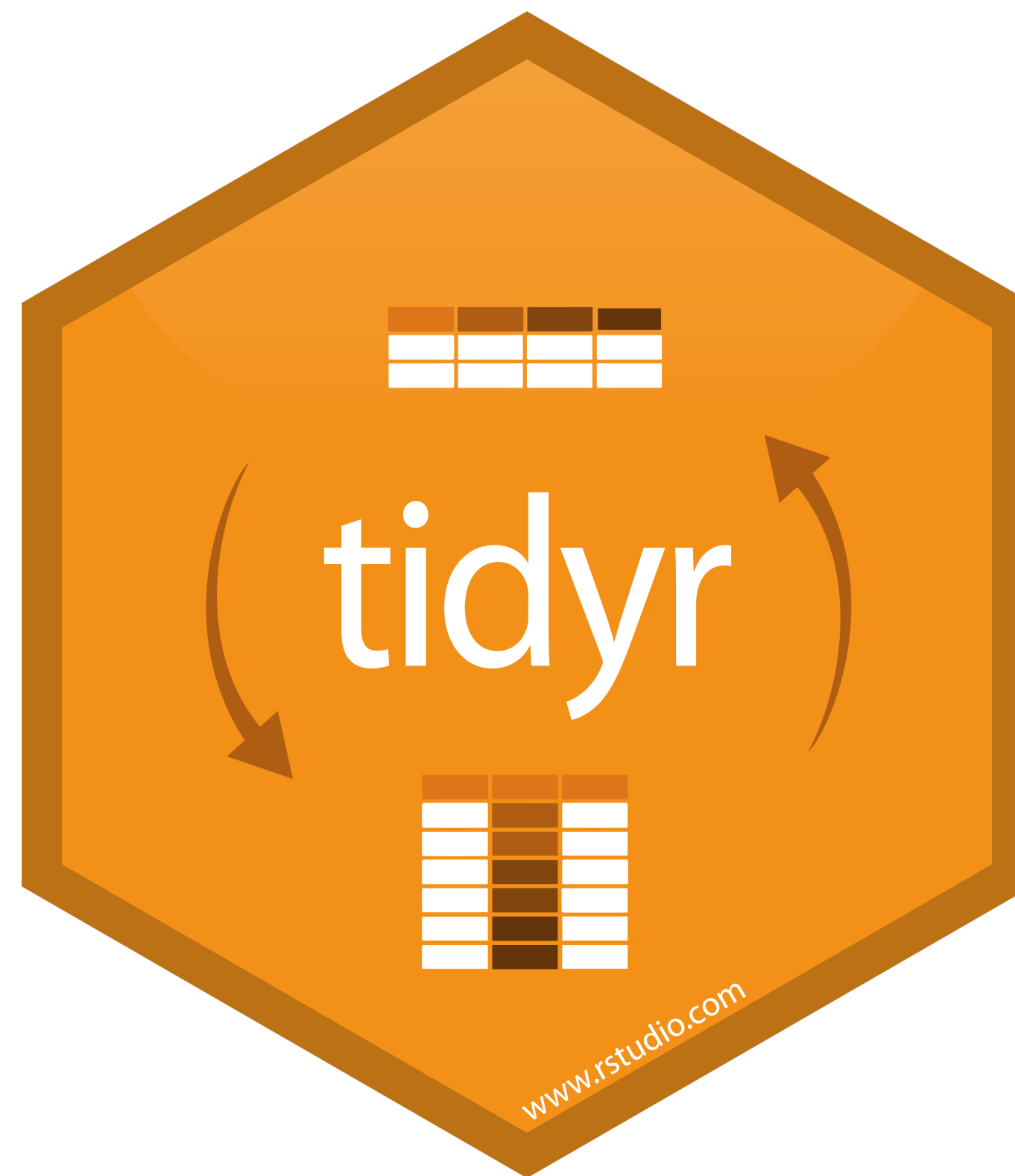


Tidy Data with



Your Turn 0

1. Open 03-Tidy.Rmd
2. Run the setup chunk

03:00

Tidy Data

country	year	cases	pop
Afghanistan	1999	745	10137321
Afghanistan	2000	666	20125120
Afghanistan	2001	165	22795522
Afghanistan	2002	155	23755333
Afghanistan	2003	220	24795272
Afghanistan	2004	220	25795272
Afghanistan	2005	220	26795272
Afghanistan	2006	220	27795272
Afghanistan	2007	220	28795272
Afghanistan	2008	220	29795272
Afghanistan	2009	220	30795272
Afghanistan	2010	220	31795272
Afghanistan	2011	220	32795272
Afghanistan	2012	220	33795272
Afghanistan	2013	220	34795272
Afghanistan	2014	220	35795272
Afghanistan	2015	220	36795272
Afghanistan	2016	220	37795272
Afghanistan	2017	220	38795272
Afghanistan	2018	220	39795272
Afghanistan	2019	220	40795272
Afghanistan	2020	220	41795272

A data set is **tidy** iff:

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

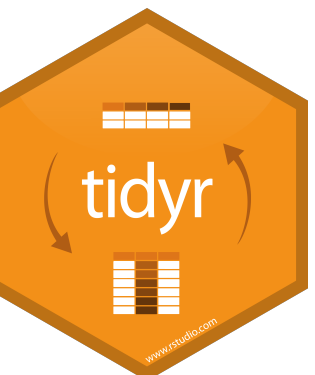
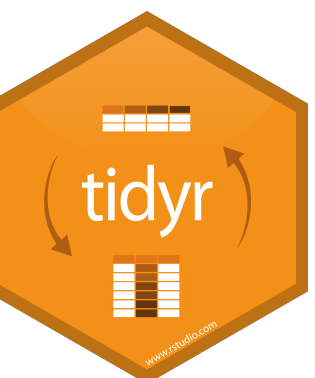


table1 is tidy

The data is a subset of the data contained in the
World Health Organization Global Tuberculosis
Report


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



Tidy data is easy to manipulate

The data is a subset of the data contained in the
World Health Organization Global Tuberculosis
Report



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 %>%  
  mutate(rate = cases/population)
```

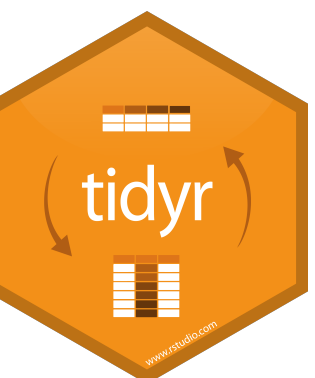


table2 isn't tidy

contains two
variables

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
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272

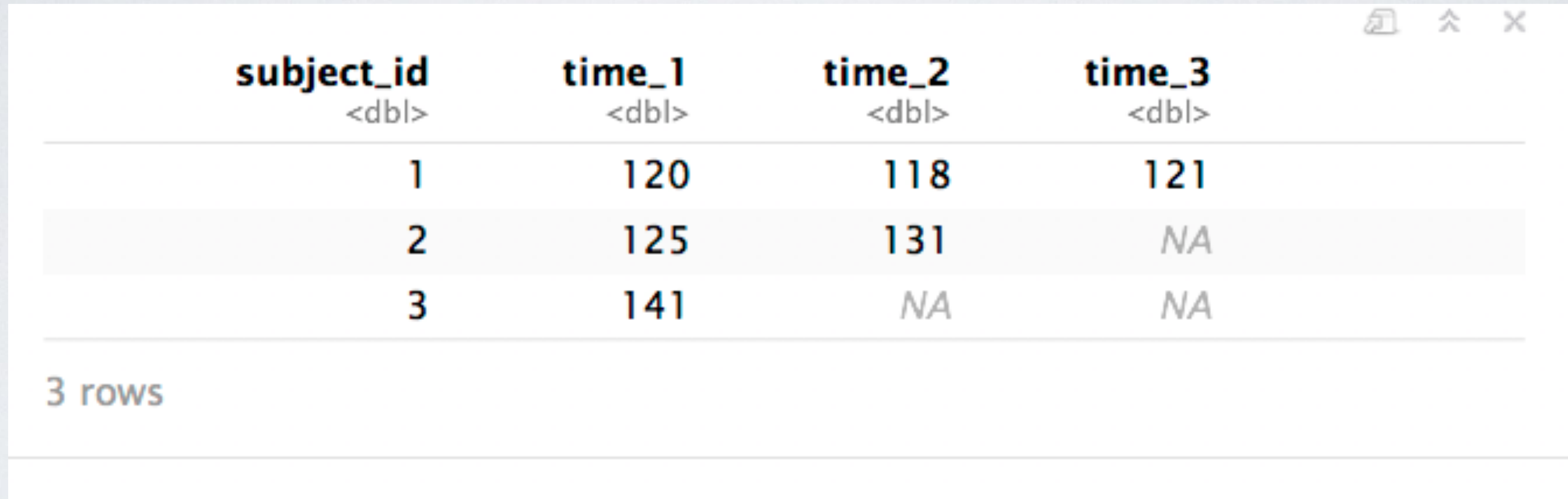
1-10 of 12 rows

Previous

It's hard to manipulate

Your Turn 1

Is bp_systolic tidy? What are the variables?



subject_id <dbl>	time_1 <dbl>	time_2 <dbl>	time_3 <dbl>
1	120	118	121
2	125	131	NA
3	141	NA	NA

3 rows

Your Turn 1

Is bp_systolic tidy? What are the variables?

subject_id <dbl>	time_1 <dbl>	time_2 <dbl>	time_3 <dbl>
1	120	118	121
2	125	131	NA
3	141	NA	NA
3 rows			

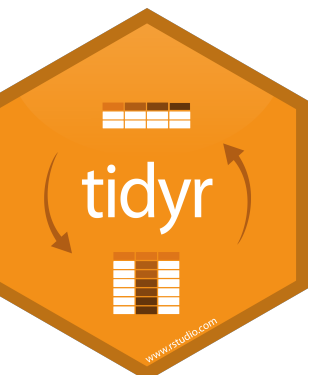
Variables:

- subject
- time
- systolic blood pressure

bp_systolic2 is tidy

subject_id <dbl>	time <dbl>	systolic <dbl>
1	1	120
1	2	118
1	3	121
2	1	125
2	2	131
3	1	141

6 rows



Your Turn 2

Using `bp_systolic2` with `group_by()`, and `summarise()`:

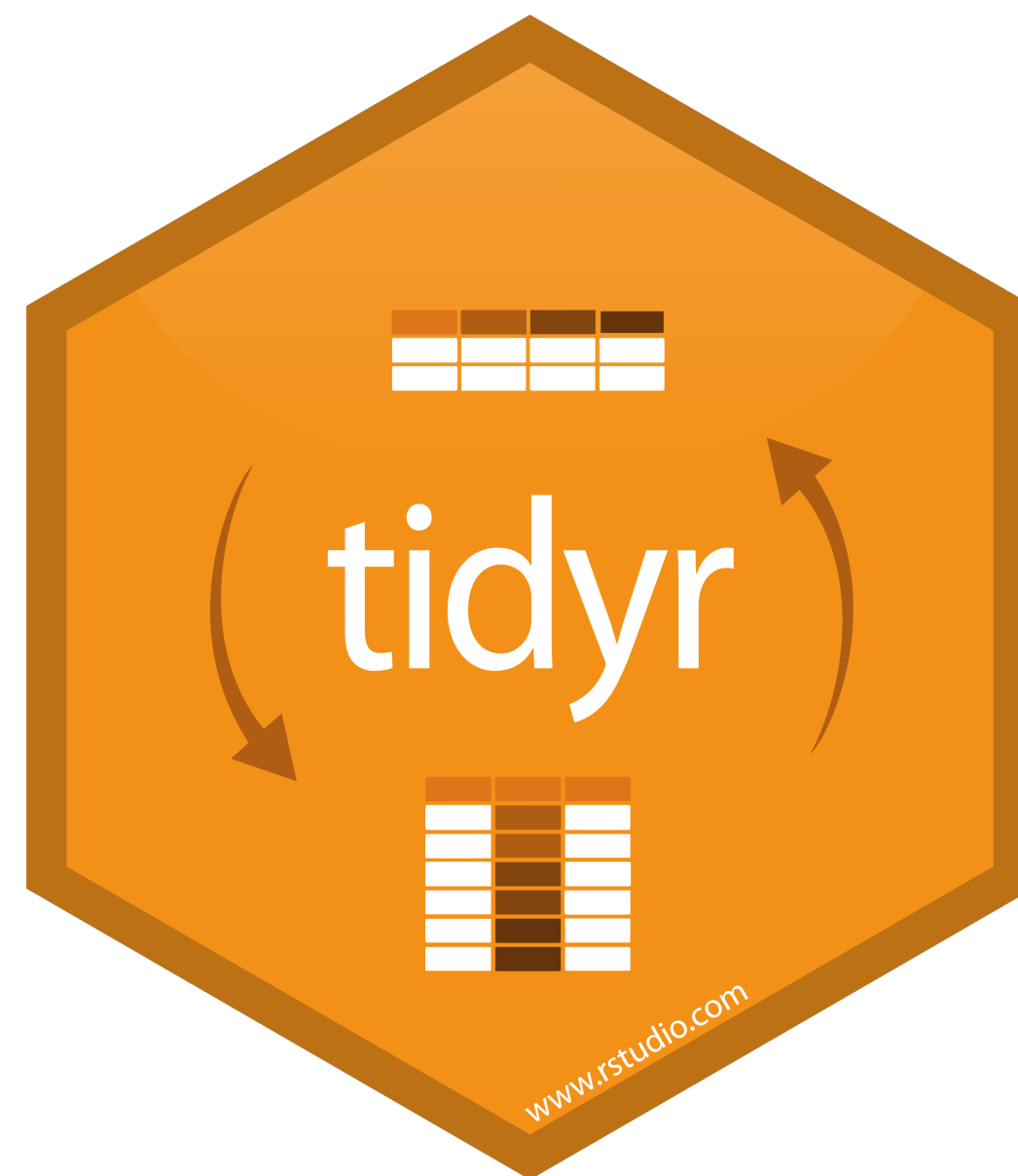
- Find the average systolic blood pressure for each subject
- Find the last time each subject was measured

```
bp_systolic2 %>%  
  group_by(subject_id) %>%  
  summarise(avg_sys = mean(systolic),  
            last_measurement = max(time))
```

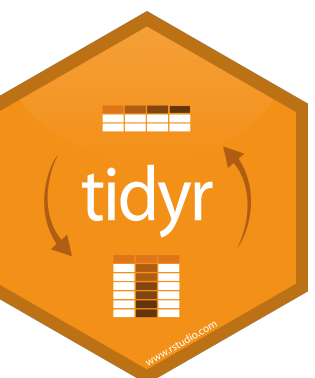
subject_id <dbl>	avg_sys <dbl>	last_measurement <dbl>
1	119.6667	3
2	128.0000	2
3	141.0000	1

3 rows

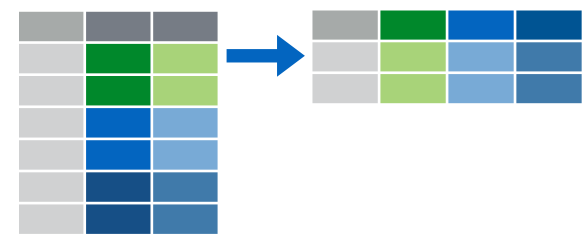
tidyr



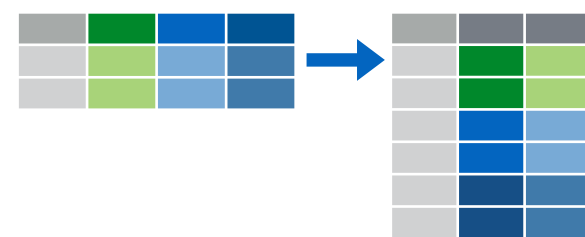
A tidyverse package that reshapes the layout of tabular data.



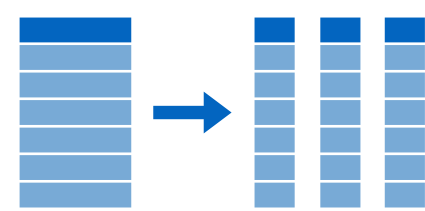
tidyr verbs



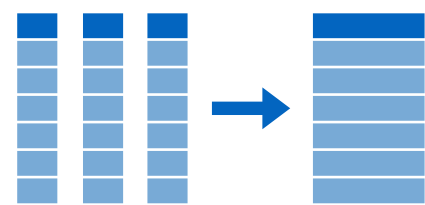
Move values into column names with **spread()**



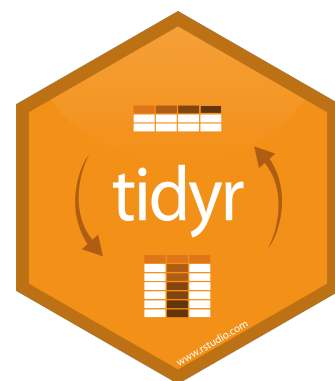
Move column names into values with **gather()**



Split a column with **separate()** or **separate_rows()**



Unite columns with **unite()**



gather()

Toy Data

```
01-Reshaping-Data.Rmd x
1 ---
2 title: "Reshaping Data"
3 output: html_notebook
4 ---
5
6 ```{r setup}
7 library(tidyverse)
8
9 # Toy data
10 cases <- tribble(
11   ~Country, ~"2011", ~"2012",
12   "FR", 7000, 6900,
13   "DE", 5800, 6000,
14   "US", 15000, 14000,
15 )
16
17 pollution <- tribble(
18   ~city, ~size, ~amount,
19   "New York", "large", 22,
20   "New York", "small", 16,
21   "London", "large", 121,
22   "London", "small", 121,
23   "Beijing", "large", 121,
24   "Beijing", "small", 121,
25 )
26
27
28 bp_systolic <- tribble(
29   ~subject_id, ~time_1, ~time_2, ~time_3,
30   1, 120, 118, 121,
```

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

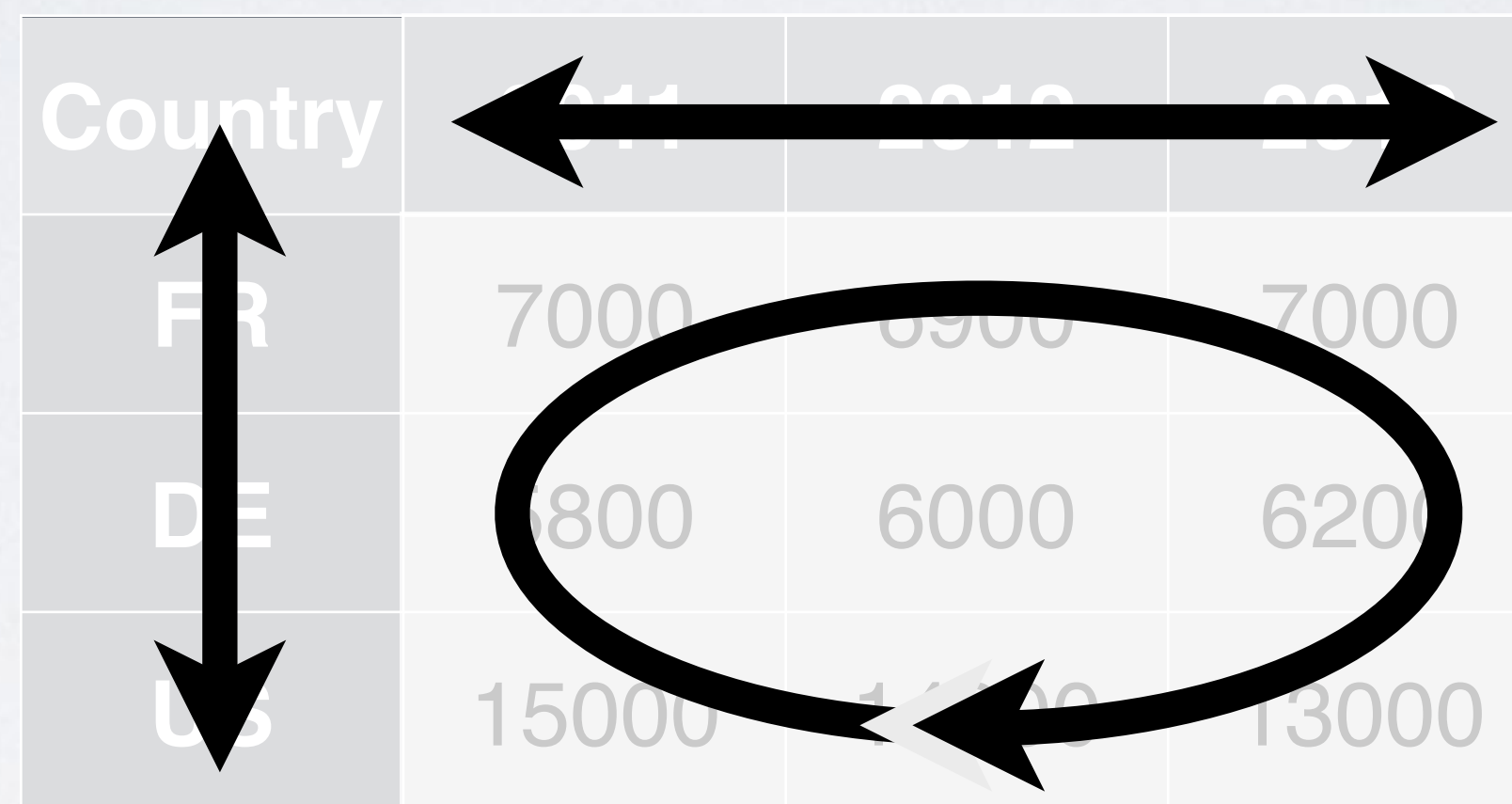
Quiz

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?



A diagram illustrating the variables in a dataset. It shows a table with three columns: Country, Year, and Count. The rows represent different countries: FR, DE, and US. A vertical double-headed arrow on the left indicates the 'Country' variable. A horizontal double-headed arrow at the top indicates the 'Year' variable. A curved double-headed arrow on the right indicates the 'Count' variable.

Country	2011	2012	2013
FR	7000	6900	7000
DE	6800	6000	6200
US	15000	14000	13000

- Country
- Year
- Count

Your Turn 3

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

gather()

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

12

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

key (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

key **value** (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

gather()

```
cases %>% gather(key = "year", value = "n", 2:4)
```

**data frame to
reshape**

**name of the
new key
column**

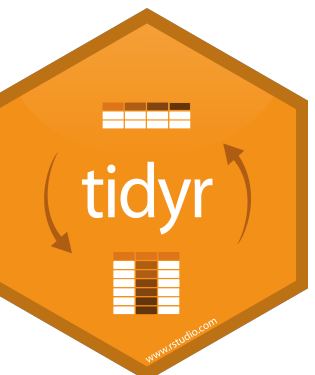
(a character
string)

**name of the
new value
column**

(a character
string)

**numeric
indexes of
columns to
collapse**

(or names)

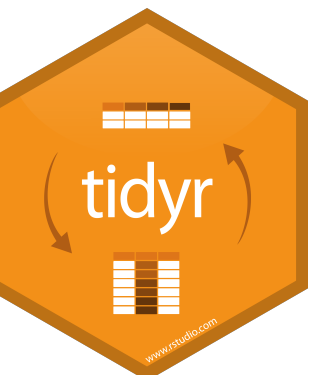


gather()

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

numeric
indexes

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

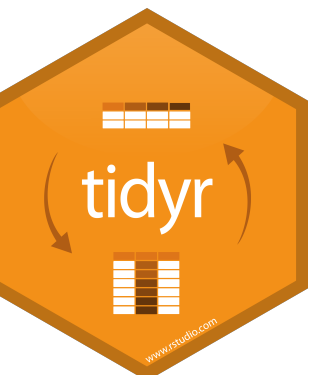


gather()

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

names

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

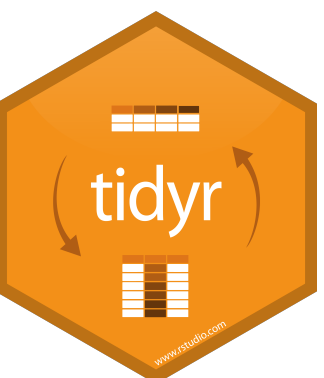


gather()

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

Everything
except...

Country <chr>	Not Country Not Country Not Country		
	2011 <dbl>	2012 <dbl>	2013 <dbl>
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000



Your Turn 4


Use **gather()** 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

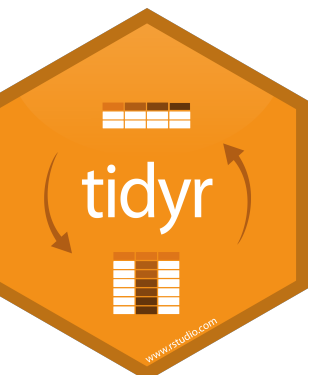
03:00


```
table4a %>%  
  gather(key = "year", value = "n", 2:3)
```




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



```
table4a %>%  
  gather(key = "year", value = "n", 2:3, convert = TRUE)
```



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

6 rows



`spread()`

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", ~
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
)
```

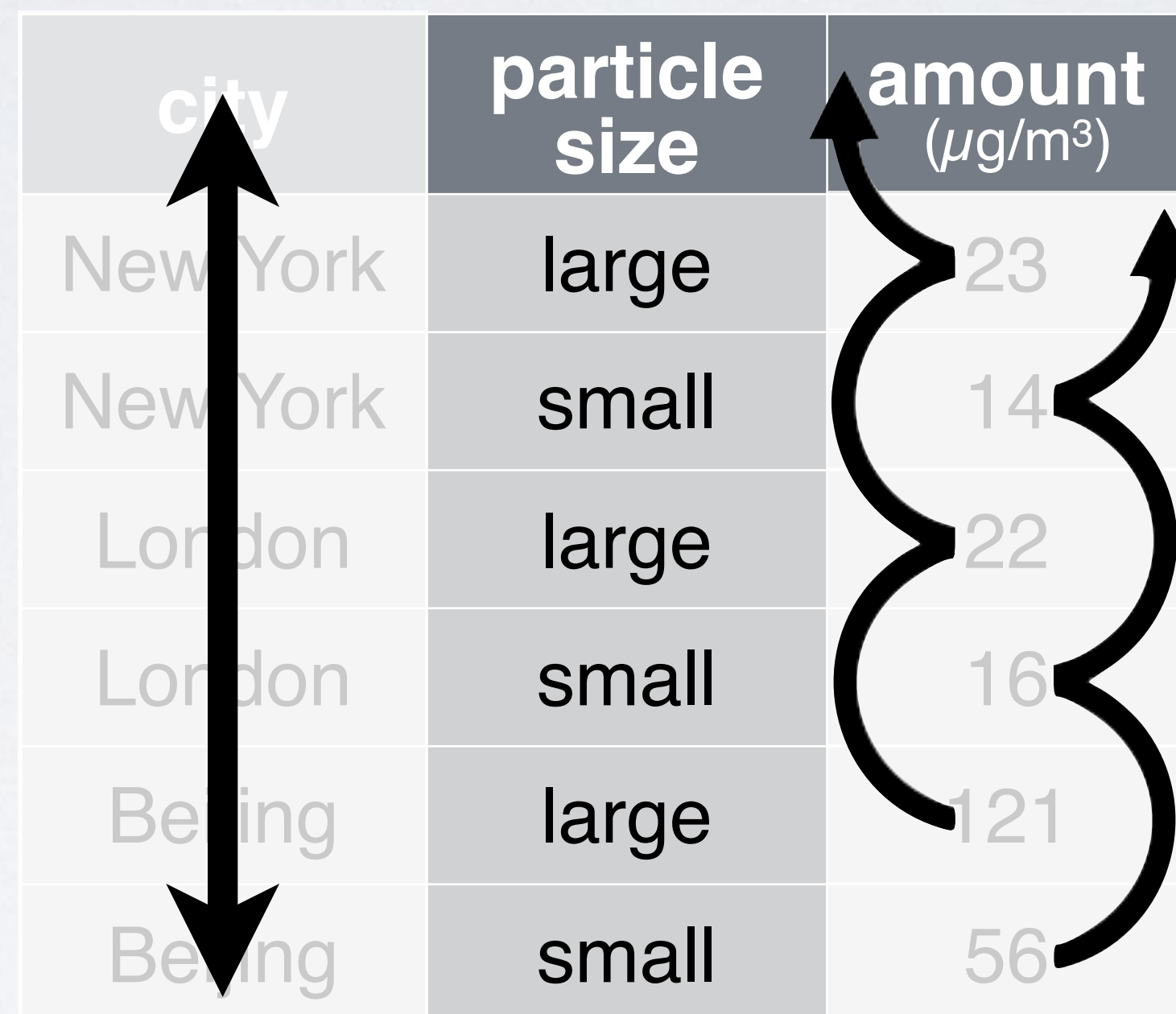
Quiz

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

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

Your Turn 5

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

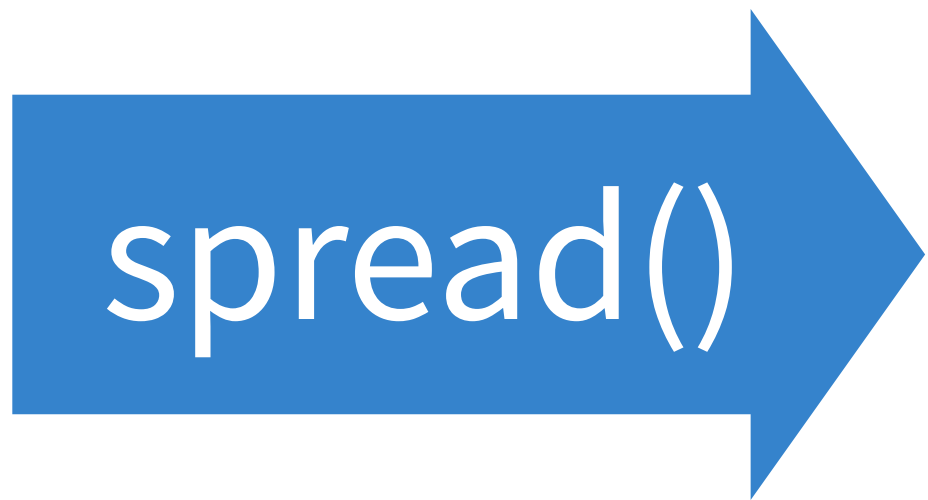
city	large	small
New York	23	14
London	22	16
Beijing	121	56

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



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

key (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

key value (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

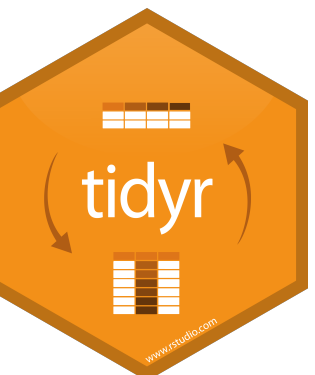
spread()

```
pollution %>% spread(key = size, value = amount)
```

data frame to
reshape

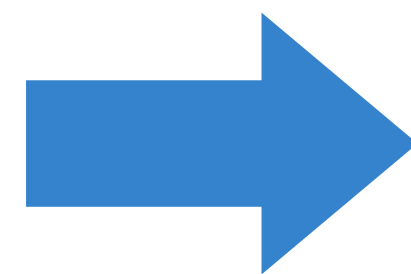
column to use for keys
(becomes new
column names)

column to use for values
(becomes new
column cells)



```
pollution %>% spread(size, amount)
```

	city	size	amount
1	New York	large	23
2	New York	small	14
3	London	large	22
4	London	small	16
5	Beijing	large	121
6	Beijing	small	56



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


Your Turn 6

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

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

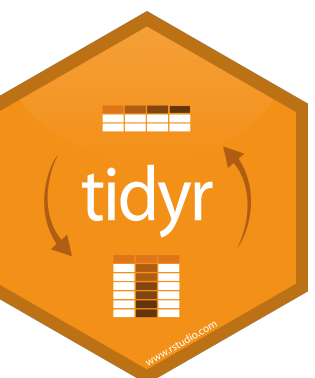
03:00


```
table2 %>%  
  spread(key = type, value = count)
```

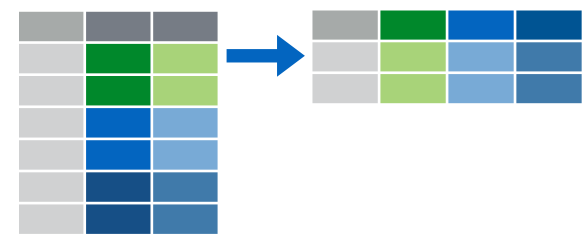


	country <chr>	year <int>	cases <int>	population <int>
1	Afghanistan	1999	745	19987071
2	Afghanistan	2000	2666	20595360
3	Brazil	1999	37737	172006362
4	Brazil	2000	80488	174504898
5	China	1999	212258	1272915272
6	China	2000	213766	1280428583

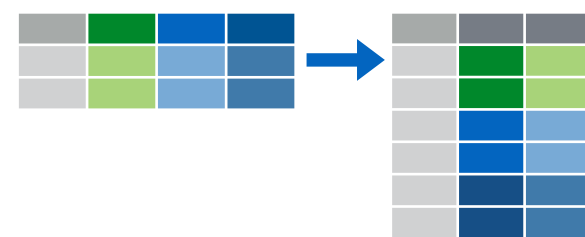
6 rows



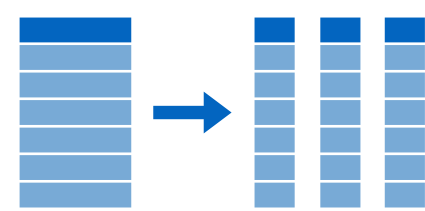
Reshaping verbs in tidyr



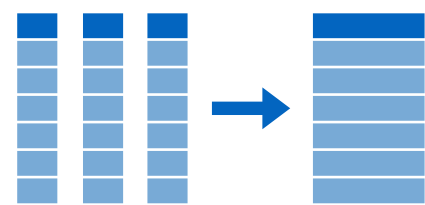
Move values into column names with **spread()**



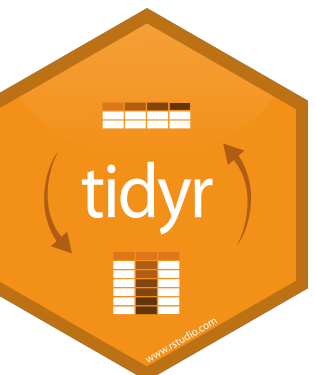
Move column names into values with **gather()**



Split a column with **separate()** or **separate_rows()**



Unite columns with **unite()**



Tidy Data with

