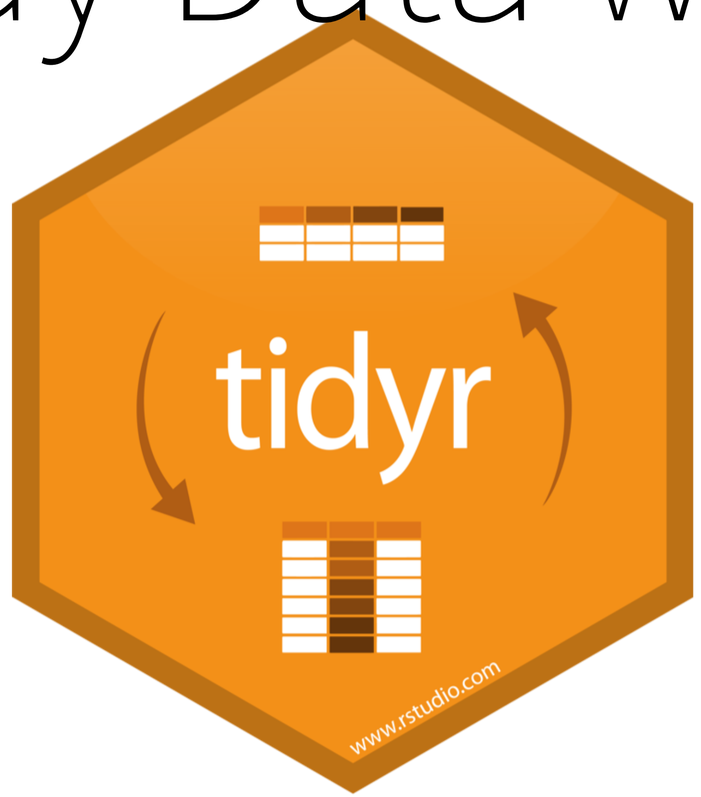
Transform Data with

Group-wise operations with `group_by()` and `summarize()`

- `group_by()` changes each function from operating on the full dataset to specified groups. *This can be done in conjunction with other* `dplyr` *functions!*
- `summarize()` reduces multiple values down to a single summary

```
bcwq %>%
    group_by(date) %>%
    summarize(Sal_psu_mean= mean(Sal_psu, na.rm = TRUE),
              DO_pct_mean = mean(DO_pct, na.rm = TRUE)
                )
```

Tidy Data with



Adapted for Cressman and Dunnigan 2020 | Adapted further for Cressman 2022

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

- Garrett Grolemund

Tidy data

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*

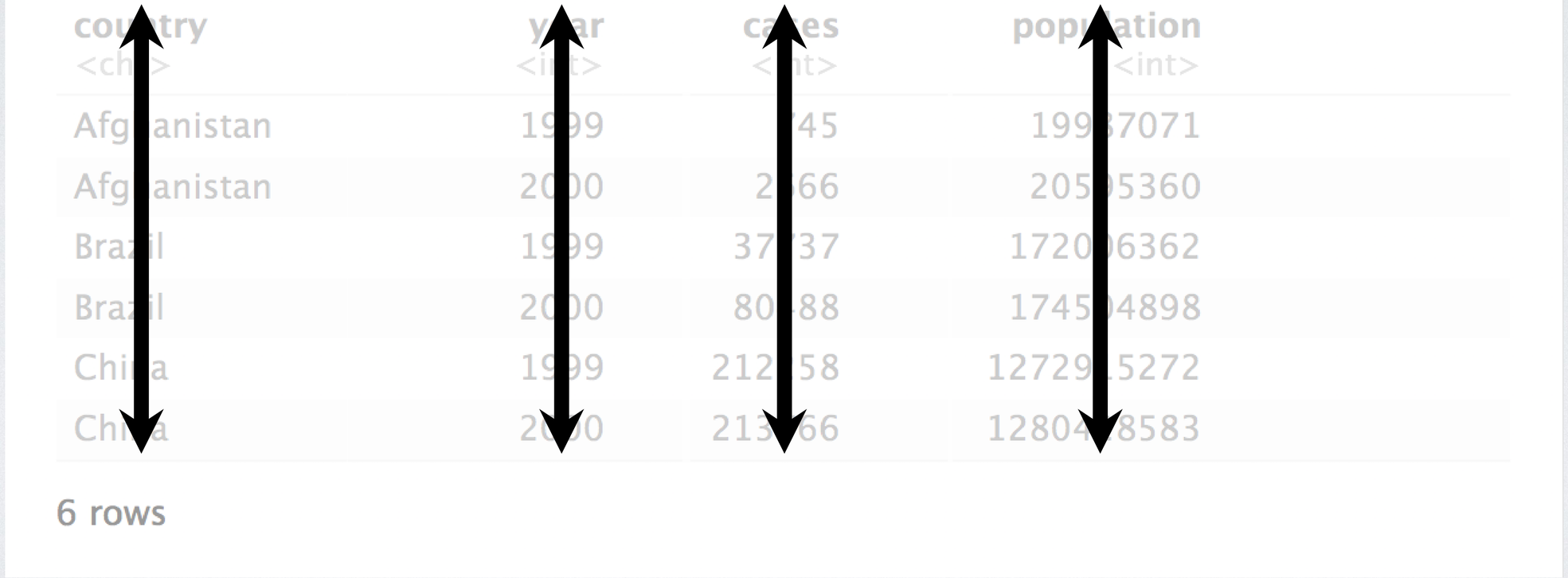
Broman and Woo, 2017: *Data Organization in Spreadsheets*

Adapted for Cressman and Dunnigan 2020 | Adapted further for Cressman 2022

Quiz

What are the variables in this data set?

table1



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

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*

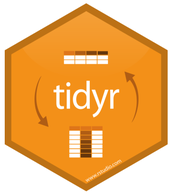
value: belongs to one **variable** and one **observation**

Adapted for Cressman and Dunnigan 2020 | Adapted further for Cressman 2022

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

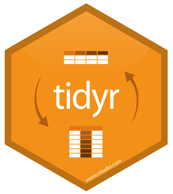
```
table1$country
table1$year
table1$cases
table1$population
```



country <chr>	year <int>	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

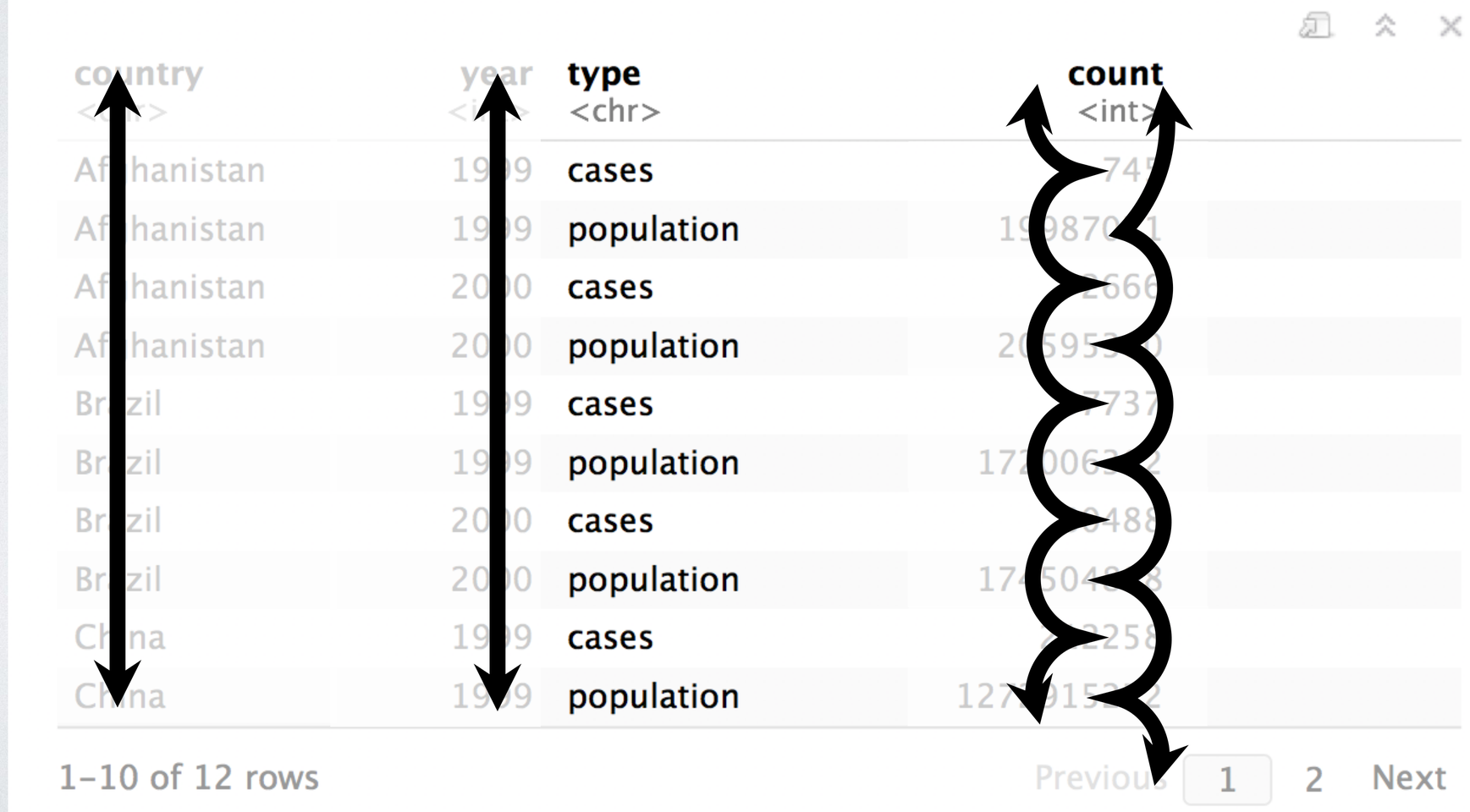
```
table1$cases / table1$population -> table1$rate
```



Quiz

What are the variables in this data set?

table2



country	year	type	count
		<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

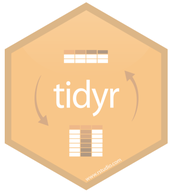
1–10 of 12 rows Previous 1 2 Next

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

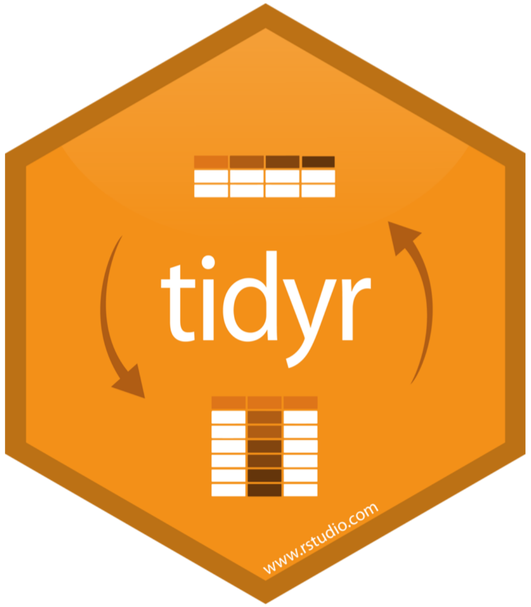
vious 1 2 Next

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

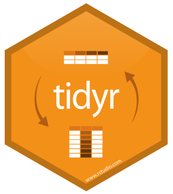


tidyr

tidyr



A package that reshapes the layout of tabular data.



pivot_wider()

Our data use-case

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Reserve	Date	SiteID	TransectID	PlotID	Lat	Long	Orthome tr	Juncus roemerianus	Spartina alterniflora	Borrichia frutescens	Distichlis spicata	Spartina patens	Other
2	GND	7/26/2018	spal	1	1	30.36248	-88.4139	0.22874	20	10				
3	GND	7/26/2018	spal	1	2	30.36236	-88.4138	0.18774	20	5				
4	GND	7/26/2018	spal	1	3	30.36224	-88.4137	0.18604	0	50				
5	GND	7/26/2018	spal	1	4	30.36212	-88.4136	0.20855	55	2.5				
6	GND	7/26/2018	spal	1	5	30.36199	-88.4135	0.21419	30	0				

	A	B	C	D	E	F	G	H	I	J
1	Reserve	Date	SiteID	TransectID	PlotID	Lat	Long	Orthome r	Species	Cover
2	GND	7/26/2018	spal	1	1	30.36248	-88.4139	0.2287 4	Juncus roemerianus	20
3	GND	7/26/2018	spal	1	1	30.36248	-88.4139	0.2287 4	Spartina alterniflora	10
4	GND	7/26/2018	spal	1	2	30.36236	-88.4138	0.1877 4	Juncus roemerianus	20
5	GND	7/26/2018	spal	1	2	30.36236	-88.4138	0.1877 4	Spartina alterniflora	5
6	GND	7/26/2018	spal	1	3	30.36224	-88.4137	0.1860 4	Juncus roemerianus	0
7	GND	7/26/2018	spal	1	3	30.36224	-88.4137	0.1860 4	Spartina alterniflora	50
8	GND	7/26/2018	spal	1	4	30.36212	-88.4136	0.2085 5	Juncus roemerianus	55
1071	GND	8/12/2020	spal	3	6	30.3617	-88.4139	0.1895 4	Juncus roemerianus	30
1072	GND	8/12/2020	spal	3	6	30.3617	-88.4139	0.1895 4	Spartina alterniflora	5
1073	GND	8/12/2020	spal	3	7	30.36159	-88.4138	0.1486 6	Juncus roemerianus	50
1074	GND	8/12/2020	spal	3	7	30.36159	-88.4138	0.1486 6	Spartina alterniflora	2.5
1075	GND	8/12/2020	spal	3	8	30.36147	-88.4138	0.1678 6	Juncus roemerianus	50
1076	GND	8/12/2020	spal	3	8	30.36147	-88.4138	0.1678 6	Spartina alterniflora	2.5
1077	GND	8/12/2020	spal	3	9	30.36131	-88.4137	0.17727	Juncus roemerianus	40
1078	GND	8/12/2020	spal	3	9	30.36131	-88.4137	0.17727	Spartina alterniflora	5

0
0
2.5
30
20
10
10
20
2.5

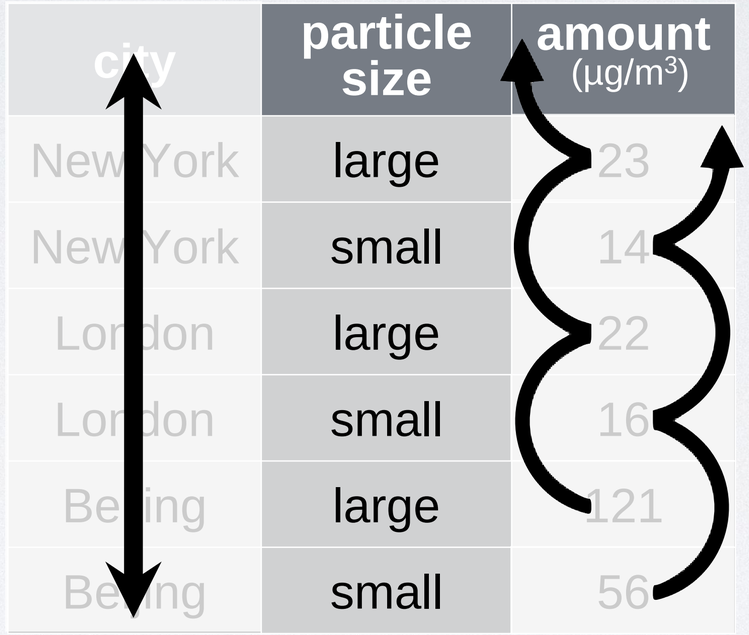
Quiz

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?

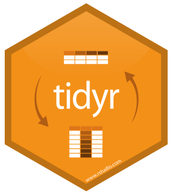


city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

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

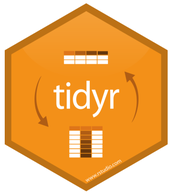
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	



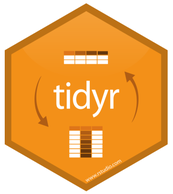
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14



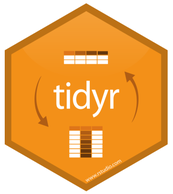
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	



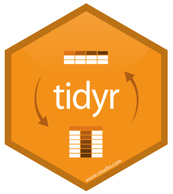
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	large	small
New York	23	14
London	22	16



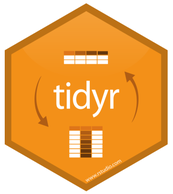
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

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

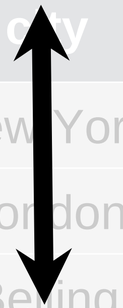


city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

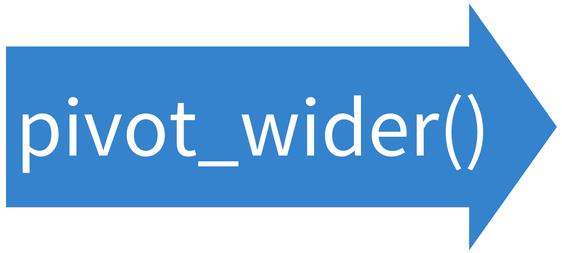
city	large	small
New York	23	14
London	22	16
Beijing	121	56



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



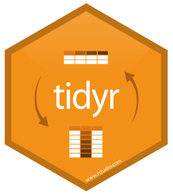
city	size	amount
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56



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

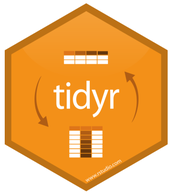
city	large	small
New York	23	14
London	22	16
Beijing	121	56



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

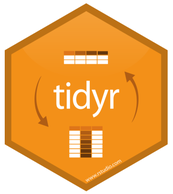
city	large	small
New York	23	14
London	22	16
Beijing	121	56



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



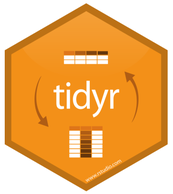
pivot_wider()

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

data frame to reshape

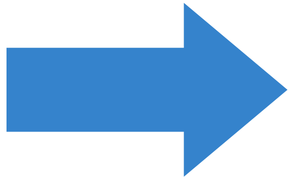
column to use for keys (becomes new column names)

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

pivot_longer()

Quiz

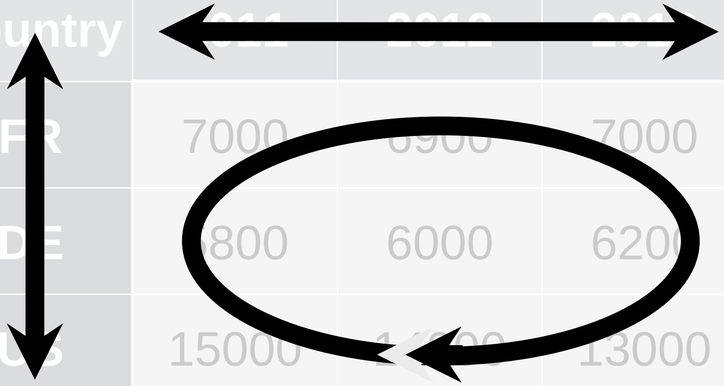
What are the variables in cases?

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

Quiz

What are the variables in cases?

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14800	13000



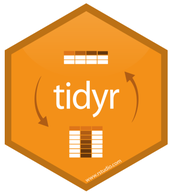
- Country
- Year
- Count

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*

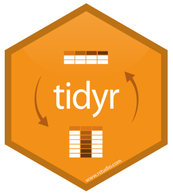
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



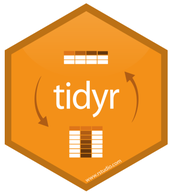
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n



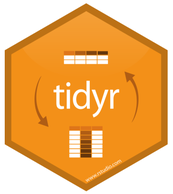
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000



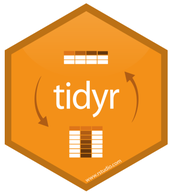
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Country	Year	n
FR	2011	7000
DE	2011	5800



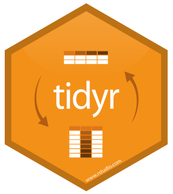
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



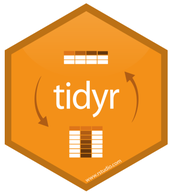
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



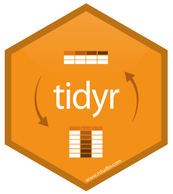
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



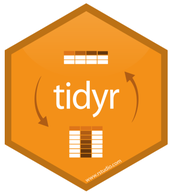
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



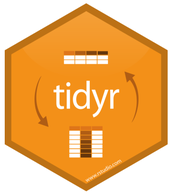
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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



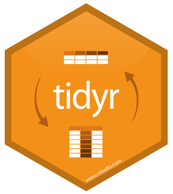
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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

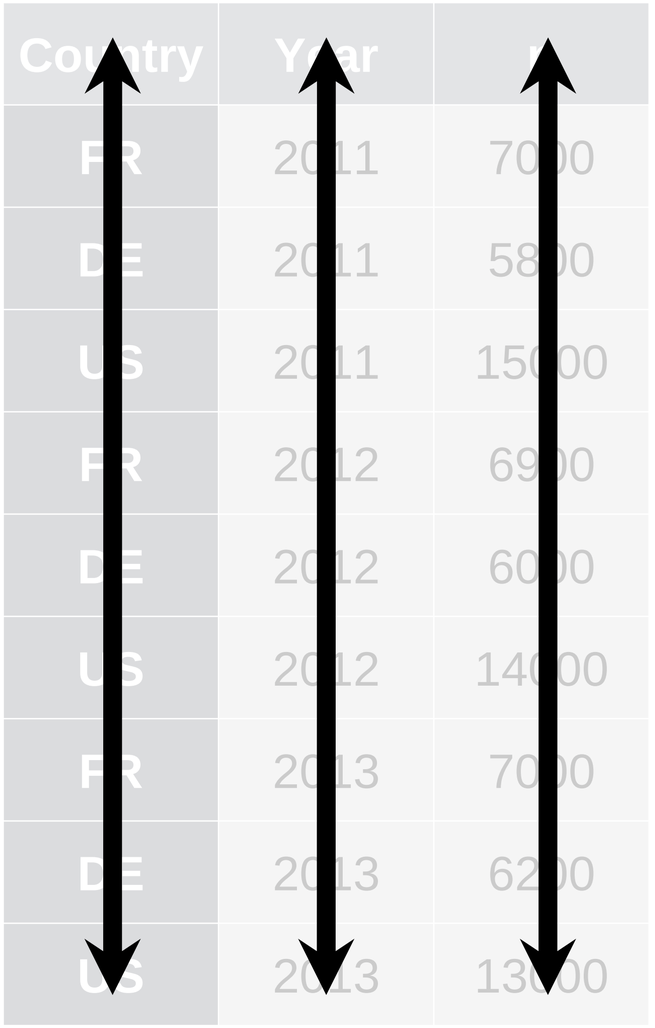


Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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

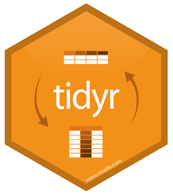


Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000



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

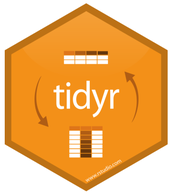
Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



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

1 2

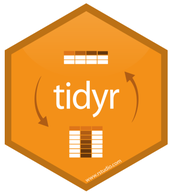
Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



names_to (former column names)

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

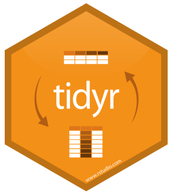
Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000



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()

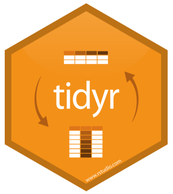
```
cases %>% pivot_longer(cols = 2:4, names_to = "year", values_to = "n")
```

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)

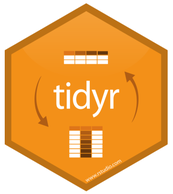


pivot_longer()

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

numeric indices

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

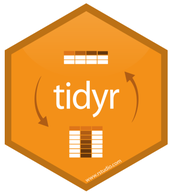


pivot_longer()

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

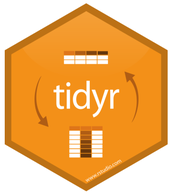


pivot_longer()

```
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 4

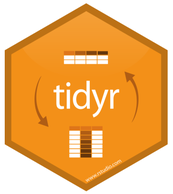
Use **pivot_wider()** to reorganize **table2** into four columns: *country*, *year*, *cases*, and *population*.

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



Tidy Data with