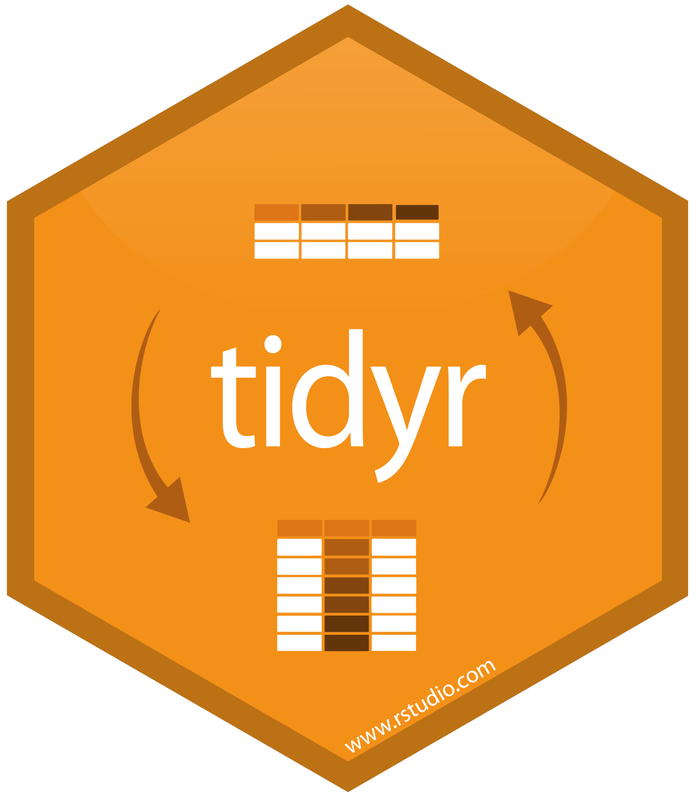
Tidy Data with



Navigate up to the **06-Tidy** folder.

Open on **06-Tidy-Exercises.Rmd** and run the first chunk.

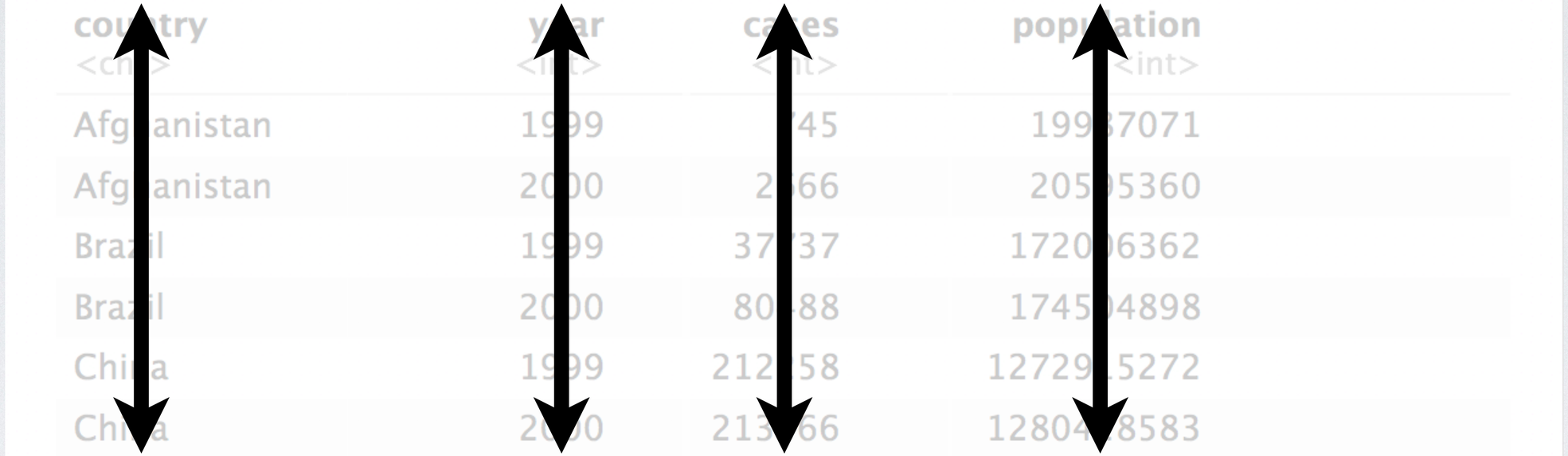
"Data are not just numbers,
they are numbers with a context."

- George Cobb and David Moore (1997)

Recall

What are the variables in this data set?

```
table1
```



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

6 rows

Recall

What are the variables in this data set?

```
table2
```

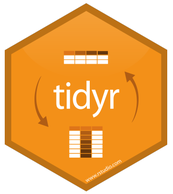
country	year	type	count
<chr>	<int>	<chr>	<int>
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999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

1–10 of 12 rows

```
table3
```

	country <chr>	year <int>	rate <chr>
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



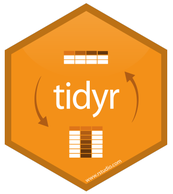
```
table4a
table4b
```

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

3 rows

	country <chr>	1999 <int>	2000 <int>
1	Afghanistan	19987071	20595360
2	Brazil	172006362	174504898
3	China	1272915272	1280428583

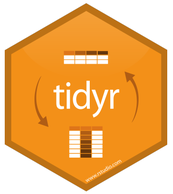
3 rows



```
table5
```

	country <chr>	century <chr>	year <chr>	rate <chr>
1	Afghanistan	19	99	745/19987071
2	Afghanistan	20	00	2666/20595360
3	Brazil	19	99	37737/172006362
4	Brazil	20	00	80488/174504898
5	China	19	99	212258/1272915272
6	China	20	00	213766/1280428583

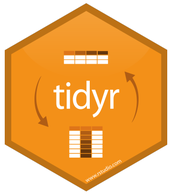
6 rows



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

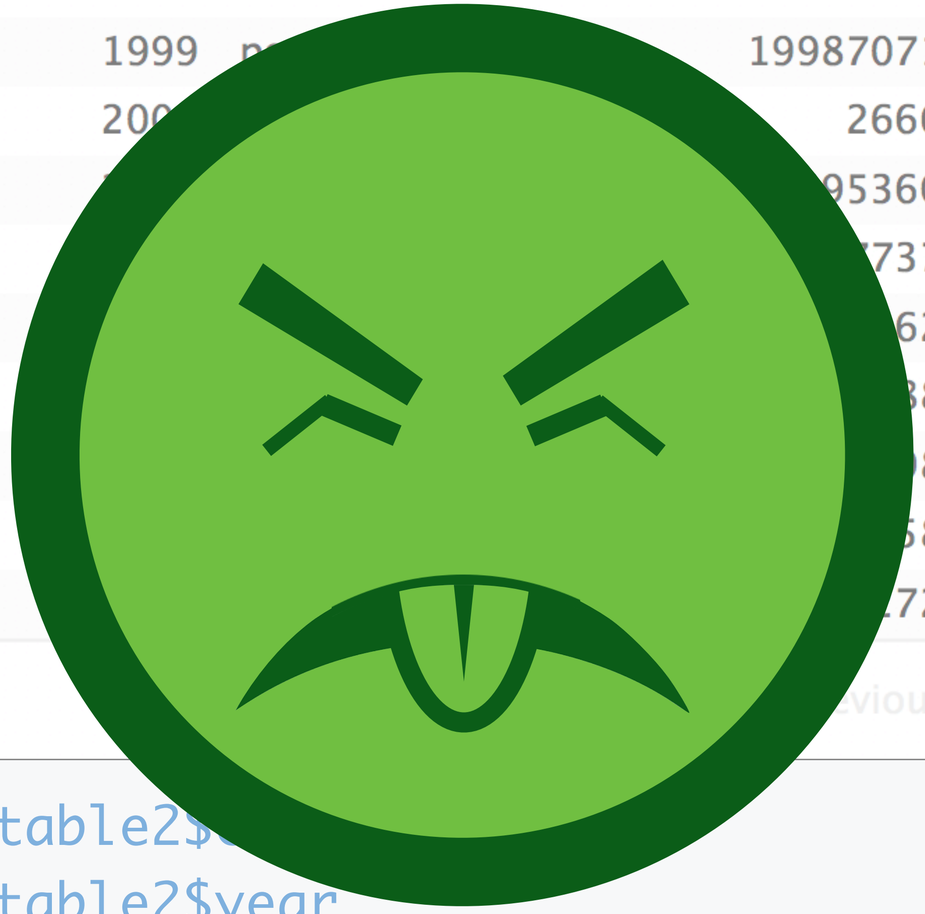
```
table1$country
table1$year
table1$cases
table1$population
```



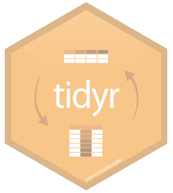
country <chr>	year <int>	type <chr>	count <int>
Afghanistan	1999	cases	745
Afghanistan	1999	p	19987071
Afghanistan	200		2666
Afghanistan			95360
Brazil			737
Brazil			62
Brazil			8
Brazil			8
China			58
China			72

1–10 of 12 rows

vious 1 2 Next

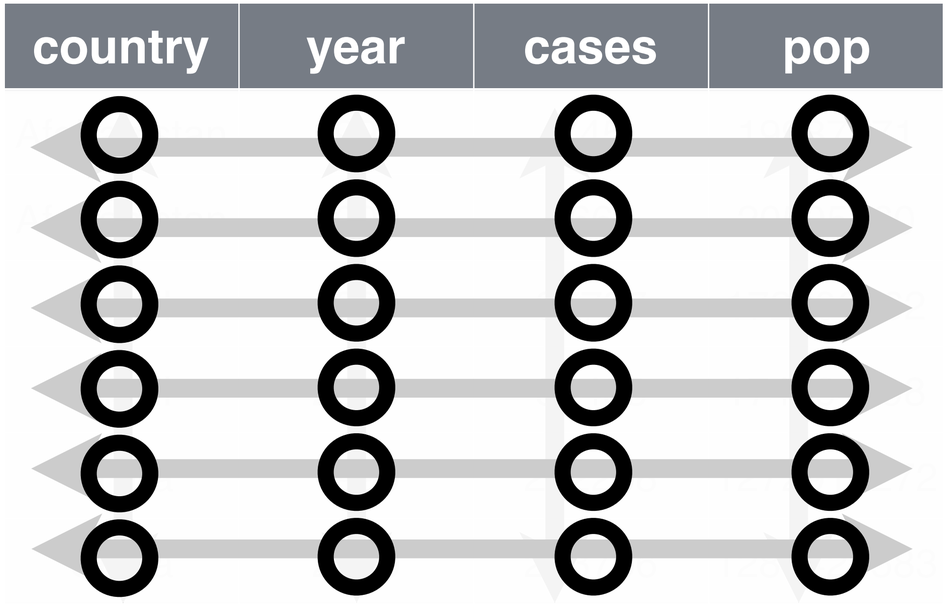


```
table2$
table2$year
table2$count[c(1,3,5,7,9,11)]
table2$count[c(2,4,6,8,10,12)]
```



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

Tidy data



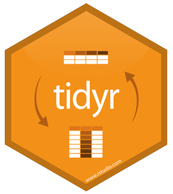
A data set is **tidy** iff:

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

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
```

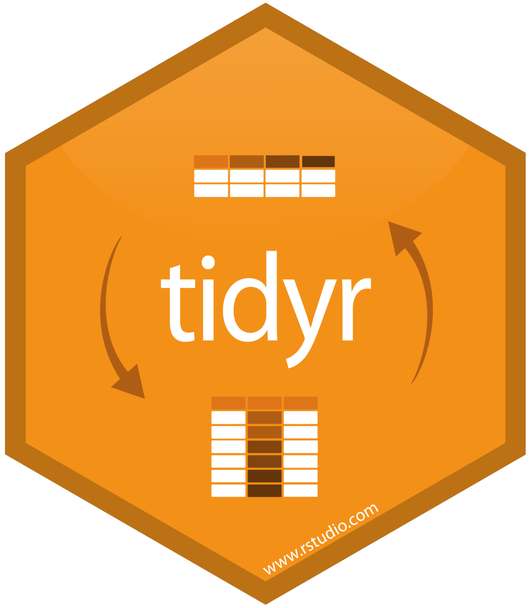


"Tidy data sets are all alike; but every messy data set is messy in its own way."

- Hadley Wickham

tidyr

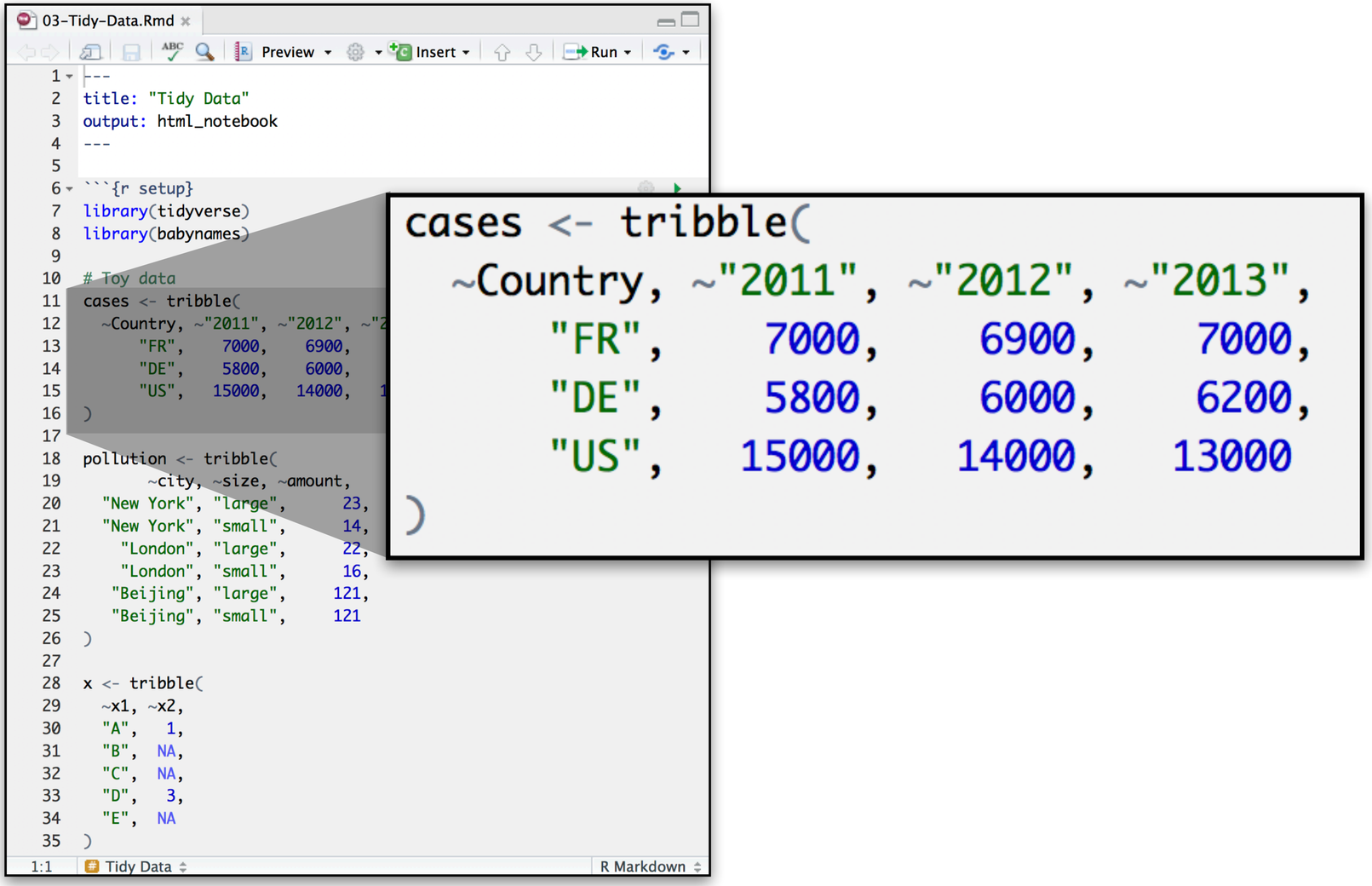
tidyr



A package that reshapes the layout of tabular data.

pivot_longer()

Toy data


03-Tidy-Data.Rmd
Preview
Insert
Run
---
title: "Tidy Data"
output: html_notebook

```{r setup}
library(tidyverse)
library(babynames)
# Toy data
cases <- tribble(
~Country, ~"2011", ~"2012", ~"2
"FR", 7000, 6900,
"DE", 5800, 6000,
"US", 15000, 14000, 1
)
pollution <- tribble(
~city, ~size, ~amount,
"New York", "large", 23,
"New York", "small", 14,
"London", "large", 22,
"London", "small", 16,
"Beijing", "large", 121,
"Beijing", "small", 121
)
x <- tribble(
~x1, ~x2,
"A", 1,
"B", NA,
"C", NA,
"D", 3,
"E", NA
)
1:1
Tidy Data
R Markdown
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?

Country	11	2012	20
FR	7000	6900	7000
DE	800	6000	620
US	15000	14600	13000

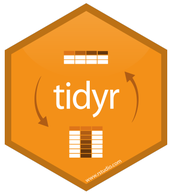
- Country
- Year
- Count

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*

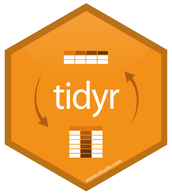
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



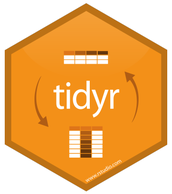
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n



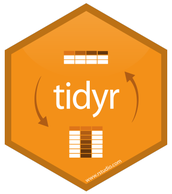
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000



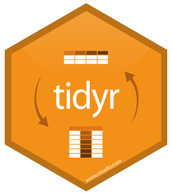
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800



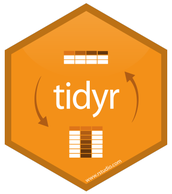
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



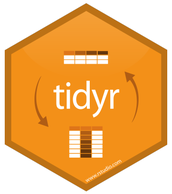
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



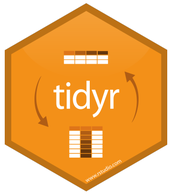
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



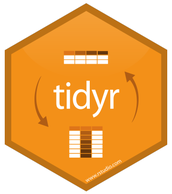
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



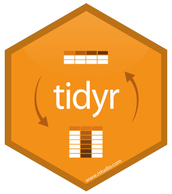
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



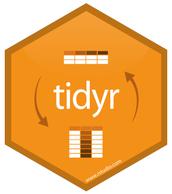
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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

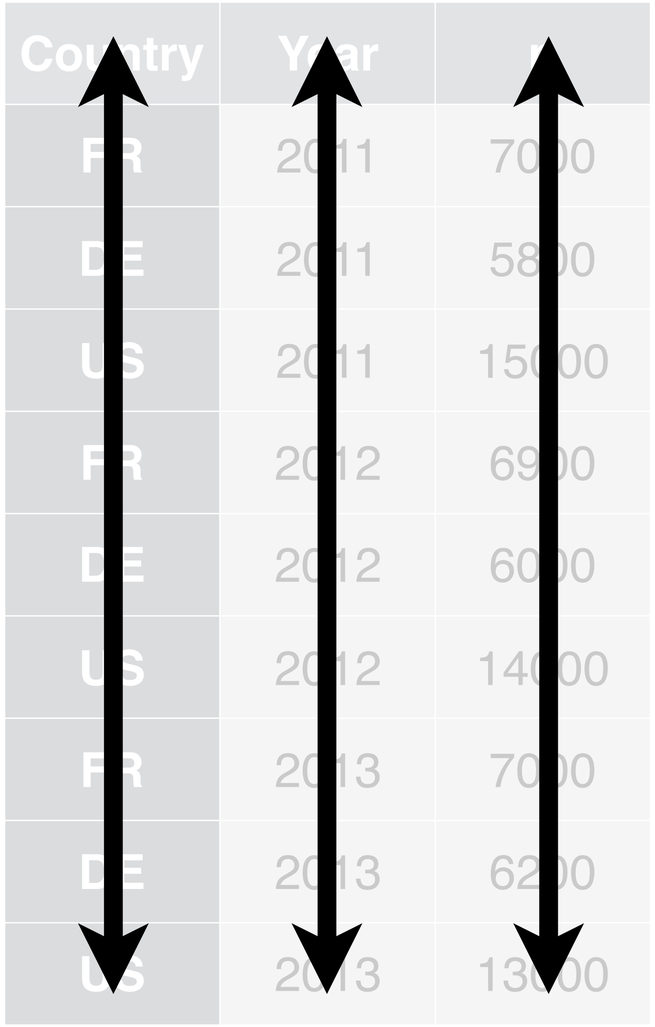


Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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

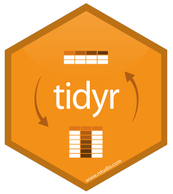


Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000



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

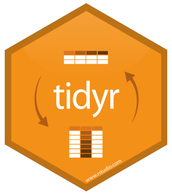
Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



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

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

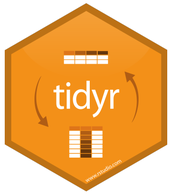


Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

names_to

(former column names)

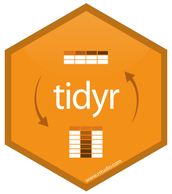
Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



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

names_to **values_to**
(former cells)

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()

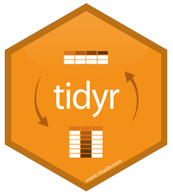
```
cases %>%
pivot_longer(cols = 2:4, names_to = "year", values_to = "n")
```

data frame to reshape

numeric indexes of columns to collapse (or names)

name of the new key column (a character string)

name of the new value column (a character string)

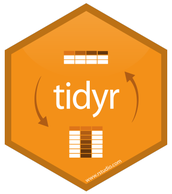


pivot_longer()

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

numeric indexes

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

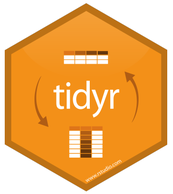


pivot_longer()

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

names

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

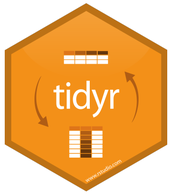


pivot_longer()

```
cases %>%
pivot_longer(starts_with("2"), names_to = "year", values_to = "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

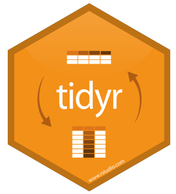


pivot_longer()

```
cases %>%
pivot_longer(-Country, names_to = "year", values_to = "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

```
table4a %>%
  pivot_longer(names_to = "year", values_to = "n", cols = 2:3)
```

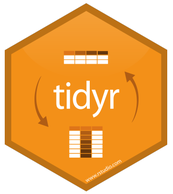
country <chr>	year <chr>	n <int>
Afghanistan	1999	745
Brazil	1999	37737
China	1999	212258
Afghanistan	2000	2666
Brazil	2000	80488
China	2000	213766

6 rows

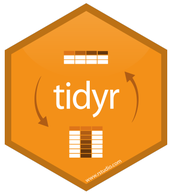
```
table4a %>%
  pivot_longer(names_to = "year", values_to = "n", cols = 2:3,
               names_ptypes = list(year = integer()))
```

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

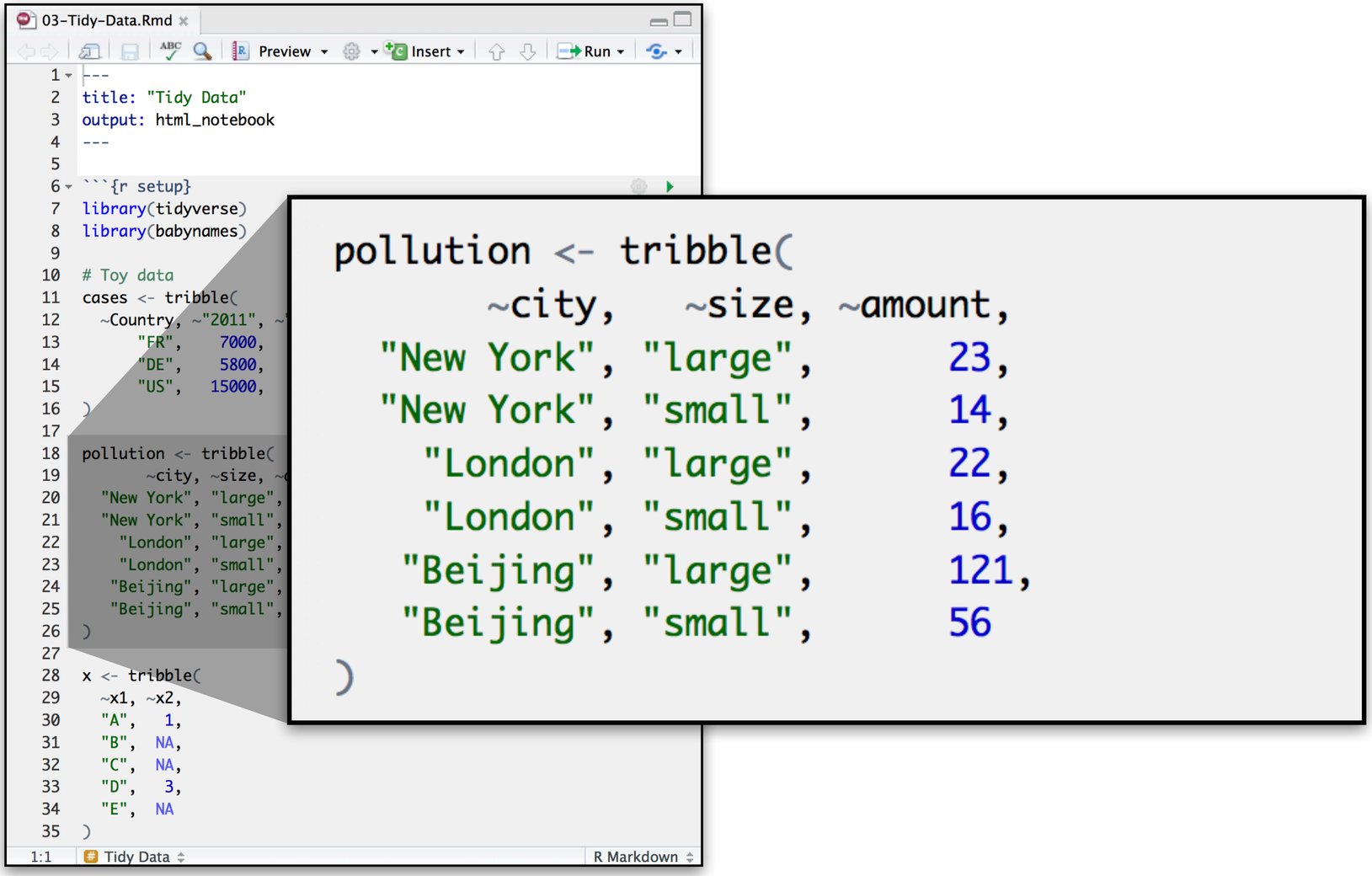


```
pivot_longer(
  data,
  cols,
  names_to = "name",
  names_prefix = NULL,
  names_sep = NULL,
  names_pattern = NULL,
  names_ptypes = list(),
  names_repair = "check_unique",
  values_to = "value",
  values_drop_na = FALSE,
  values_ptypes = list()
)
```



pivot_wider()

Toy data


03-Tidy-Data.Rmd
Preview
Insert
Run
---
title: "Tidy Data"
output: html_notebook

```{r setup}
library(tidyverse)
library(babynames)
# Toy data
cases <- tribble(
~Country, ~"2011", ~
"FR", 7000,
"DE", 5800,
"US", 15000,
)
pollution <- tribble(
~city, ~size, ~
"New York", "large",
"New York", "small",
"London", "large",
"London", "small",
"Beijing", "large",
"Beijing", "small",
)
x <- tribble(
~x1, ~x2,
"A", 1,
"B", NA,
"C", NA,
"D", 3,
"E", NA
)
1:1
Tidy Data
R Markdown
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

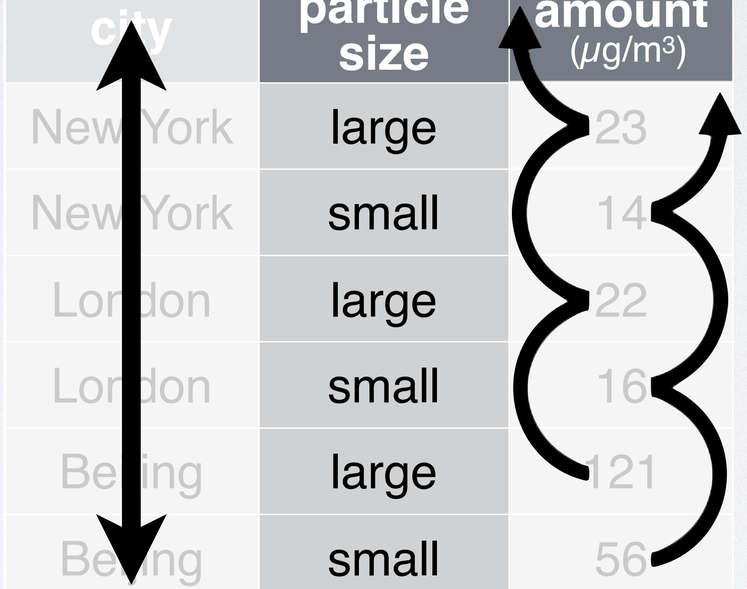
What are the variables in pollution?

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



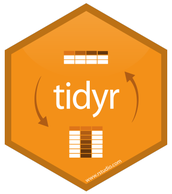
- City
- Amount of large particulate
- Amount of small particulate

Your Turn 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*

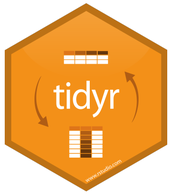
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

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



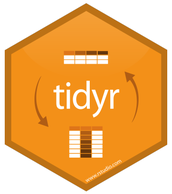
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small



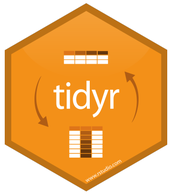
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	



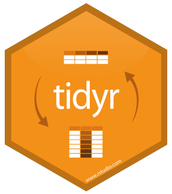
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14



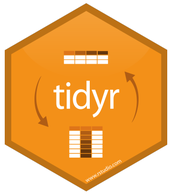
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	



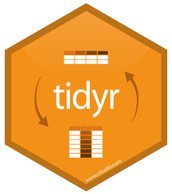
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	16



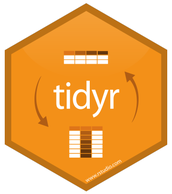
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

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

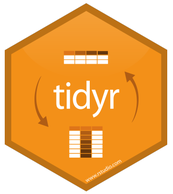


city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

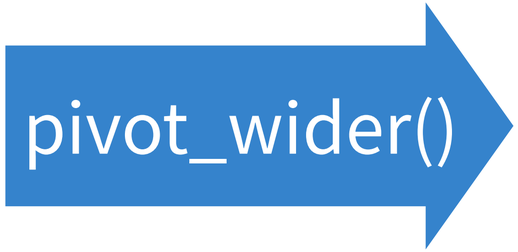
city	large	small
New York	23	14
London	22	16
Beijing	121	56



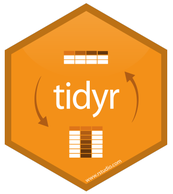
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56



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

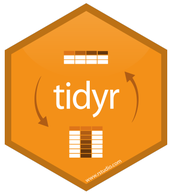


city	large	small
New York	23	14
London	22	16
Beijing	121	56



	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

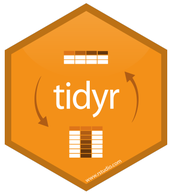
city	large	small
New York	23	14
London	22	16
Beijing	121	56



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

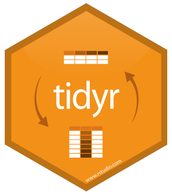
city	large	small
New York	23	14
London	22	16
Beijing	121	56



names_from **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



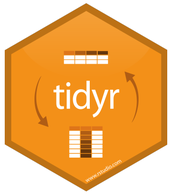
pivot_wider()

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

data frame to reshape

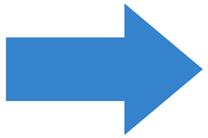
column to use for keys (becomes new column names)

column to use for values (becomes new column cells)



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

```
      city  size amount
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 4

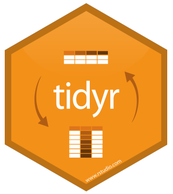
Use **pivot_wider()** to reorganize **table2** into four columns: *country*, *year*, *cases*, and *population*.

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

```
table2 %>%
  pivot_wider(names_from = type, values_from = count)
```

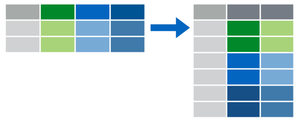
	country <chr>	year <int>	cases <int>	population <int>
1	Afghanistan	1999	745	19987071
2	Afghanistan	2000	2666	20595360
3	Brazil	1999	37737	172006362
4	Brazil	2000	80488	174504898
5	China	1999	212258	1272915272
6	China	2000	213766	1280428583

6 rows



Recap

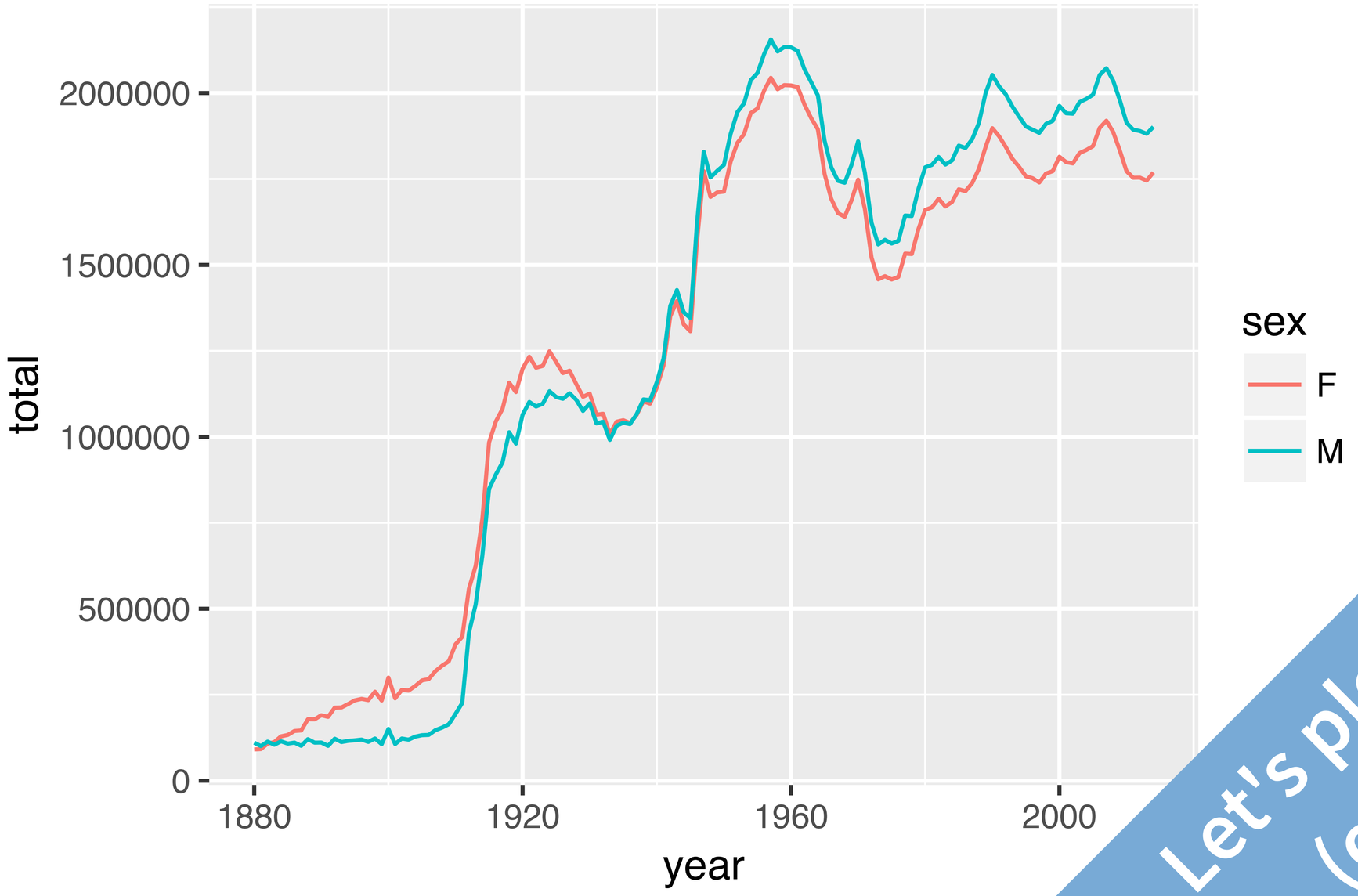
Move values into column names with **pivot_wider()**



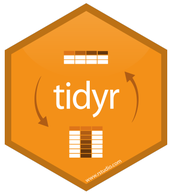
Move column names into values with **pivot_longer()**

Reshaping Final Exam

Number of children by year and gender



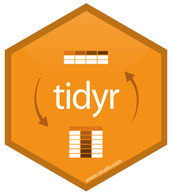
Let's plot % male (or female)



Can we calculate the yearly percent of boys (or girls)?

```
babynames
```

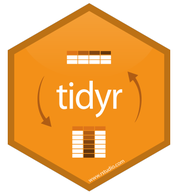
```
   year sex    name           n    prop
  <dbl> <chr> <chr>      <int>   <dbl>
1  1880 F     Mary        7065 0.0724
2  1880 F     Anna        2604 0.0267
3  1880 F     Emma        2003 0.0205
4  1880 F     Elizabeth   1939 0.0199
5  1880 F     Minnie      1746 0.0179
6  1880 F     Margaret    1578 0.0162
```



Can we calculate the yearly percent of boys (or girls)?

```
babynames %>%
  group_by(year, sex) %>%
  summarise(n = sum(n))
```

```
    year   sex       n
   <dbl> <chr>   <int>
1   1880     F   90993
2   1880     M  110491
3   1881     F   91954
4   1881     M  100745
5   1882     F  107850
6   1882     M  113688
```



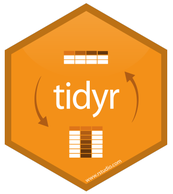
Can we calculate the yearly percent of boys (or girls)?

```
babynames %>%
  group_by(year, sex) %>%
  summarise(n = sum(n))
```

```
   year   sex      n
  <dbl> <chr>  <int>
1  1880     F  90993
2  1880     M 110491
3  1881     F  91954
4  1881     M 100745
5  1882     F 107850
6  1882     M 113688
```

$$\%\, male = \frac{male}{male + female} \times 100$$

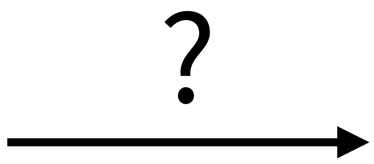
Now what?



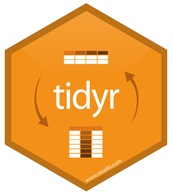
Can we calculate the yearly percent of boys (or girls)?

```
better_layout %>%
  mutate(percent_male = M / (M + F) * 100)
```

	year	sex	n
	<dbl>	<chr>	<int>
1	1880	F	90993
2	1880	M	110491
3	1881	F	91954
4	1881	M	100745
5	1882	F	107850
6	1882	M	113688



	year	F	M
*	<dbl>	<int>	<int>
1	1880	90993	110491
2	1881	91954	100745
3	1882	107850	113688
4	1883	112321	104629
5	1884	129022	114445
6	1885	133055	107800



Your Turn 7

Extend this code to reshape the data. Calculate the percent of male (or female) children by year. Then plot the percent over time.

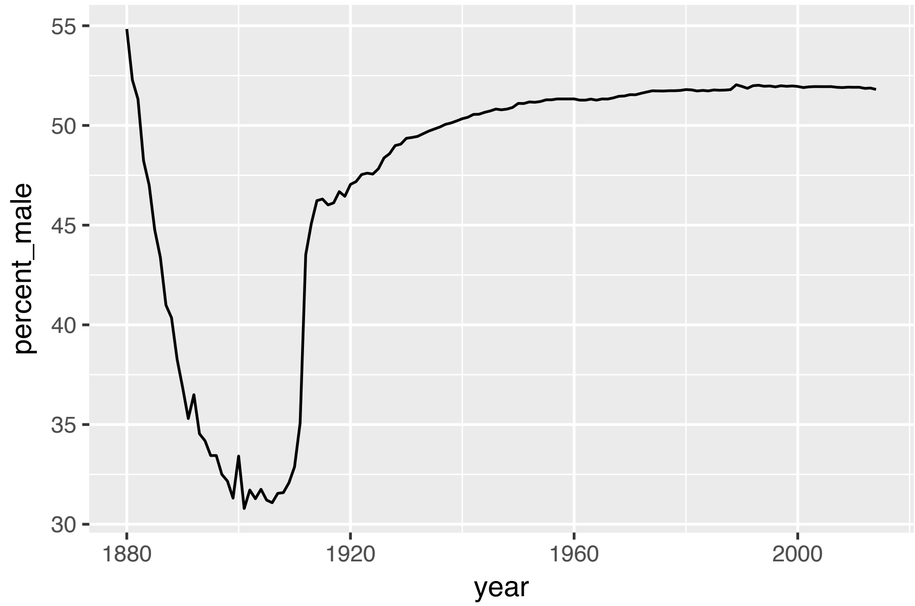
```
babynames %>%
   group_by(year, sex) %>%
   summarise(n = sum(n))
```

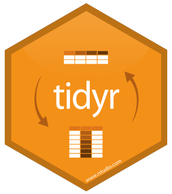
	year	sex	n
	<dbl>	<chr>	<int>
1	1880	F	90993
2	1880	M	110491
3	1881	F	91954

?

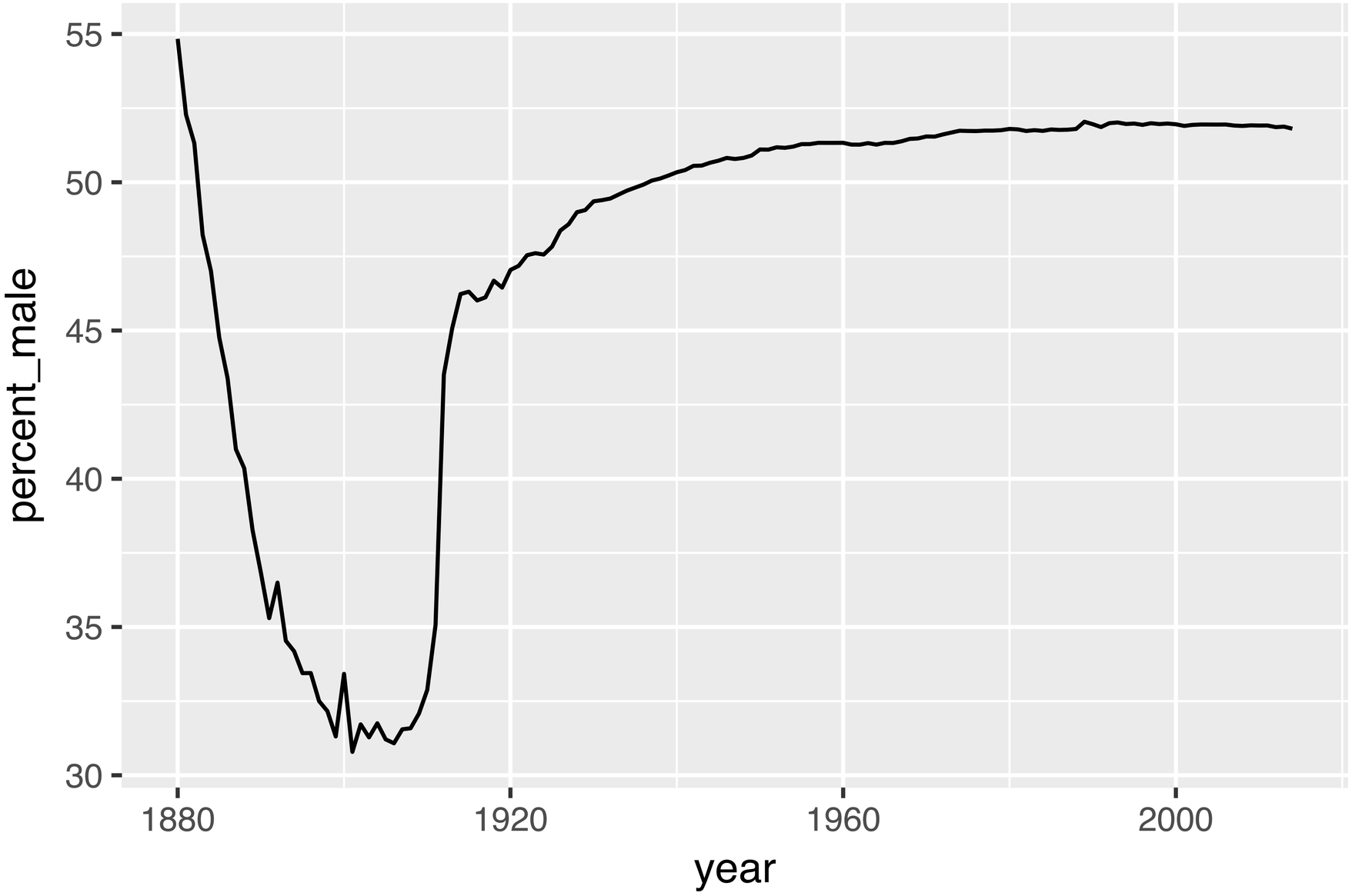
	year	F	M
*	<dbl>	<int>	<int>
1	1880	90993	110491
2	1881	91954	100745
3	1882	107850	113688

```
babynames %>%
  group_by(year, sex) %>%
  summarise(n = sum(n)) %>%
  pivot_wider(names_from = sex, values_from = n) %>%
  mutate(percent_male = M / (M + F) * 100) %>%
  ggplot(aes(year, percent_male)) + geom_line()
```





Percent of children that are male by year

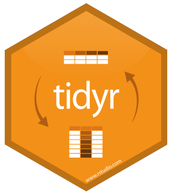


General advice

Describe what you want to do in an **equation**. Each **variable** in the equation should correspond to a column in your data:

- "color by sex"
  color = sex

- "calculate the proportion of males"
  prop male = number of males / number of females + number of males



Tidy Data with