Data Science for Data Wranglers Part 3:

Tidy Data

Data Manipulation

Changing the variables, values, and units of analysis contained in the data set.

Data Tidying

Changing the layout of tabular data to make it suitable for a particular piece of software (R).

Data Visualization

Transforming the data to a visual format that reveals visual patterns.

```
# devtools::install_github("rstudio/EDAWR")
library(EDAWR)
```

storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

cases

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

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

```
# devtools::install_github("rstudio/EDAWR")
library(EDAWR)
```

storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

- Storm name
- Wind Speed (mph)
- Air Pressure
- Date

cases

Country			
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	[illegible]	13000

- Country
- Year
- Count

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 particles
- Amount of small particles

```
# devtools::install_github("rstudio/EDAWR")
library(EDAWR)
```

storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

```
storms$storm
storms$wind
storms$pressure
storms$date
```

cases

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

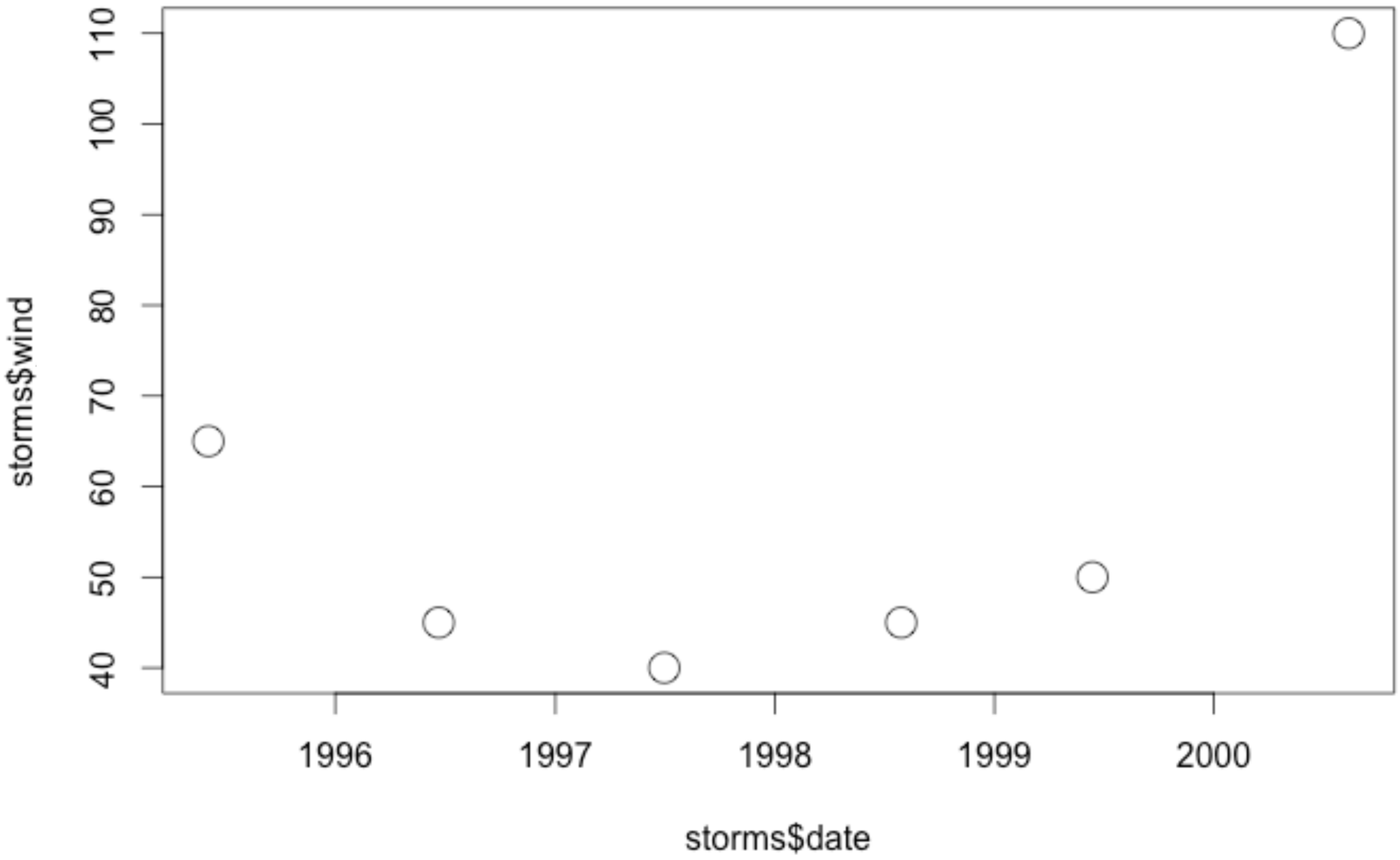
```
cases$country
names(cases)[-1]
unlist(cases[1:3, 2:4])
```

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

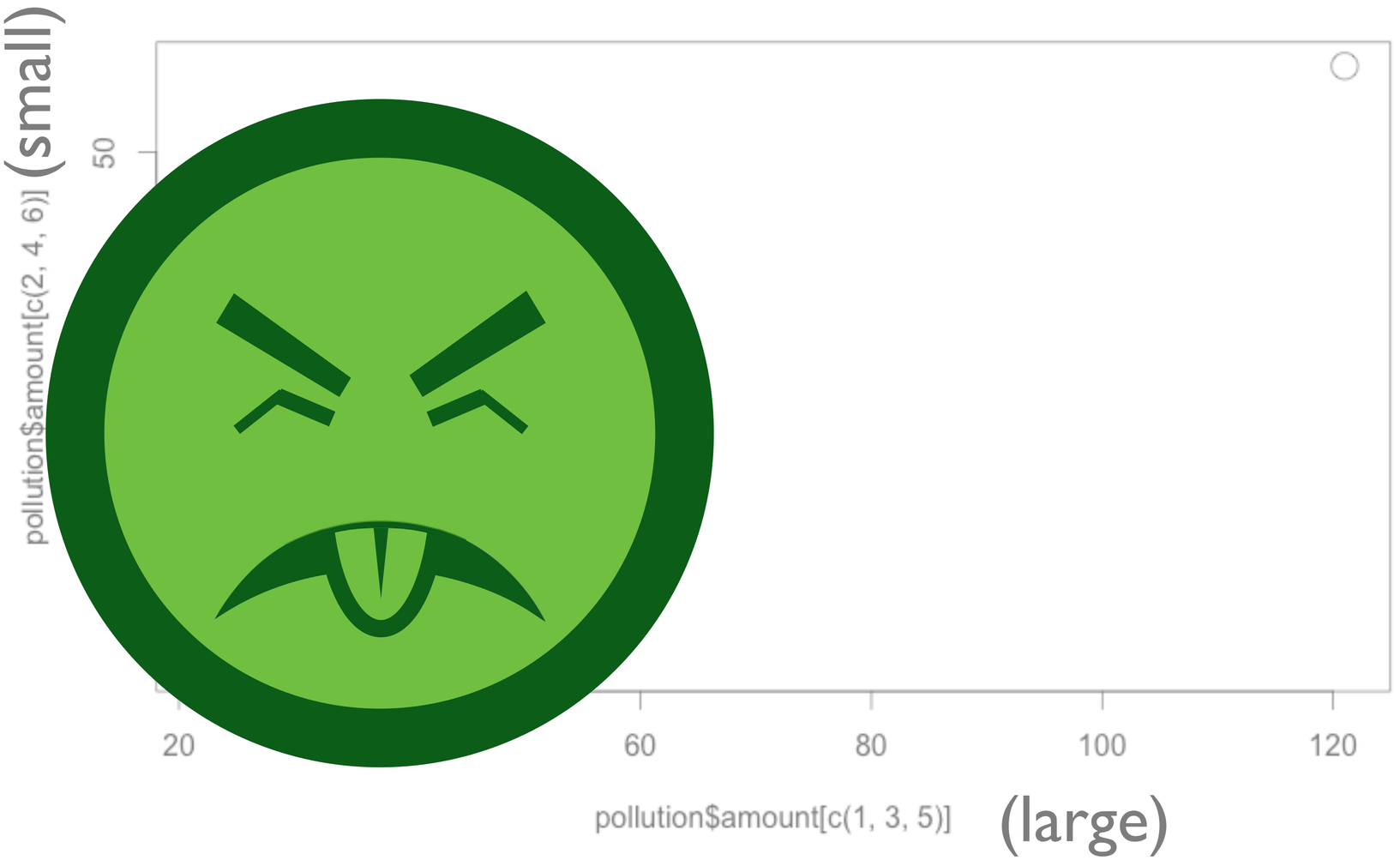
```
pollution$city[1,3,5]
pollution$amount[1,3,5]
pollution$amount[2,4,6]
```

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21




```
plot(storms$date, storms$wind)
```

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




```
plot(pollution$amount[c(1,3,5)], pollution$amount[c(2,4,6)])
```

pollution

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

```
plot(pollution$amount[c(1,3,5)], pollution$amount[c(2,4,6)])
```

pollution

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

city, large, small

```
plot(pollution$amount[c(1,2,5)], pollution$amount[c(2,4,6)])
```

pollution

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

city, large, small

```
plot(pollution$amount[c(1,3,5)], pollution$amount[c(2,4,6)])
```

```
# devtools::install_github("rstudio/EDAWR")
library(EDAWR)
```

storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

```
storms$storm
storms$wind
storms$pressure
storms$date
```

cases

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

```
cases$country
names(cases)[-1]
unlist(cases[1:3, 2:4])
```

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

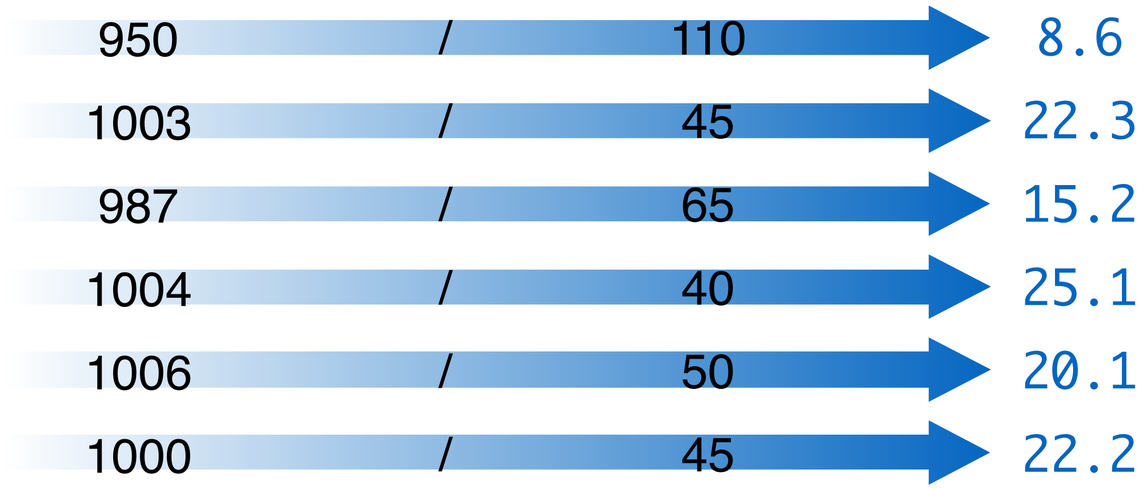
```
pollution$city[1,3,5]
pollution$amount[1,3,5]
pollution$amount[2,4,6]
```

$$\text{ratio} = \frac{\text{pressure}}{\text{wind}}$$

storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

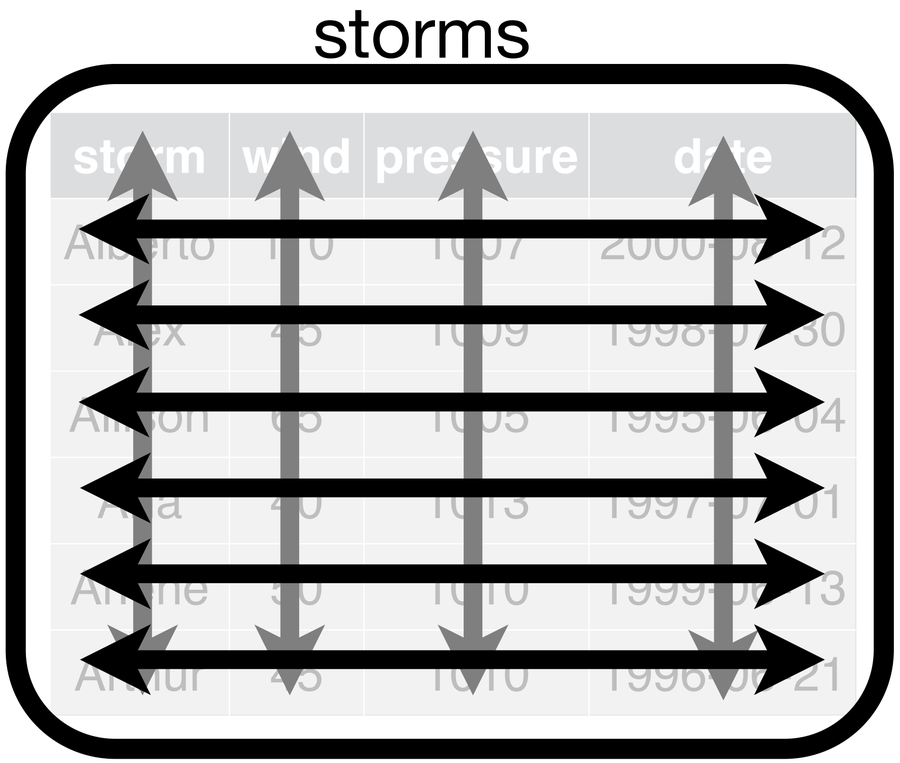
```
storms$pressure / storms$wind
```



Data sets come in many formats

…but R prefers just one.

Tidy data



1. Each **variable** is saved in its own **column**.

2. Each **observation** is saved in its own **row**.

3. Each "type" of observation stored in a **single table** (here, storms).

Recap: Tidy data

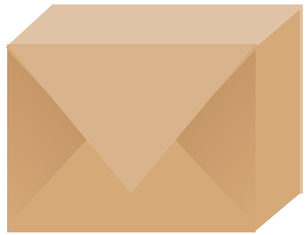
123

Variables in columns, observations in rows, each type in a table

Easy to access variables

Automatically preserves observations

tidyr



A package that reshapes the layout of tables.

Two main functions: **gather()** and **spread()**

```
# install.packages("tidyr")
library(tidyr)
?gather
?spread
```

tb2

Tuberculosis **counts** by country collected by the WHO for the *Global Tuberculosis Report*

```
tb2 <- tb %>%
  mutate(cases = child + adult +
           elderly) %>%
  select(country:sex, cases) %>%
  filter(!is.na(cases)) %>%
  group_by(country, year) %>%
  summarise(cases = sum(cases)) %>%
  ungroup()
```

World Health Organization

tb2

Tuberculosis **counts** by country collected by the WHO for the *Global Tuberculosis Report*

(1,691 x 3)

country	year	cases
Afghanistan	1997	128
Afghanistan	1998	1778
Afghanistan	1999	745
Afghanistan	2000	2666
Afghanistan	2001	4639
Afghanistan	2002	6509

tb2

Tuberculosis **rates** by country collected by the WHO for the *Global Tuberculosis Report*

$$rate = \frac{cases}{population} \times 10000$$

(1,691 x 3)

country	year	cases
Afghanistan	1997	128
Afghanistan	1998	1778
Afghanistan	1999	745

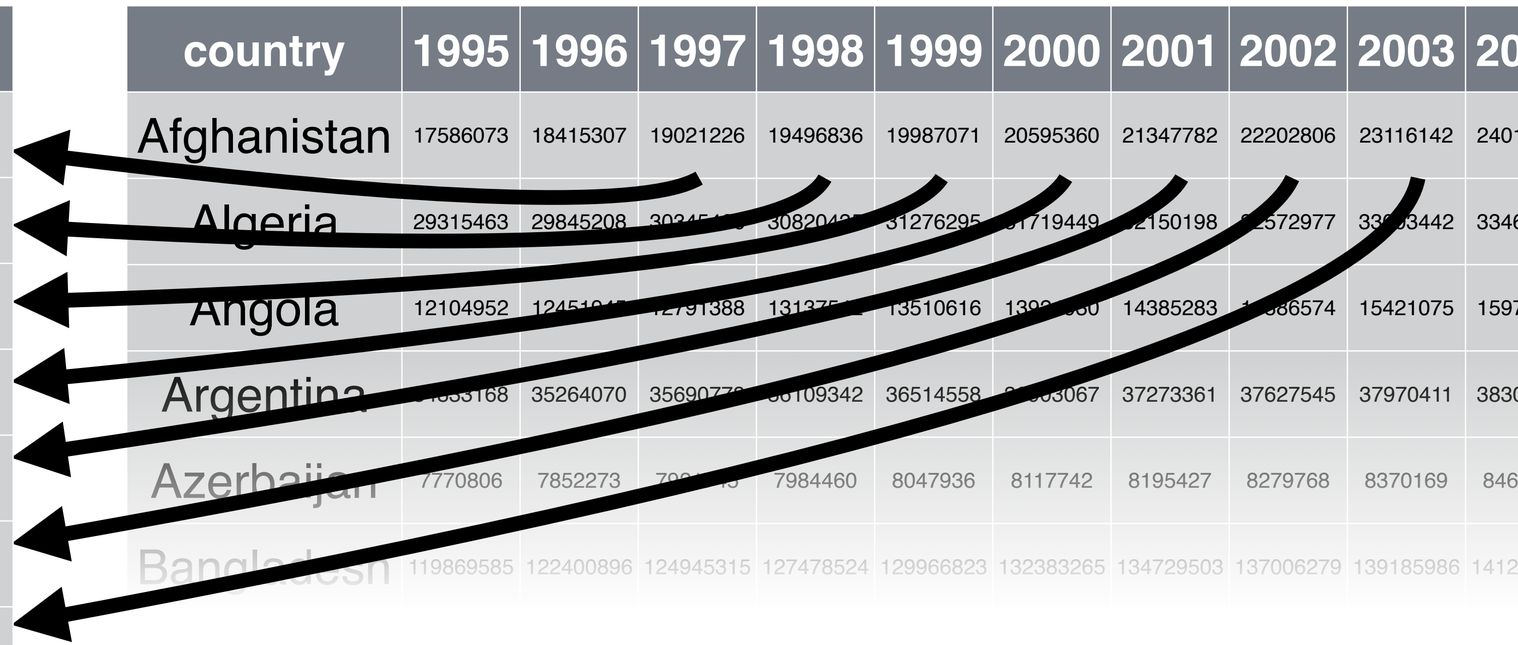
population

```
# library(EDAWR)
View(population)
```

Strategy

1. Tidy the population data set

country	year	population
Afghanistan	1997	19021226
Afghanistan	1998	19496836
Afghanistan	1999	19987071
Afghanistan	2000	20595360
Afghanistan	2001	21347782
Afghanistan	2002	22202806
Afghanistan	2003	23116142

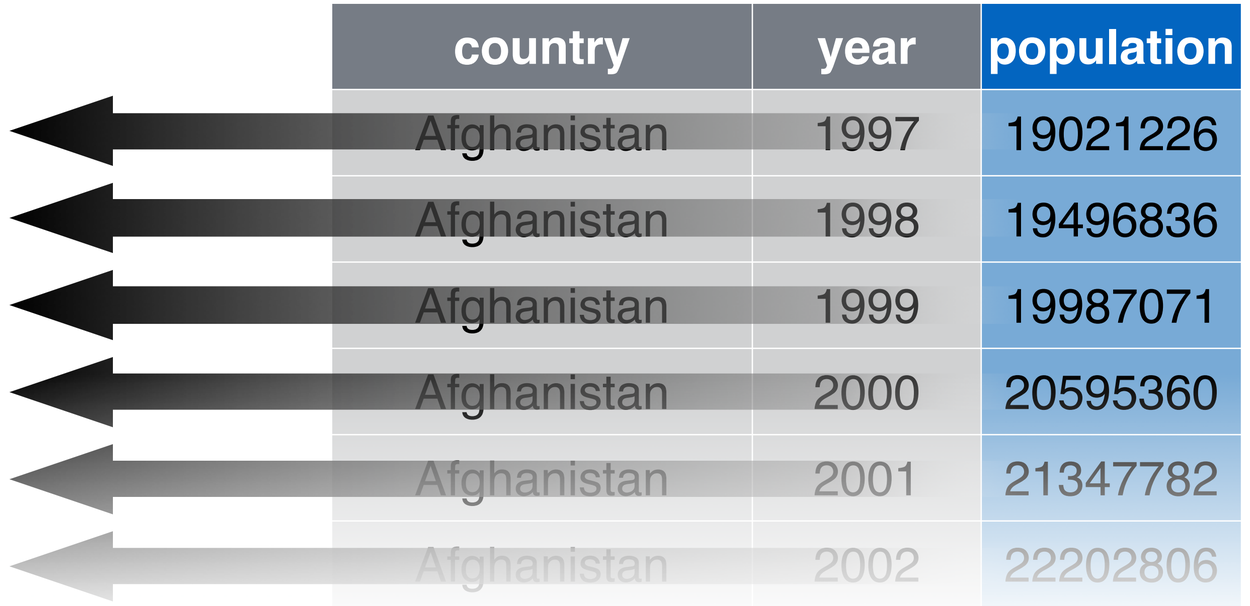


country	1995	1996	1997	1998	1999	2000	2001	2002	2003	20
Afghanistan	17586073	18415307	19021226	19496836	19987071	20595360	21347782	22202806	23116142	240
Algeria	29315463	29845208	30345	3082043	31276295	31719449	32150198	32572977	33003442	334
Angola	12104952	12451045	12791388	131375	13510616	1392	14385283	14886574	15421075	159
Argentina	34533168	35264070	3569077	36109342	36514558	3	37273361	37627545	37970411	383
Azerbaijan	7770806	7852273	79	7984460	8047936	8117742	8195427	8279768	8370169	846
Bangladesh	119869585	122400896	124945315	127478524	129966823	132383265	134729503	137006279	139185986	1412

Strategy

1. Tidy the population data set
2. Join the population values to the tb2 data set

country	year	cases	population
Afghanistan	1997	128	19021226
Afghanistan	1998	1778	19496836
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Afghanistan	2001	4639	21347782
Afghanistan	2002	6509	22202806



country	year	population
Afghanistan	1997	19021226
Afghanistan	1998	19496836
Afghanistan	1999	19987071
Afghanistan	2000	20595360
Afghanistan	2001	21347782
Afghanistan	2002	22202806

Strategy

1. Tidy the population data set
2. Join the population values to the tb2 data set
3. Use mutate() to calculate the rate from cases and population.

country	year	cases	population	rate
Afghanistan	1997	128	19021226	0.07
Afghanistan	1998	1778	19496836	0.91
Afghanistan	1999	745	19987071	0.37
Afghanistan	2000	2666	20595360	1.29
Afghanistan	2001	4639	21347782	2.17
Afghanistan	2002	6509	22202806	2.93
Afghanistan	2003	6528	23116142	2.82

Your Turn

If you do not have tb2, recreate it now to use in the next sections.

```
tb2 <- tb %>%
  mutate(cases = child + adult + elderly) %>%
  select(country:sex, cases) %>%
  filter(!is.na(cases)) %>%
  group_by(country, year) %>%
  summarise(cases = sum(cases)) %>%
  ungroup()
```

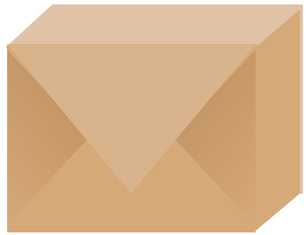
Reshape the layout of your data

country	year	population
Afghanistan	1997	19021226
Afghanistan	1998	19496836
Afghanistan	1999	19987071
Afghanistan	2000	20595360
Afghanistan	2001	21347782
Afghanistan	2002	22202806
Afghanistan	2003	23116142
Afghanistan	2004	24018682
Afghanistan	2005	24860855
Afghanistan	2006	25631282
Afghanistan	2007	26349243

country	1995	1996	1997	1998	1999	2000	2001	2002	2003	20
Afghanistan	17586073	18415307	19021226	19496836	19987071	20595360	21347782	22202806	23116142	240
Algeria	29315463	29845208	30345	3082	31276295	31719449	32150198	32572977	33003442	334
Angola	12104952	12451	12791388	131375	13510616	139	14385283	14886574	15421075	159
Argentina	33	35264070	35690	36109342	36514558	3	37273361	37627545	37970411	383
Azerbaijan	7770806	7852273	79	7984460	8047936	8117742	8195427	8279768	8370169	846
Bangladesh	119869585	122400896	124945315	127478524	129966823	132383265	134729503	137006279	139185986	1412

This will require more than mutate() and summarise()

tidyr



A package that reshapes the layout of tables.

Two main functions: **gather()** and **spread()**

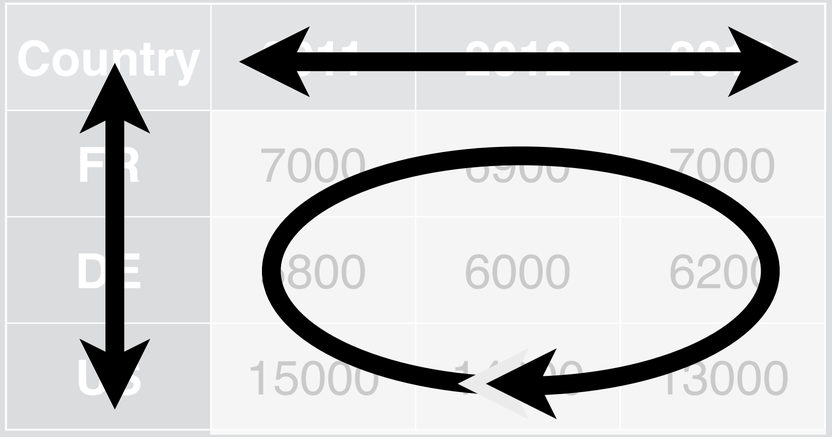
```
# install.packages("tidyr")
library(tidyr)
?gather
?spread
?separate
?unite
```

Your Turn

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*

cases

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

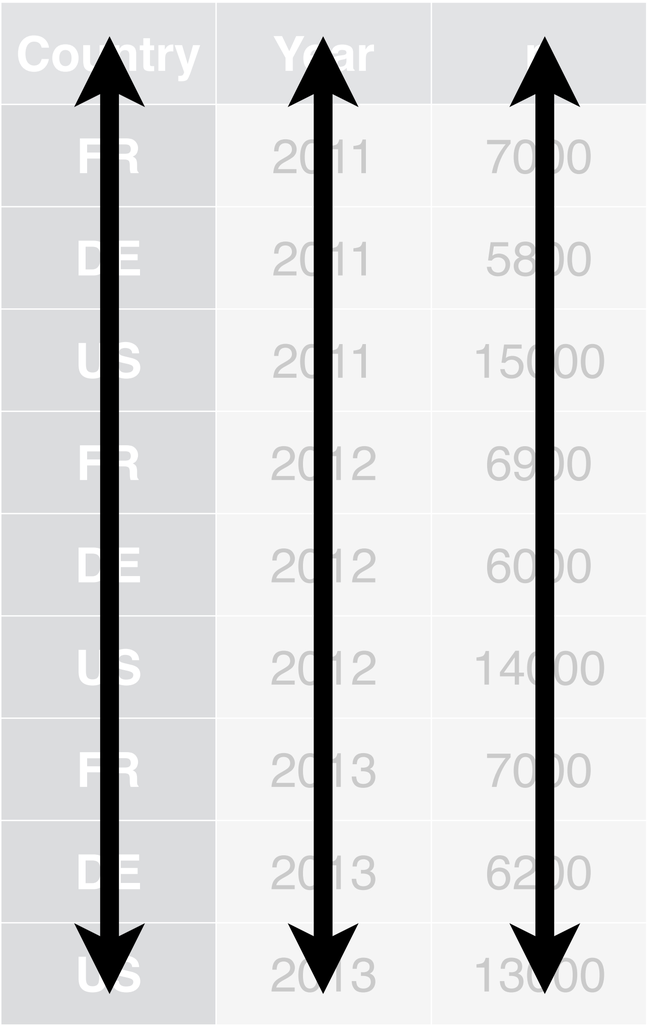
Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

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

key (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

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()

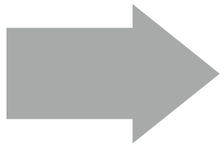
Collapses multiple columns into two columns:

1. a **key** column that contains the former column names
2. a **value** column that contains the former column cells

```
gather(cases, "year", "n", 2:4, convert = TRUE)
```

- `cases`: data frame to reshape
- `"year"`: name of the new key column (a character string)
- `"n"`: name of the new value column (a character string)
- `2:4`: names or numeric indexes of columns to collapse

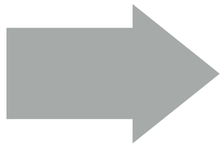
```
##   country  2011  2012  2013
## 1      FR  7000  6900  7000
## 2      DE  5800  6000  6200
## 3      US 15000 14000 13000
```



```
##    country year     n
## 1       FR 2011  7000
## 2       DE 2011  5800
## 3       US 2011 15000
## 4       FR 2012  6900
## 5       DE 2012  6000
## 6       US 2012 14000
## 7       FR 2013  7000
## 8       DE 2013  6200
## 9       US 2013 13000
```

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

```
##   country  2011  2012  2013
## 1      FR  7000  6900  7000
## 2      DE  5800  6000  6200
## 3      US 15000 14000 13000
```



```
##   country year     n
## 1      FR 2011  7000
## 2      DE 2011  5800
## 3      US 2011 15000
## 4      FR 2012  6900
## 5      DE 2012  6000
## 6      US 2012 14000
## 7      FR 2013  7000
## 8      DE 2013  6200
## 9      US 2013 13000
```

Converts numbers in the keys column from factors to numerics

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

Your Turn

Use `gather()` and `arrange()` to build the data set on the right from the original tb data set (`EDAWR::tb`)

EDAWR::tb

country	year	sex	child	adult	elderly
Afghanistan	1995	female	NA	NA	NA
Afghanistan	1995	male	NA	NA	NA
Afghanistan	1996	female	NA	NA	NA
Afghanistan	1996	male	NA	NA	NA
Afghanistan	1997	female	5	96	1
Afghanistan	1997	male	0	26	0
Afghanistan	1998	female	45	1142	20
Afghanistan	1999	male	30	500	41

country	year	sex	age	cases
Afghanistan	1995	female	child	NA
Afghanistan	1995	female	adult	NA
Afghanistan	1995	female	elderly	NA
Afghanistan	1995	male	child	NA
Afghanistan	1995	male	adult	NA
Afghanistan	1995	male	elderly	NA
Afghanistan	1996	female	child	NA
Afghanistan	1996	female	adult	NA

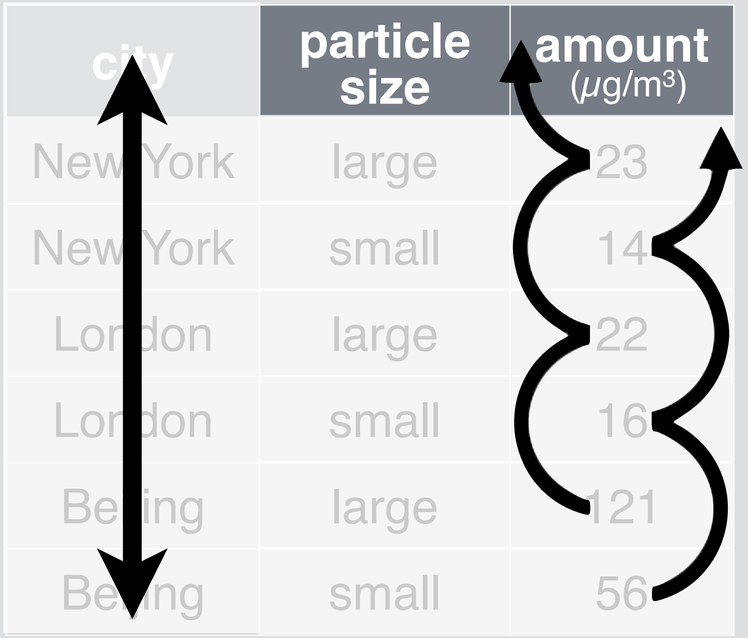
```
EDAWR::tb %>%
  gather("age", "cases", 4:6) %>%
  arrange(country, year, sex, age)

EDAWR::tb %>%
  gather("age", "cases", child:elderly) %>%
  arrange(country, year, sex, age)
```

Your Turn

On a sheet of paper, draw how the pollution data set would look if it had three columns: *city, large, small*

pollution

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

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()

Generates multiple columns from two columns:

1. each unique value in the **key** column becomes a column name
2. each value in the **value** column becomes a cell in the new columns

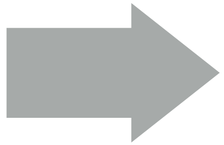
```
spread(pollution, size, amount)
```

data frame to reshape

column to use for keys (new columns names)

column to use for values (new column cells)

```
##       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
```



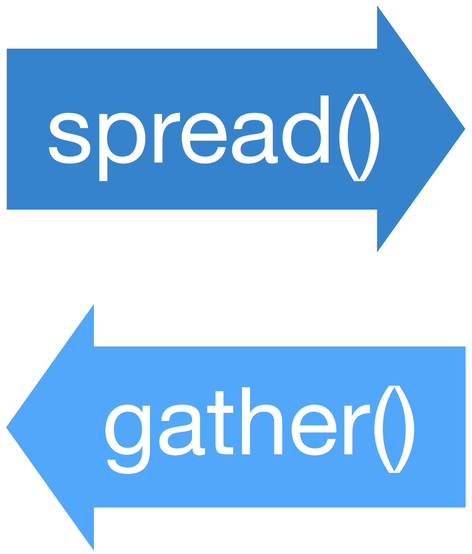
```
##       city large small
## 1  Beijing   121    56
## 2   London    22    16
## 3 New York    23    14
```

```
pollution %>% spread(size, 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

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



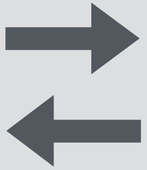
city	large	small
New York	23	14
London	22	16
Beijing	121	56

Your Turn

Use `spread()` to turn `pollution` into the data set on the right. Then use `gather()` to turn it back into the data set on the left.

pollution

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



pollution2

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

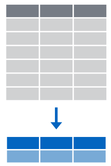
```
pollution2 <- pollution %>% spread(size, amount)
##       city large small
## 1  Beijing   121    56
## 2   London    22    16
## 3 New York    23    14
```

```
pollution2 %>% gather("size", "amount", 2:3)
##       city  size amount
## 1  Beijing large    121
## 2   London large     22
## 3 New York large     23
## 4  Beijing small     56
## 5   London small     16
## 6 New York small     14
```

Data manipulation tool kit

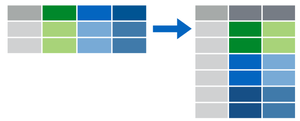
build **variables** from **variables**

dplyr::**mutate()**



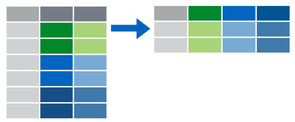
build *observations* from *observations*

dplyr::**summarise()**



build *observations* from **variables**

tidyr::**gather()**



build **variables** from *observations*

tidyr::**spread()**

Separate and unite variables

unite() and separate()

There are three more variables hidden in storms:

storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

- Year
- Month
- Day

separate()

Separate splits a column by a character string separator.

```
separate(storms, date, c("year", "month", "day"), sep = "-")
```

storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

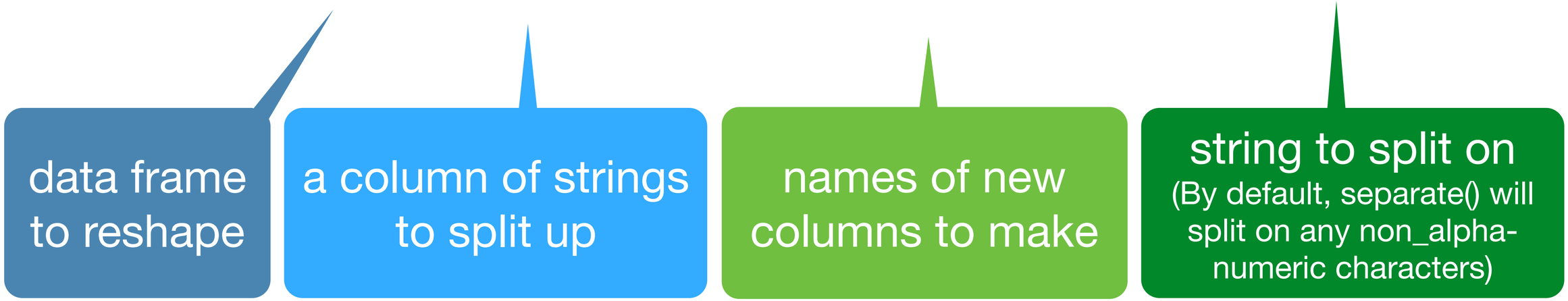
storms2

storm	wind	pressure	year	month	day
Alberto	110	1007	2000	08	12
Alex	45	1009	1998	07	30
Allison	65	1005	1995	06	04
Ana	40	1013	1997	07	1
Arlene	50	1010	1999	06	13
Arthur	45	1010	1996	06	21

separate()

Separate splits a column by a character string separator.

```
separate(storms, date, c("year", "month", "day"), sep = "-")
```



unite()

Unite unites columns into a single column.

```
unite(storms2, "date", year, month, day, sep = "-")
```

storms2

storm	wind	pressure	year	month	day
Alberto	110	1007	2000	08	12
Alex	45	1009	1998	07	30
Allison	65	1005	1995	06	04
Ana	40	1013	1997	07	1
Arlene	50	1010	1999	06	13
Arthur	45	1010	1996	06	21

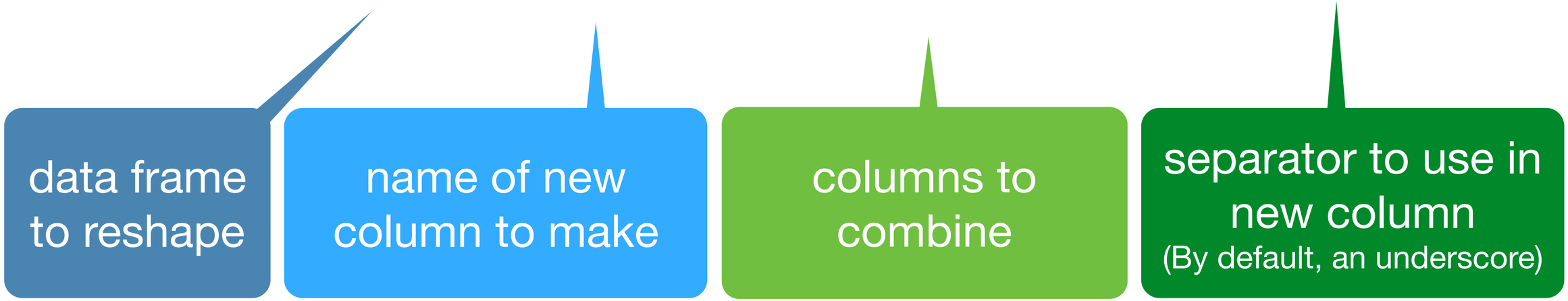
storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

unite()

Unite unites columns into a single column.

```
unite(storms2, "date", year, month, day, sep = "-")
```



Your Turn

Use `separate()` and then `unite()` to change how storms codes date, as below.

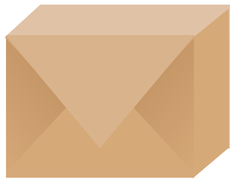
storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

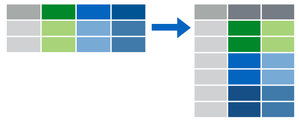
storm	wind	pressure	date
Alberto	110	1007	08/12/2000
Alex	45	1009	07/30/1998
Allison	65	1005	06/04/1995
Ana	40	1013	07/01/1997
Arlene	50	1010	06/13/1999
Arthur	45	1010	06/21/1996

```
storms %>%
  separate(date, c("year", "month", "day")) %>%
  unite("date", month, day, year, sep = "/")
```

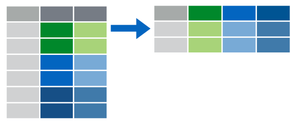
Recap: tidyr



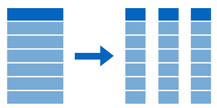
A package that reshapes the layout of data sets.



Make observations from variables with `gather()`



Make variables from observations with `spread()`



Split and merge columns with `unite()` and `separate()`

Data Science for Data Wranglers Part 4:

Choosing a format

```
tb_alt <- tb %>%
  gather("age", "n", 4:6)
```

country	year	sex	child	adult	elderly
Afghanistan	1995	female	NA	NA	NA
Afghanistan	1995	male	NA	NA	NA
Afghanistan	1996	female	NA	NA	NA
Afghanistan	1996	male	NA	NA	NA
Afghanistan	1997	female	5	96	1
Afghanistan	1997	male	0	26	0

country	year	sex	age	n
Afghanistan	1996	female	adult	NA
Afghanistan	1996	female	child	NA
Afghanistan	1996	female	elderly	NA
Afghanistan	1996	male	adult	NA
Afghanistan	1996	male	child	NA
Afghanistan	1996	male	elderly	NA
Afghanistan	1997	female	adult	96
Afghanistan	1997	female	child	5
Afghanistan	1997	female	elderly	1
Afghanistan	1997	male	adult	25

Which is tidy?

country	year	sex	child	adult	elderly
Afghanistan	1995	female	NA	NA	NA
Afghanistan	1995	male	NA	NA	NA
Afghanistan	1996	female	NA	NA	NA
Afghanistan	1996	male	NA	NA	NA
Afghanistan	1997	female	5	96	1
Afghanistan	1997	male	0	26	0

country	year	sex	age	n
Afghanistan	1996	female	adult	NA
Afghanistan	1996	female	child	NA
Afghanistan	1996	female	elderly	NA
Afghanistan	1996	male	adult	NA
Afghanistan	1996	male	child	NA
Afghanistan	1996	male	elderly	NA
Afghanistan	1997	female	adult	96
Afghanistan	1997	female	child	5
Afghanistan	1997	female	elderly	1
Afghanistan	1997	male	adult	25

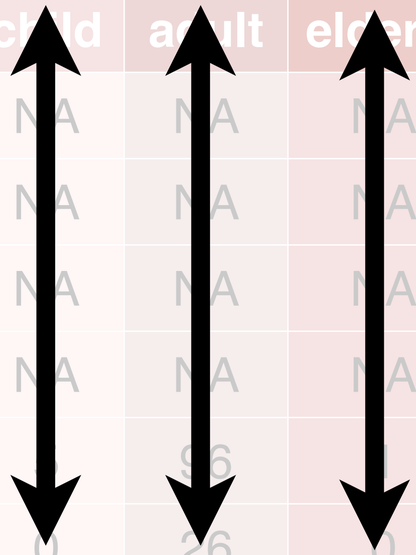
Which is tidy?

country	year	sex	child	adult	elderly
Afghanistan	1995	female	NA	NA	NA
Afghanistan	1995	male	NA	NA	NA
Afghanistan	1996	female	NA	NA	NA
Afghanistan	1996	male	NA	NA	NA
Afghanistan	1997	female	5	96	1
Afghanistan	1997	male	0	26	0

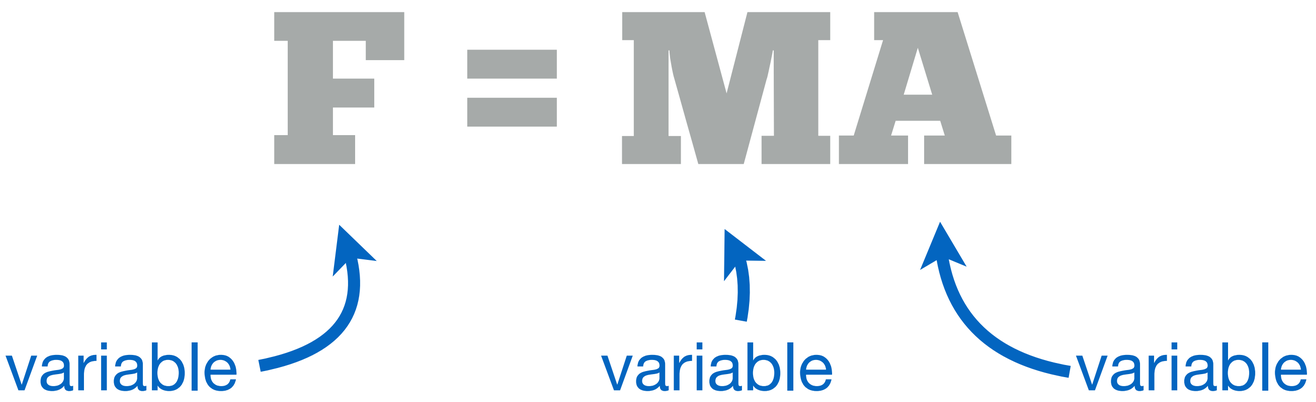
country	year	sex	age	n
Afghanistan	1996	female	adult	NA
Afghanistan	1996	female	child	NA
Afghanistan	1996	female	elderly	NA
Afghanistan	1996	male	adult	NA
Afghanistan	1996	male	child	NA
Afghanistan	1996	male	elderly	NA
Afghanistan	1997	female	adult	96
Afghanistan	1997	female	child	5
Afghanistan	1997	female	elderly	1
Afghanistan	1997	male	adult	25

Which is tidy?

country	year	sex	child	adult	elderly
Afghanistan	1995	female	NA	NA	NA
Afghanistan	1995	male	NA	NA	NA
Afghanistan	1996	female	NA	NA	NA
Afghanistan	1996	male	NA	NA	NA
Afghanistan	1997	female	5	96	1
Afghanistan	1997	male	0	26	0



country	year	sex	age	n
Afghanistan	1996	female	adult	NA
Afghanistan	1996	female	child	NA
Afghanistan	1996	female	elderly	NA
Afghanistan	1996	male	adult	NA
Afghanistan	1996	male	child	NA
Afghanistan	1996	male	elderly	NA
Afghanistan	1997	female	adult	96
Afghanistan	1997	female	child	5
Afghanistan	1997	female	elderly	1
Afghanistan	1997	male	adult	25



Variable - A quantity, quality, or property that you can measure.

Variable - A quantity, quality, or property that you can measure.

- ✔ **age** - You can measure the age of an individual or the age of a group of people who all have the same age
- ✔ **child** - You can measure the number of cases of children with TB reported in a group of people
- ✔ **adult** - You can measure the number of cases of adults with TB reported in a group of people
- ✔ **elderly** - You can measure the number of cases of elderly people with TB reported in a group of people

Variable - A quantity, quality, or property that you can measure.

age - You can measure the age of an individual or the age of a group of people who all have the same age

child - You can measure the number of cases of children with TB reported in **a group of people**

adult - You can measure the number of cases of adults with TB reported in **a group of people**

elderly - You can measure the number of cases of elderly people with TB reported in **a group of people**

Variable - A quantity, quality, or property that you can measure.

age - You can measure the age of an individual or the age of **a group of people who all have the same age**

child - You can measure the number of cases of children with TB reported in **a group of people**

adult - You can measure the number of cases of adults with TB reported in **a group of people**

elderly - You can measure the number of cases of elderly people with TB reported in **a group of people**

What is a variable (and what is tidy) depends on your unit of analysis

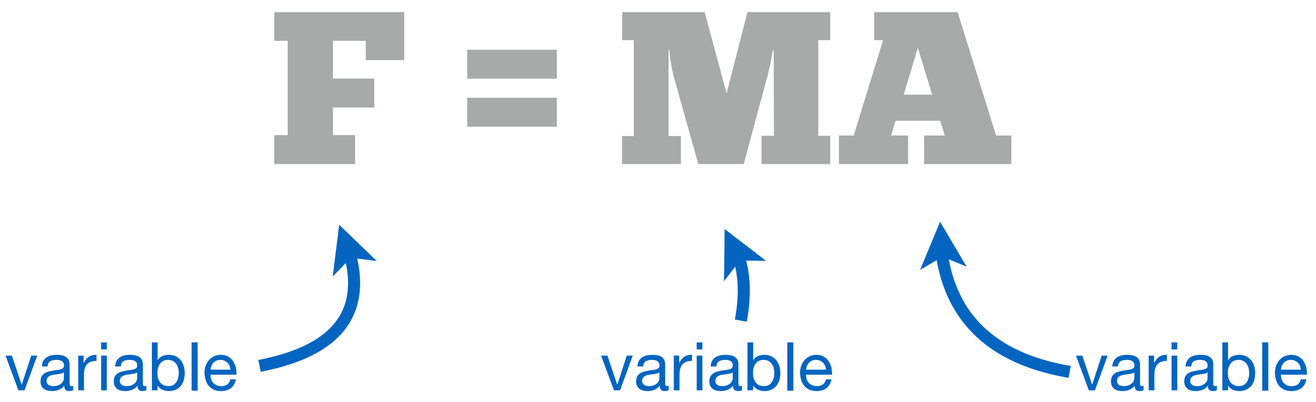
Unit of Analysis
groups of people of the same gender in the same country in the same year

country	year	sex	child	adult	elderly
Afghanistan	1995	female	NA	NA	NA
Afghanistan	1995	male	NA	NA	NA
Afghanistan	1996	female	NA	NA	NA
Afghanistan	1996	male	NA	NA	NA
Afghanistan	1997	female	5	96	1
Afghanistan	1997	male	0	26	0

cases = child + adult + elderly

Unit of Analysis
groups of people of the same gender **and age** in the same country in the same year

country	year	sex	age	n
Afghanistan	1996	female	adult	NA
Afghanistan	1996	female	child	NA
Afghanistan	1996	female	elderly	NA
Afghanistan	1996	male	adult	NA
Afghanistan	1996	male	child	NA
Afghanistan	1996	male	elderly	NA
Afghanistan	1997	female	adult	96
Afghanistan	1997	female	child	5
Afghanistan	1997	female	elderly	1
Afghanistan	1997	male	adult	25



cases = child + adult + elderly

Unit of Analysis: All measured on a group of people (who live in the same country and have the same sex) in the same year

cases = child + adult + elderly

Unit of Analysis: All measured on a group of people (who live in the same country and have the same sex) in the same year

Goal: calculate cases, where cases = child + adult + elderly

country	year	sex	child	adult	elderly
Afghanistan	1995	female	NA	NA	NA
Afghanistan	1995	male	NA	NA	NA
Afghanistan	1996	female	NA	NA	NA
Afghanistan	1996	male	NA	NA	NA
Afghanistan	1997	female	5	96	1
Afghanistan	1997	male	0	26	0

```
tb %>%
  mutate(
    cases = child + adult + elderly)
```

country	year	sex	age	n
Afghanistan	1996	female	adult	NA
Afghanistan	1996	female	child	NA
Afghanistan	1		elderly	NA
Afghanistan	1		adult	NA
Afghanistan	1		child	NA
Afghanistan	1		elderly	NA
Afghanistan	1		adult	96
Afghanistan	1997	female	child	5
Afghanistan	1997	female	elderly	1
Afghanistan	1997	male	adult	25

Goal: calculate cases, where cases = child + adult + elderly

```
tb_alt %>%
  group_by(country, year, sex) %>%
  summarise(cases = sum(n))
```

country	year	sex	age	n
Afghanistan	1996	female	adult	NA
Afghanistan	1996	female	child	NA
Afghanistan	1		elderly	NA
Afghanistan			adult	NA
Afghanistan			child	NA
Afghanistan			elderly	NA
Afghanistan	19	le	adult	96
Afghanistan	1997	female	child	5
Afghanistan	1997	female	elderly	1
Afghanistan	1997	male	adult	25

Which is tidy?

country	year	sex	n
Afghanistan	1999	female	1
Afghanistan	1999	male	1
Afghanistan	2000	female	1
Afghanistan	2000	male	1
Brazil	1999	female	2
Brazil	1999	male	2
Brazil	2000	female	2
Brazil	2000	male	2
China	1999	female	3
China	1999	male	3
China	2000	female	3
China	2000	male	3

country	year	n
Afghanistan	1999	2
Afghanistan	2000	2
Brazil	1999	4
Brazil	2000	4
China	1999	6
China	2000	6

country	n
Afghanistan	4
Brazil	8
China	12

n
24

Which is tidy?

country	year	sex	n
Afghanistan	1999	female	1
Afghanistan	1999	male	1
Afghanistan	2000	female	1
Afghanistan	2000	male	1
Brazil	1999	female	2
Brazil	1999	male	2
Brazil	2000	female	2
Brazil	2000	male	2
China	1999	female	3
China	1999	male	3
China	2000	female	3
China	2000	male	3

country	year	n
Afghanistan	1999	2
Afghanistan	2000	2
Brazil	1999	4
Brazil	2000	4
China	1999	6
China	2000	6

country	n
Afghanistan	4
Brazil	8
China	12

n
24

Goal: test a hypothesis about the total n per country per year

Which is tidy?

country	year	sex	n
Afghanistan	1999	female	1
Afghanistan	1999	male	1
Afghanistan	2000	female	1
Afghanistan	2000	male	1
Brazil	1999	female	2
Brazil	1999	male	2
Brazil	2000	female	2
Brazil	2000	male	2
China	1999	female	3
China	1999	male	3
China	2000	female	3
China	2000	male	3

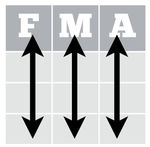
country	year	n
Afghanistan	1999	2
Afghanistan	2000	2
Brazil	1999	4
Brazil	2000	4
China	1999	6
China	2000	6

Goal: test a hypothesis about the total n per country per year

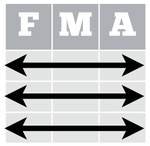
Best format for analysis

Unit of analysis matches the unit of analysis

you wish to study/manipulate/explore



Variables in columns



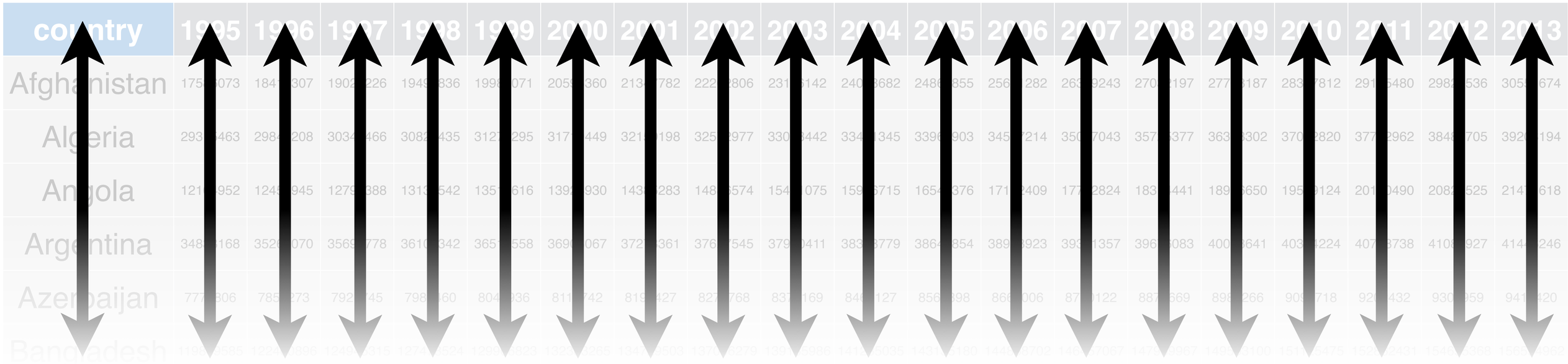
Observations in rows

What is the unit of analysis of population?

country	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Afghanistan	17586073	18415307	19021226	19496836	19987071	20595360	21347782	22202806	23116142	24018682	24860855	25631282	26349243	27032197	27708187	28397812	29105480	29824536	30551674
Algeria	29315463	29845208	30345466	30820435	31276295	31719449	32150198	32572977	33003442	33461345	33960903	34507214	35097043	35725377	36383302	37062820	37762962	38481705	39208194
Angola	12104952	12451945	12791388	13137542	13510616	13924930	14385283	14886574	15421075	15976715	16544376	17122409	17712824	18314441	18926650	19549124	20180490	20820525	21471618
Argentina	34833168	35264070	35690778	36109342	36514558	36903067	37273361	37627545	37970411	38308779	38647854	38988923	39331357	39676083	40023641	40374224	40728738	41086927	41446246
Azerbaijan	7770806	7852273	7921745	7984460	8047936	8117742	8195427	8279768	8370169	8465127	8563398	8665006	8770122	8877669	8986266	9094718	9202432	9308959	9413420
Bangladesh	119869585	122400896	124945315	127478524	129966823	132383265	134729503	137006279	139185986	141235035	143135180	144868702	146457067	147969967	149503100	151125475	152862431	154695368	156594962

What is the unit of analysis of population?

Individual countries



What is the unit of analysis of tb2?

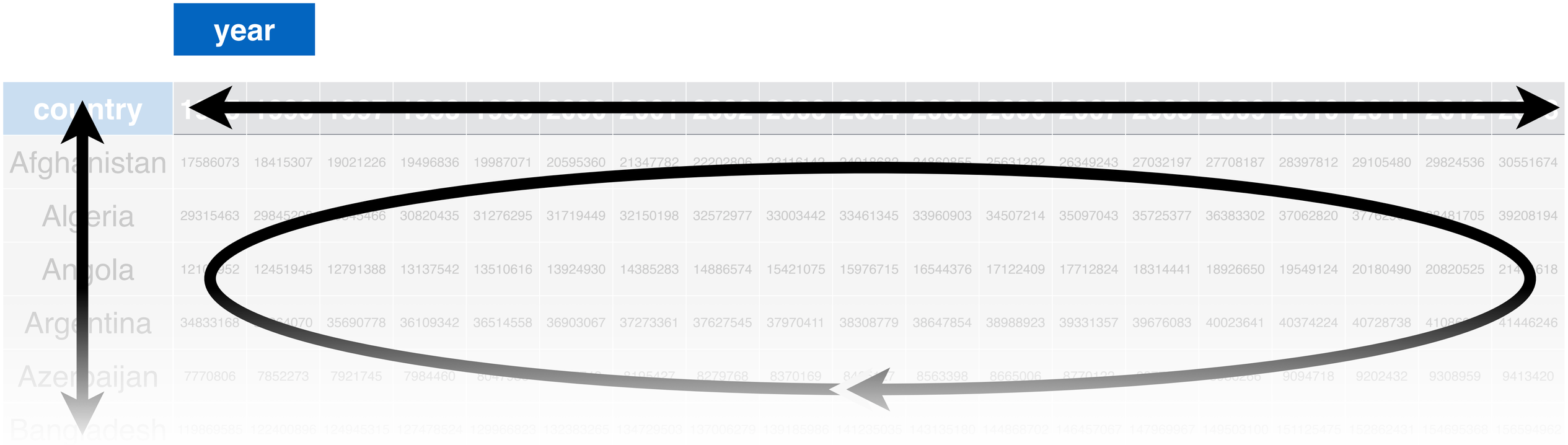
Countries by year

(1,691 x 3)

country	year	cases
Afghanistan	1997	128
Afghanistan	1998	1778
Afghanistan	1999	745
Afghanistan	2000	2666
Afghanistan	2001	4639
Afghanistan	2002	6509

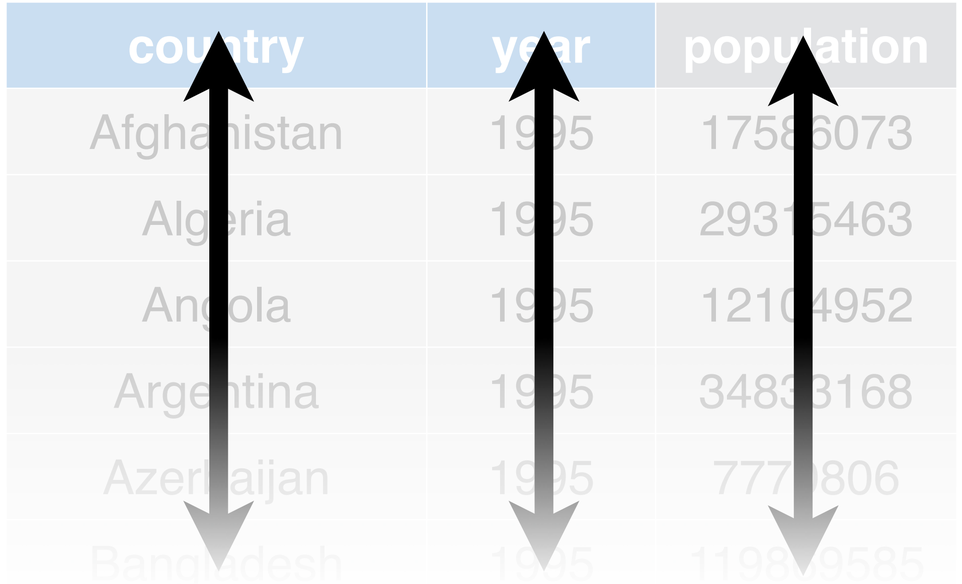
What is the unit of analysis of population?

Countries by year



What is the unit of analysis of population?

Countries by year



country	year	cases
Afghanistan	1997	128
Afghanistan	1998	1778
Afghanistan	1999	745
Afghanistan	2000	2666
Afghanistan	2001	4639
Afghanistan	2002	6509

country	year	population
Afghanistan	1995	17586073
Algeria	1995	29315463
Angola	1995	12104952
Argentina	1995	34833168
Azerbaijan	1995	7770806
Bangladesh	1995	119869585

multiple tables

Which is tidy?

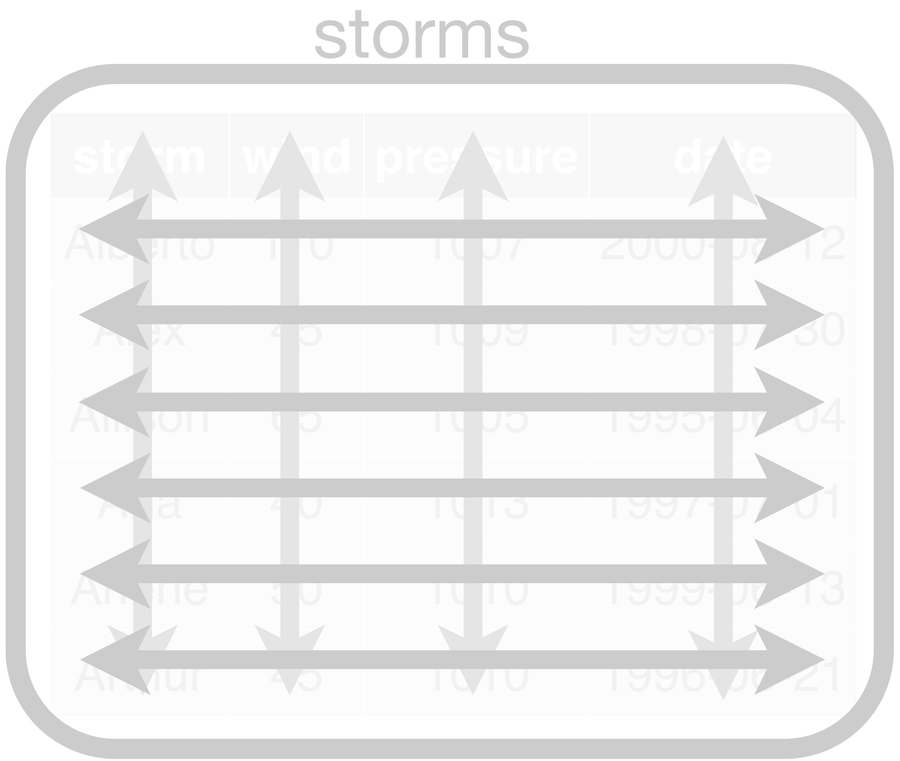
employee id	country	income
0001	Afghanistan	$100000
0002	Afghanistan	$100000
0003	Brazil	$100000
0004	Brazil	$100000
0005	China	$100000
0006	China	$100000

+

country	tax rate
Afghanistan	0.04
Brazil	0.12
China	0.50

country	employee id	income	tax rate
Afghanistan	0001	$100000	0.04
Afghanistan	0002	$100000	0.04
Brazil	0003	$100000	0.12
Brazil	0004	$100000	0.12
China	0005	$100000	0.50
China	0006	$100000	0.50

Tidy data



1. Each **variable** is saved in its own **column**.

2. Each **observation** is saved in its own **row**.

3. Each "type" of observation stored in a **single table**.

Goal: Net income = income - income * tax rate

employee id	country	income
0001	Afghanistan	$100000
0002	Afghanistan	$100000
0003	Brazil	$100000
0004	Brazil	$100000
0005	China	$100000
0006	China	$100000

+

country	tax rate
Afghanistan	0.04
Brazil	0.12
China	0.50

country	employee id	income	tax rate
Afghanistan	0001	$100000	0.04
Afghanistan	0002	$100000	0.04
Brazil	0003	$100000	0.12
Brazil	0004	$100000	0.12
China	0005	$100000	0.50
China	0006	$100000	0.50

```
data %>%
  mutate(
    net = income - income * tax rate)
```

Goal: Net income = income - income * tax rate

employee id	country
0001	Afgha
0002	Afgh
0003	
0004	
0005	C
0006	Chin

country	tax rate
Afghanistan	0.04
Brazil	0.12
China	0.50

country	employee id	income	tax rate
Afghanistan	0001	$100000	0.04
Afghanistan	0002	$100000	0.04
Brazil	0003	$100000	0.12
Brazil	0004	$100000	0.12
China	0005	$100000	0.50
China	0006	$100000	0.50

```
data %>%
  mutate(
    net = income - income * tax rate)
```

Normalized data

employee id	country	income
0001	Afghanistan	$100000
0002	Afghanistan	$100000
0003	Brazil	$100000
0004	Brazil	$100000
0005	China	$100000
0006	China	$100000

+

country	tax rate
Afghanistan	0.04
Brazil	0.12
China	0.50

Data reduced to separate tables to prevent redundancy

Easy to store and update

Hard to analyze

Single table

country	employee id	income	tax rate
Afghanistan	0001	$100000	0.04
Afghanistan	0002	$100000	0.04
Brazil	0003	$100000	0.12
Brazil	0004	$100000	0.12
China	0005	$100000	0.50
China	0006	$100000	0.50

Data combined in a single table for efficiency

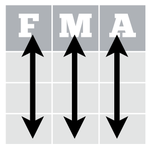
Hard to store and update

Easy to analyze

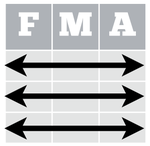
Recap: Best format for analysis

Unit of analysis matches the unit of analysis

you wish to study/manipulate/explore



Variables in columns

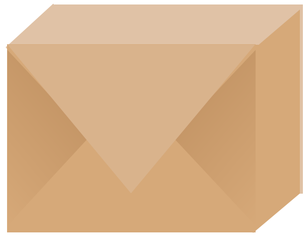


Observations in rows

Single table

Combine data sets

dplyr



A package that helps transform tabular data.

```
# install.packages("dplyr")
library(dplyr)
?left_join
?inner_join
?semi_join
?anti_join
?select
?filter
?mutate
?summarise
?group_by
```

simple binds

dplyr::**bind_rows()**

y

x1	x2
A	1
B	2
C	3

+

z

x1	x2
B	2
C	3
D	4

=

x1	x2
A	1
B	2
C	3
B	2
C	3
D	4

```
bind_rows(y, z)
```

dplyr::**bind_cols()**

y

x1	x2
A	1
B	2
C	3

+

z

x1	x2
B	2
C	3
D	4

=

x1	x2	x1	x2
A	1	B	2
B	2	C	3
C	3	D	4

```
bind_cols(y, z)
```

set operations

dplyr::**union()**

y

x1	x2
A	1
B	2
C	3

+

z

x1	x2
B	2
C	3
D	4

=

x1	x2
A	1
B	2
C	3
D	4

```
union(y, z)
```

dplyr::**intersect()**

y

x1	x2
A	1
B	2
C	3

+

z

x1	x2
B	2
C	3
D	4

=

x1	x2
B	2
C	3

```
intersect(y, z)
```

dplyr::**setdiff()**

y

x1	x2
A	1
B	2
C	3

+

z

x1	x2
B	2
C	3
D	4

=

x1	x2
A	1

```
setdiff(y, z)
```

joins

dplyr::**left_join()**

songs

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul
Peggy Sue	Buddy

+

artists

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums

=

song	name	plays
Across the Universe	John	guitar
Come Together	John	guitar
Hello, Goodbye	Paul	bass
Peggy Sue	Buddy	**<NA>**

```
songs %>% left_join(artists, by = "name")
```

dplyr::**left_join()**

songs

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul
Peggy Sue	Buddy

+

artists

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums

=

song	name	plays
Across the Universe	John	guitar
Come Together	John	guitar
Hello, Goodbye	Paul	bass
Peggy Sue	Buddy	**<NA>**

```
songs %>% left_join(artists, by = "name")
```

dplyr::**left_join()**

songs2

song	first	last
Across the Universe	John	Lennon
Come Together	John	Lennon
Hello, Goodbye	Paul	McCartney
Peggy Sue	Buddy	Holly

+

artists2

first	last	plays
George	Harrison	sitar
John	Lennon	guitar
Paul	McCartney	bass
Ringo	Starr	drums
Paul	Simon	guitar
John	Coltrane	sax

=

song	first	last	plays
Across the Universe	John	Lennon	guitar
Come Together	John	Lennon	guitar
Hello, Goodbye	Paul	McCartney	bass
Peggy Sue	Buddy	Holly	**<NA>**

```
songs %>% left_join(artists2, by = c("first", "last"))
```

dplyr::**left_join()**

songs2

song	first	last
Across the Universe	John	Lennon
Come Together	John	Lennon
Hello, Goodbye	Paul	McCartney
Peggy Sue	Buddy	Holly

+

artists2

first	last	plays
George	Harrison	sitar
John	Lennon	guitar
Paul	McCartney	bass
Ringo	Starr	drums
Paul	Simon	guitar
John	Coltrane	sax

=

song	first	last	plays
Across the Universe	John	Lennon	guitar
Come Together	John	Lennon	guitar
Hello, Goodbye	Paul	McCartney	bass
Peggy Sue	Buddy	Holly	**<NA>**

```
songs %>% left_join(artists2, by = c("first", "last"))
```

left_join()

songs

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul
Peggy Sue	Buddy

+

artists

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums

=

song	name	plays
Across the Universe	John	guitar
Come Together	John	guitar
Hello, Goodbye	Paul	bass
Peggy Sue	Buddy	**<NA>**

```
songs %>% left_join(artists, by = "name")
```

right_join()

songs

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul
Peggy Sue	Buddy

+

artists

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums

=

song	name	plays
<NA>	George	sitar
Across the Universe	John	guitar
Come Together	John	guitar
Hello, Goodbye	Paul	bass
<NA>	Ringo	drums

```
songs %>% right_join(artists, by = "name")
```

inner_join()

songs

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul
Peggy Sue	Buddy

+

artists

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums

=

song	name	plays
Across the Universe	John	guitar
Come Together	John	guitar
Hello, Goodbye	Paul	bass

```
songs %>% inner_join(artists, by = "name")
```

full_join()

songs

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul
Peggy Sue	Buddy

+

artists

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums

=

song	name	plays
Across the Universe	John	guitar
Come Together	John	guitar
Hello, Goodbye	Paul	bass
Peggy Sue	Buddy	**<NA>**
<NA>	George	sitar
<NA>	Ringo	drums

```
songs %>% full_join(artists, by = "name")
```

semi_join()

songs

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul
Peggy Sue	Buddy

+

artists

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums

=

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul

```
songs %>% semi_join(artists, by = "name")
```

anti_join()

songs

song	name
Across the Universe	John
Come Together	John
Hello, Goodbye	Paul
Peggy Sue	Buddy

+

artists

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums

=

song	name
Peggy Sue	Buddy

```
songs %>% anti_join(artists, by = "name")
```

RStudio

Simple binds

dplyr::**bind_rows**

dplyr::**bind_cols**

Set operations

dplyr::**union**

dplyr::**intersect**

dplyr::**setdiff**

Matching joins

dplyr::**left_join**

dplyr::**right_join**

dplyr::**inner_join**

dplyr::**full_join**

Filtering joins

dplyr::**semi_join**

dplyr::**anti_join**

```
delays <- flights %>%
  filter(!is.na(arr_delay)) %>%
  group_by(carrier) %>%
  summarise(avg_delay = mean(arr_delay))
```

(16 x 2)

carrier	avg_delay
9E	7.38
AA	0.36
AS	-9.93
B6	9.95
DL	1.64

```
View(airlines)
```

(16 x 2)

carrier	name
9E	Endeavor Air Inc.
AA	American Airlines Inc.
AS	Alaska Airlines Inc.
B6	JetBlue Airways
DL	Delta Air Lines Inc.

Your Turn

Use a dplyr function to combine delays and names to make the data set below.

Then `arrange()` by **avg_delay** to see which airline had the shortest average delays.

(16 x 3)

carrier	avg_delay	name
9E	7.38	Endeavor Air Inc.
AA	0.36	American Airlines Inc.
AS	-9.93	Alaska Airlines Inc.
B6	9.95	JetBlue Airways
DL	1.64	Delta Air Lines Inc.
EV	15.80	ExpressJet Airlines Inc.

```
delays %>%
  left_join(airlines, by = "carrier") %>%
  arrange(avg_delay)
```

(16 x 3)

carrier	avg_delay	name
AS	-9.93	Alaska Airlines Inc.
HA	-6.92	Hawaiian Airlines Inc.
AA	0.36	American Airlines Inc.
DL	1.64	Delta Air Lines Inc.
VX	1.76	Virgin America
US	2.13	US Airways Inc.

Case Study 2: TB rates

tb2

Tuberculosis **counts** by country collected by the WHO for the *Global Tuberculosis Report*

```
tb2 <- tb %>%
  mutate(cases = child + adult +
           elderly) %>%
  select(country:sex, cases) %>%
  filter(!is.na(cases)) %>%
  group_by(country, year) %>%
  summarise(cases = sum(cases)) %>%
  ungroup()
```

World Health Organization

tb2

Tuberculosis **counts** by country collected by the WHO for the *Global Tuberculosis Report*

(1,691 x 3)

country	year	cases
Afghanistan	1997	128
Afghanistan	1998	1778
Afghanistan	1999	745
Afghanistan	2000	2666
Afghanistan	2001	4639
Afghanistan	2002	6509

tb2

Tuberculosis **rates** by country collected by the WHO for the *Global Tuberculosis Report*

$$rate = \frac{cases}{population} \times 10000$$

(1,691 x 3)

country	year	cases
Afghanistan	1997	128
Afghanistan	1998	1778
Afghanistan	1999	745

population

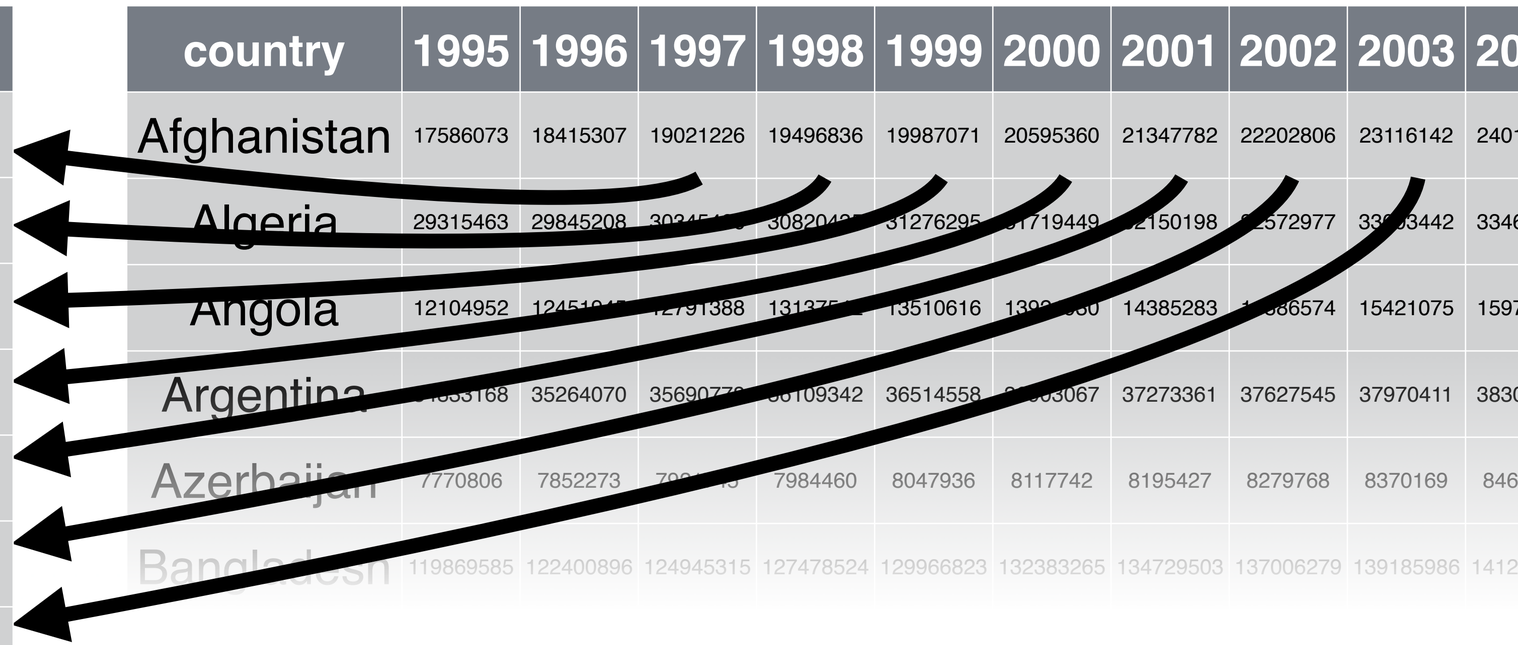
```
# library(EDAWR)
View(population)
```

country	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Afghanistan	17586073	18415307	19021226	19496836	19987071	20595360	21347782	22202806	23116142	24018682	24860855	25631282	26349243	27032197	27708187	28397812	29105480	29824536	30551674
Algeria	29315463	29845208	30345466	30820435	31276295	31719449	32150198	32572977	33003442	33461345	33960903	34507214	35097043	35725377	36383302	37062820	37762962	38481705	39208194
Angola	12104952	12451945	12791388	13137542	13510616	13924930	14385283	14886574	15421075	15976715	16544376	17122409	17712824	18314441	18926650	19549124	20180490	20820525	21471618
Argentina	34833168	35264070	35690778	36109342	36514558	36903067	37273361	37627545	37970411	38308779	38647854	38988923	39331357	39676083	40023641	40374224	40728738	41086927	41446246
Azerbaijan	7770806	7852273	7921745	7984460	8047936	8117742	8195427	8279768	8370169	8465127	8563398	8665006	8770122	8877669	8986266	9094718	9202432	9308959	9413420
Bangladesh	119869585	122400896	124945315	127478524	129966823	132383265	134729503	137006279	139185986	141235035	143135180	144868702	146457067	147969967	149503100	151125475	152862431	154695368	156594962

Strategy

1. Tidy the population data set

country	year	population
Afghanistan	1997	19021226
Afghanistan	1998	19496836
Afghanistan	1999	19987071
Afghanistan	2000	20595360
Afghanistan	2001	21347782
Afghanistan	2002	22202806
Afghanistan	2003	23116142

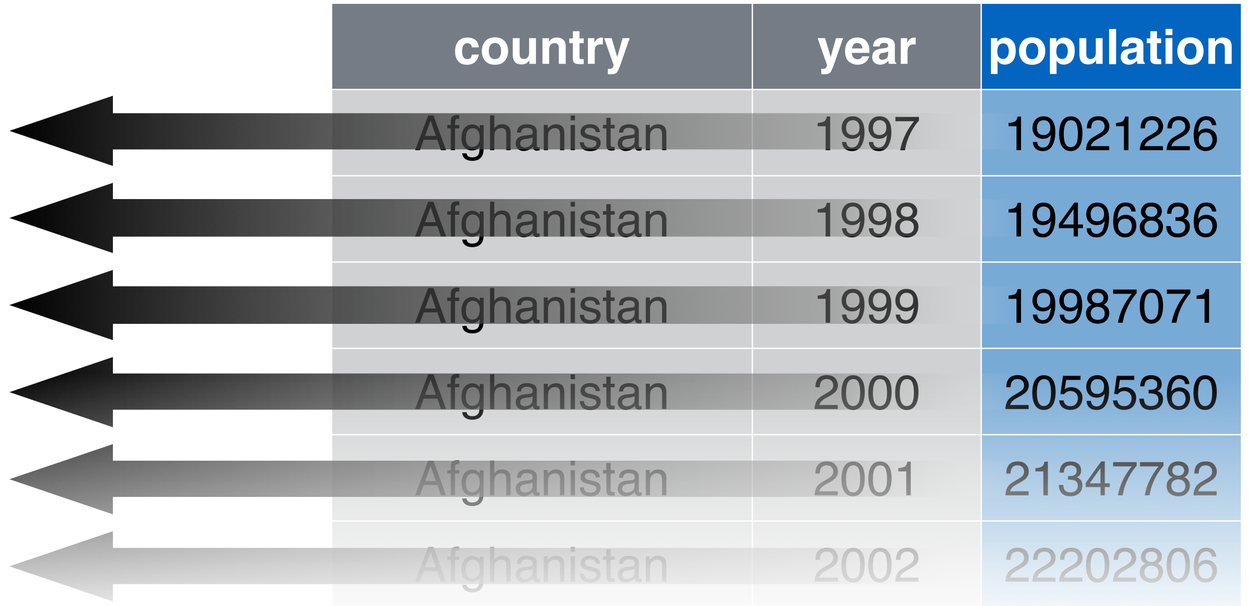


country	1995	1996	1997	1998	1999	2000	2001	2002	2003	20
Afghanistan	17586073	18415307	19021226	19496836	19987071	20595360	21347782	22202806	23116142	240
Algeria	29315463	29845208	30345	3082043	31276295	31719449	32150198	32572977	33003442	334
Angola	12104952	12451045	12791388	131375	13510616	1392	14385283	14886574	15421075	159
Argentina	34533168	35264070	3569077	36109342	36514558	3	37273361	37627545	37970411	383
Azerbaijan	7770806	7852273	79	7984460	8047936	8117742	8195427	8279768	8370169	846
Bangladesh	119869585	122400896	124945315	127478524	129966823	132383265	134729503	137006279	139185986	1412

Strategy

1. Tidy the population data set
2. Join the population values to the tb2 data set

country	year	cases	population
Afghanistan	1997	128	19021226
Afghanistan	1998	1778	19496836
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Afghanistan	2001	4639	21347782
Afghanistan	2002	6509	22202806



country	year	population
Afghanistan	1997	19021226
Afghanistan	1998	19496836
Afghanistan	1999	19987071
Afghanistan	2000	20595360
Afghanistan	2001	21347782
Afghanistan	2002	22202806

Strategy

1. Tidy the population data set
2. Join the population values to the tb2 data set
3. Use mutate() to calculate the rate from cases and population.

country	year	cases	population	rate
Afghanistan	1997	128	19021226	0.07
Afghanistan	1998	1778	19496836	0.91
Afghanistan	1999	745	19987071	0.37
Afghanistan	2000	2666	20595360	1.29
Afghanistan	2001	4639	21347782	2.17
Afghanistan	2002	6509	22202806	2.93
Afghanistan	2003	6528	23116142	2.82

Your Turn

1. Use tidyr functions to reshape population into a tidy data set with three columns: *country, year*, and *population.*
2. Combine tb2 with population.
3. Use dplyr functions to create a rate variable (cases / population * 10000).
4. Select just the country, year, and rate variables of tb2.

country	year	rate
Afghanistan	1997	0.07
Afghanistan	1998	0.91
Afghanistan	1999	0.87
Afghanistan	2000	1.29

Step 1 - Gather the year columns of population

```
population <- population %>%
  gather("year", "population", -1, convert = TRUE)
```

Step 2 - Join population to tb2

```
population <- population %>%
  gather("year", "population", -1, convert = TRUE)

tb2 %>%
  left_join(population, by = c("country", "year"))
```

Step 3 - Calculate rate variable

```
population <- population %>%
  gather("year", "population", -1, convert = TRUE)

tb2 %>%
  left_join(population, by = c("country", "year")) %>%
  mutate(rate = cases / population * 10000)
```

Step 4 - Select country, year, rate

```
population <- population %>%
  gather("year", "population", -1, convert = TRUE)

tb2 %>%
  left_join(population, by = c("country", "year")) %>%
  mutate(rate = cases / population * 10000) %>%
  select(country, year, rate)
```