Reshaping data

Warmup activity

Work on the activity (handout) with a neighbor, then we will discuss as a class

Warmup

```
# A tibble: 260 × 38
   country `1975` `1976` `1977` `1978` `1979` `1980` `1981` `1982`
`1983` `1984`
   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>  <dbl>
 1 Afghan…     NA     NA     NA     NA   4.99   NA       NA     NA
NA   NA
 2 Albania     NA     NA     NA     NA  NA      NA       NA     NA
NA   NA
 3 Algeria     NA     NA     NA     NA  NA      NA       NA     NA
NA   NA
 4 Andorra     NA     NA     NA     NA  NA      NA       NA     NA
NA   NA
```

Question: What does a row in this data represent?

Warmup

```
# A tibble: 260 × 38
   country `1975` `1976` `1977` `1978` `1979` `1980` `1981` `1982`
`1983` `1984`
   <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>  <dbl>
 1 Afghan…     NA     NA     NA     NA   4.99   NA       NA     NA
NA    NA
 2 Albania     NA     NA     NA     NA  NA       NA       NA     NA
NA    NA
 3 Algeria     NA     NA     NA     NA  NA       NA       NA     NA
NA    NA
 4 Andorra     NA     NA     NA     NA  NA       NA       NA     NA
NA    NA
```

Question: What does a row in this data represent?

Each row is one country

Warmup

```
# A tibble: 260 × 38
   country `1975` `1976` `1977` `1978` `1979` `1980` `1981` `1982`
`1983` `1984`
   <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>  <dbl>
 1 Afghan…     NA     NA     NA     NA   4.99   NA       NA     NA
NA   NA
 2 Albania     NA     NA     NA     NA  NA      NA       NA     NA
NA   NA
 3 Algeria     NA     NA     NA     NA  NA      NA       NA     NA
NA   NA
 4 Andorra     NA     NA     NA     NA  NA      NA       NA     NA
NA   NA
```

Question: Is this table in “wide” or “narrow” format?

Warmup

```
# A tibble: 260 × 38
   country `1975` `1976` `1977` `1978` `1979` `1980` `1981` `1982`
`1983` `1984`
   <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>  <dbl>
 1 Afghan…     NA     NA     NA     NA   4.99   NA       NA     NA
NA    NA
 2 Albania     NA     NA     NA     NA  NA      NA       NA     NA
NA    NA
 3 Algeria     NA     NA     NA     NA  NA      NA       NA     NA
NA    NA
 4 Andorra     NA     NA     NA     NA  NA      NA       NA     NA
NA    NA
```

Question: Is this table in “wide” or “narrow” format?

Wide format – there is a column for each value of a variable (year)

Warmup

```
# A tibble: 260 × 38
   country `1975` `1976` `1977` `1978` `1979` `1980` `1981` `1982`
`1983` `1984`
   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>  <dbl>
 1 Afghan…     NA     NA     NA     NA   4.99   NA         NA     NA
NA    NA
 2 Albania     NA     NA     NA     NA  NA      NA         NA     NA
NA    NA
 3 Algeria     NA     NA     NA     NA  NA      NA         NA     NA
NA    NA
 4 Andorra     NA     NA     NA     NA  NA      NA         NA     NA
NA    NA
```

Question: What would the data look like in *narrow* form?

country	year	literacy rate
Afghanistan	1975	NA
	1976	NA
		⋮

Warmup

Literacy data in narrow form:

year has its own column

```
# A tibble: 571 × 3
   country      year  literacy_rate
   <chr>        <chr>         <dbl>
 1 Afghanistan 1979           4.99
 2 Afghanistan 2011          13
 3 Albania     2001          98.3
 4 Albania     2008          94.7
 5 Albania     2011          95.7
 6 Algeria     1987          35.8
 7 Algeria     2002          60.1
 8 Algeria     2006          63.9
 9 Angola      2001          54.2
10 Angola      2011          58.6
```

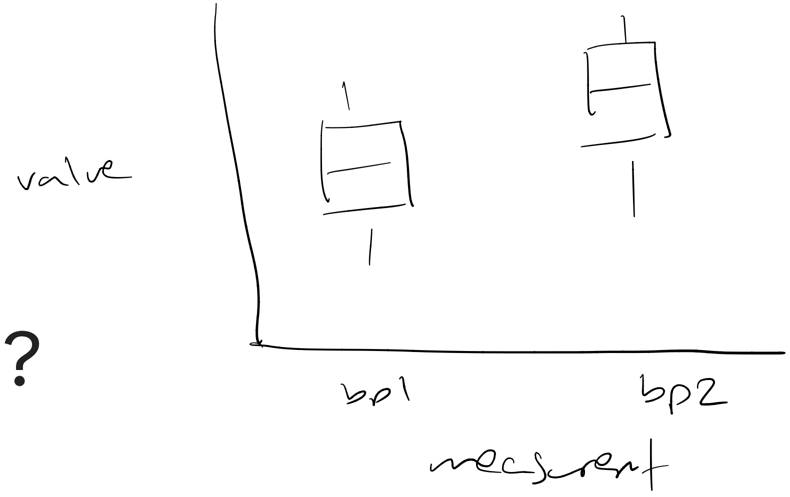
Now a row is a country - year combination

Another example

Data on three patients (A, B, C), with two blood pressure measurements (bp1 and bp2) per patient:

```
  id bp1 bp2
1  A 100 120
2  B 140 115
3  C 120 125
```

How might we want to reshape this data?



id	measurement	value
A	bp1	100
A	bp2	120
B	bp1	140
	⋮	
	etc.	

Another example

Original data:

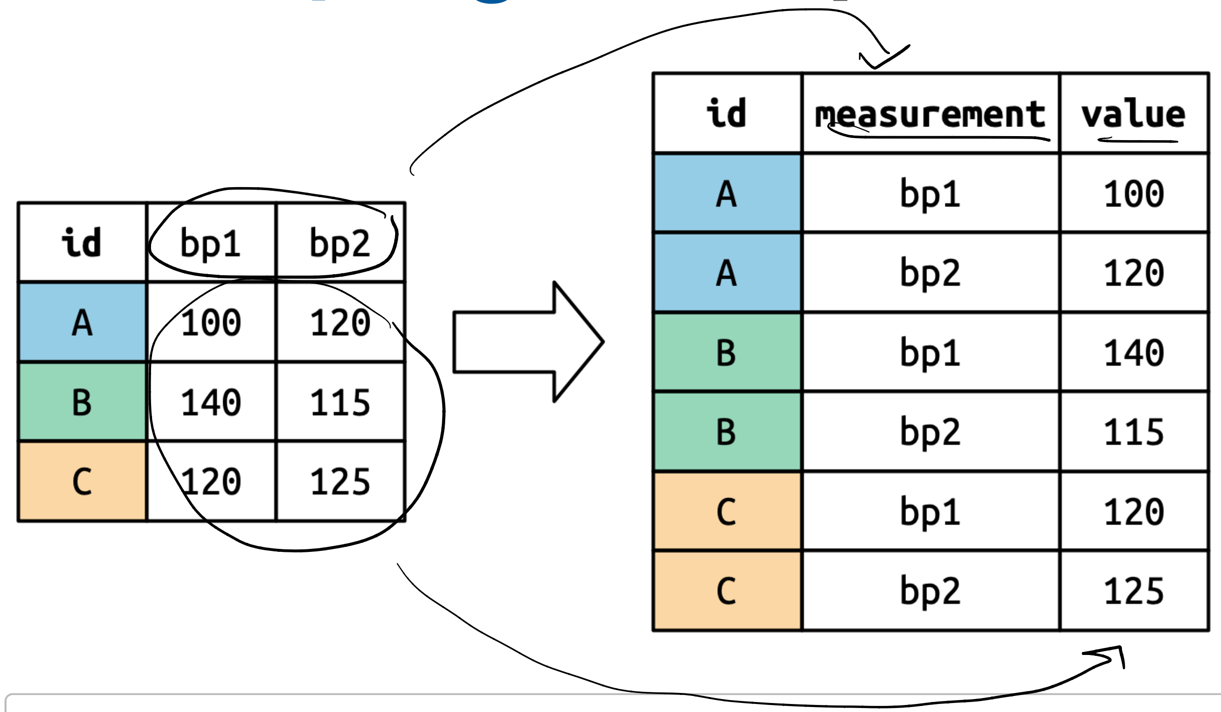
```
  id bp1 bp2
1  A 100 120
2  B 140 115
3  C 120 125
```

Reshaped data:

```
  id measurement value
1  A         bp1   100
2  A         bp2   120
3  B         bp1   140
4  B         bp2   115
5  C         bp1   120
6  C         bp2   125
```

Question: how do we do this reshaping in R?

Reshaping data: `pivot_longer`



id	bp1	bp2
A	100	120
B	140	115
C	120	125

id	measurement	value
A	bp1	100
A	bp2	120
B	bp1	140
B	bp2	115
C	bp1	120
C	bp2	125

```r
df |>
  pivot_longer(
    cols = bp1:bp2,
    names_to = "measurement",
    values_to = "value"
  )
```

<- columns to pivot

(take column names & make them entries in a new column)

(Image and example from *R for Data Science*)

pivot_longer

```
# A tibble: 260 × 38
   country `1975` `1976` `1977` `1978` `1979` `1980` `1981` `1982`
`1983` `1984`
   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>  <dbl>
 1 Afghan…      NA     NA     NA     NA   4.99     NA      NA     NA
NA    NA
 2 Albania      NA     NA     NA     NA NA         NA      NA     NA
NA    NA
 3 Algeria      NA     NA     NA     NA NA         NA      NA     NA
NA    NA
 4 Andorra      NA     NA     NA     NA NA         NA      NA     NA
NA    NA
```

```
litF |>
  pivot_longer(
    cols = ...,  ←   everything except country
    names_to = ...,  ←  year
    values_to = ...  ←   adult literacy rate
  )
```

pivot_longer

```
litF |>
  pivot_longer(
    cols = -country,
    names_to = "year",
    values_to = "literacy_rate"
  )
```

everything *except* country

```
# A tibble: 9,620 × 3
   country     year  literacy_rate
   <chr>       <chr>         <dbl>
 1 Afghanistan 1975          NA
 2 Afghanistan 1976          NA
 3 Afghanistan 1977          NA
 4 Afghanistan 1978          NA
 5 Afghanistan 1979           4.99
 6 Afghanistan 1980          NA
 7 Afghanistan 1981          NA
 8 Afghanistan 1982          NA
 9 Afghanistan 1983          NA
10 Afghanistan 1984          NA
```

pivot_longer

```
litF |>
  pivot_longer(
    cols = -country,
    names_to = "year",
    values_to = "literacy_rate"
  ) |>
  drop_na()          ← remove rows w/ NAs
```

```
# A tibble: 571 × 3
   country     year  literacy_rate
   <chr>       <chr>         <dbl>
 1 Afghanistan 1979           4.99
 2 Afghanistan 2011          13   
 3 Albania     2001          98.3 
 4 Albania     2008          94.7 
 5 Albania     2011          95.7 
 6 Algeria     1987          35.8 
 7 Algeria     2002          60.1 
 8 Algeria     2006          63.9 
 9 Angola      2001          54.2 
10 Angola      2011          58.6 
```

pivot_longer

```r
litF |>
  pivot_longer(
    cols = -country,
    names_to = "year",
    values_to = "literacy_rate",
    values_drop_na = T
  )
```

```
# A tibble: 571 × 3
   country     year  literacy_rate
   <chr>       <chr>         <dbl>
 1 Afghanistan 1979           4.99
 2 Afghanistan 2011          13
 3 Albania     2001          98.3
 4 Albania     2008          94.7
 5 Albania     2011          95.7
 6 Algeria     1987          35.8
 7 Algeria     2002          60.1
 8 Algeria     2006          63.9
 9 Angola      2001          54.2
10 Angola      2011          58.6
```

Example 2

Now consider the following table:

```
ex_df
```

```
  id x_1 x_2 y_1 y_2
1  1   3   5   0   2
2  2   1   8   1   7
3  3   4   9   2   9
```

What will the following code return?

```
ex_df |>
  pivot_longer(cols = -id,
               names_to = "group_obs",
               values_to = "value")
```

everything except id

id	group_obs	value
1	x_1	3
1	x_2	5
1	y_1	0

Example 2

Original data:

```r
ex_df
```

```
  id x_1 x_2 y_1 y_2
1  1   3   5   0   2
2  2   1   8   1   7
3  3   4   9   2   9
```

Reshaped data:

```r
ex_df |>
  pivot_longer(cols = -id,
               names_to = "group_obs",
               values_to = "value")
```

```
# A tibble: 12 × 3
      id group_obs value
   <dbl> <chr>     <dbl>
 1     1 x_1           3
 2     1 x_2           5
 3     1 y_1           0
 4     1 y_2           2
 5     2 x_1           1
 6     2 x_2           8
 7     2 y_1           1
 8     2 y_2           7
 9     3 x_1           4
10     3 x_2           9
```

Example 3

Consider the following example data:

```
  id bp_1 bp_2 hr_1 hr_2
1  1  100  120   60   77
2  2  120  115   75   81
3  3  125  130   80   93
```

What if we want the data to look like this:

```
# A tibble: 12 × 4
      id measurement stage value
   <dbl> <chr>       <chr> <dbl>
 1     1 bp          1       100
 2     1 bp          2       120
 3     1 hr          1        60
 4     1 hr          2        77
 5     2 bp          1       120
 6     2 bp          2       115
 7     2 hr          1        75
 8     2 hr          2        81
 9     3 bp          1       125
```

- pivot
- also need to separate names

bp_1 → bp 1

bp_2 → bp 2

Example 3

```r
df_3
```

```
  id bp_1 bp_2 hr_1 hr_2
1  1  100  120   60   77
2  2  120  115   75   81
3  3  125  130   80   93
```

```r
df_3 |>
  pivot_longer(cols = -id,
               names_to = c("measurement", "stage"),
               names_sep = "_",
               values_to = "value")
```

← separate names of original columns (by _)

```
# A tibble: 12 × 4
     id measurement stage value
  <dbl> <chr>       <chr> <dbl>
1     1 bp          1       100
2     1 bp          2       120
3     1 hr          1        60
4     1 hr          2        77
5     2 bp          1       120
6     2 bp          2       115
```

Example 3

```r
df_3 |>
  pivot_longer(cols = -id,
               names_to = c("measurement", "stage"),
               names_sep = "_",
               values_to = "value")
```

Step 1: Pivot

```
# A tibble: 6 × 3
     id measurement value
  <dbl> <chr>       <dbl>
1     1 bp_1          100
2     1 bp_2          120
3     1 hr_1           60
4     1 hr_2           77
5     2 bp_1          120
6     2 bp_2          115
```

Step 2: Separate columns

```
# A tibble: 6 × 4
     id measurement stage value
  <dbl> <chr>       <chr> <dbl>
1     1 bp          1       100
2     1 bp          2       120
3     1 hr          1        60
4     1 hr          2        77
5     2 bp          1       120
6     2 bp          2       115
```

Class activity

https://sta279-f25.github.io/class_activities/ca_04.html

- Work with a neighbor on the class activity
- At the end of class, submit your work as an HTML file on Canvas (one per group, list all your names)

For next time, read:

- Chapter 5 in *R for Data Science* (2nd ed.)