Introduction to R: **Reshaping**

Session 2, Part C

Nick Graetz[1]

[1] University of Pennsylvania, Population Studies Center

9/4/2020

In this lecture

1. Data shapes
2. Reshape long with `melt()`
3. Reshape wide with `dcast()`

Data shapes

Tabular data can be "shaped" in many different ways:

```
   fips year cig_sales_pc
1:    1 2013         64.6
2:    1 2014         61.7
3:    2 2013         39.0
4:    2 2014         37.2
5:    4 2013         24.4
6:    4 2014         23.0
```

Data shapes

Tabular data can be "shaped" in many different ways:

```
   fips year cig_sales_pc
1:    1 2013         64.6
2:    1 2014         61.7
3:    2 2013         39.0
4:    2 2014         37.2
5:    4 2013         24.4
6:    4 2014         23.0
```

```
   fips year_2013 year_2014
1:    1      64.6      61.7
2:    2      39.0      37.2
3:    4      24.4      23.0
```

Data shapes

Tabular data can be "shaped" in many different ways:

```
   fips year cig_sales_pc
1:    1 2013         64.6
2:    1 2014         61.7
3:    2 2013         39.0
4:    2 2014         37.2
5:    4 2013         24.4
6:    4 2014         23.0
```

```
   fips year_2013 year_2014
1:    1      64.6      61.7
2:    2      39.0      37.2
3:    4      24.4      23.0
```

```
   year state_1 state_2 state_4
1: 2013    64.6    39.0    24.4
2: 2014    61.7    37.2    23.0
```

Data shapes

Even though the data are the same, different shapes can be easier or harder to work with depending on the task at hand.

Changing the data shape is called "reshaping":

- Reshaping "wide" generally makes the data set *shorter* and *wider*
- Reshaping "long" generally makes the data set *longer* and *narrower*

Data shapes

Even though the data are the same, different shapes can be easier or harder to work with depending on the task at hand.

Changing the data shape is called "reshaping":

- Reshaping "wide" generally makes the data set *shorter* and *wider*
- Reshaping "long" generally makes the data set *longer* and *narrower*

There is a `reshape()` function in base R for reshaping both wide and long, however it's not very user-friendly.

Instead, we will use the `melt()` and `cast()` functions from the `data.table` library to reshape long and wide, respectively.

Reshaping long

The `melt()` function reshapes data long. Basically, it takes all of the columns you don't specify as ID variables and converts them into a single column.

```
> load(paste0(main_dir, "data/wa_data.rdata"))
> data <- as.data.table(data)
> data[, `:=`(pop, as.numeric(pop))]
> data[, `:=`(deaths, as.numeric(deaths))]
> dim(data)
[1] 24  5
> data
         cnty year sex     pop deaths
 1:      King 2010   1  965486   5770
 2:      King 2010   2  971999   5988
 3:      King 2011   1  983391   6012
 4:      King 2011   2  987922   6082
 5:      King 2012   1 1001169   6154
 6:      King 2012   2 1006405   6142
 7:      King 2013   1 1021389   6219
 8:      King 2013   2 1023060   6252
 9:    Pierce 2010   1  393265   2902
10:    Pierce 2010   2  402231   2785
11:    Pierce 2011   1  397685   2941
12:    Pierce 2011   2  405708   2812
13:    Pierce 2012   1  402480   3014
14:    Pierce 2012   2  409575   2672
15:    Pierce 2013   1  407307   3047
16:    Pierce 2013   2  412436   2972
17: Snohomish 2010   1  358067   2233
18: Snohomish 2010   2  357377   2237
19: Snohomish 2011   1  361939   2322
20: Snohomish 2011   2  360577   2353
21: Snohomish 2012   1  366949   2301
22: Snohomish 2012   2  366013   2375
23: Snohomish 2013   1  373991   2463
24: Snohomish 2013   2  371922   2409
         cnty year sex     pop deaths
```

RESHAPING LONG

```
> long <- melt(data, id.vars = c("cnty", "year", "sex"))
> dim(long)
[1] 48  5
> head(long, 30)
         cnty year sex variable   value
 1:      King 2010   1      pop  965486
 2:      King 2010   2      pop  971999
 3:      King 2011   1      pop  983391
 4:      King 2011   2      pop  987922
 5:      King 2012   1      pop 1001169
 6:      King 2012   2      pop 1006405
 7:      King 2013   1      pop 1021389
 8:      King 2013   2      pop 1023060
 9:    Pierce 2010   1      pop  393265
10:    Pierce 2010   2      pop  402231
11:    Pierce 2011   1      pop  397685
12:    Pierce 2011   2      pop  405708
13:    Pierce 2012   1      pop  402480
14:    Pierce 2012   2      pop  409575
15:    Pierce 2013   1      pop  407307
16:    Pierce 2013   2      pop  412436
17: Snohomish 2010   1      pop  358067
18: Snohomish 2010   2      pop  357377
19: Snohomish 2011   1      pop  361939
20: Snohomish 2011   2      pop  360577
21: Snohomish 2012   1      pop  366949
22: Snohomish 2012   2      pop  366013
23: Snohomish 2013   1      pop  373991
24: Snohomish 2013   2      pop  371922
25:      King 2010   1   deaths    5770
26:      King 2010   2   deaths    5988
27:      King 2011   1   deaths    6012
28:      King 2011   2   deaths    6082
29:      King 2012   1   deaths    6154
30:      King 2012   2   deaths    6142
         cnty year sex variable   value
```

Reshaping wide

The `dcast()` function reshapes data wide based on a formula you provide. Any variables listed to the left of the ~ remain columns, while variables listed to the right are used to split up the data into multiple new columns:

```
> wide <- dcast(long, cnty + year + variable ~ sex, value.var = "value
> head(wide, 15)
      cnty year variable       1       2
 1:   King 2010      pop  965486  971999
 2:   King 2010   deaths    5770    5988
 3:   King 2011      pop  983391  987922
 4:   King 2011   deaths    6012    6082
 5:   King 2012      pop 1001169 1006405
 6:   King 2012   deaths    6154    6142
 7:   King 2013      pop 1021389 1023060
 8:   King 2013   deaths    6219    6252
 9: Pierce 2010      pop  393265  402231
10: Pierce 2010   deaths    2902    2785
11: Pierce 2011      pop  397685  405708
12: Pierce 2011   deaths    2941    2812
13: Pierce 2012      pop  402480  409575
14: Pierce 2012   deaths    3014    2672
15: Pierce 2013      pop  407307  412436
```

Reshaping wide

There are often many different ways to reshape data wide:

```
> wide <- dcast(long, sex + year + variable ~ cnty, value.var = "value
> wide
    sex year variable    King Pierce Snohomish
 1:   1 2010      pop  965486 393265    358067
 2:   1 2010   deaths    5770   2902      2233
 3:   1 2011      pop  983391 397685    361939
 4:   1 2011   deaths    6012   2941      2322
 5:   1 2012      pop 1001169 402480    366949
 6:   1 2012   deaths    6154   3014      2301
 7:   1 2013      pop 1021389 407307    373991
 8:   1 2013   deaths    6219   3047      2463
 9:   2 2010      pop  971999 402231    357377
10:   2 2010   deaths    5988   2785      2237
11:   2 2011      pop  987922 405708    360577
12:   2 2011   deaths    6082   2812      2353
13:   2 2012      pop 1006405 409575    366013
14:   2 2012   deaths    6142   2672      2375
15:   2 2013      pop 1023060 412436    371922
16:   2 2013   deaths    6252   2972      2409
```

Reshaping wide

```
> wide <- dcast(long, cnty + sex + variable ~ year, value.var = "value
> wide
          cnty sex variable   2010   2011    2012    2013
 1:       King   1      pop 965486 983391 1001169 1021389
 2:       King   1   deaths   5770   6012    6154    6219
 3:       King   2      pop 971999 987922 1006405 1023060
 4:       King   2   deaths   5988   6082    6142    6252
 5:     Pierce   1      pop 393265 397685  402480  407307
 6:     Pierce   1   deaths   2902   2941    3014    3047
 7:     Pierce   2      pop 402231 405708  409575  412436
 8:     Pierce   2   deaths   2785   2812    2672    2972
 9: Snohomish   1      pop 358067 361939  366949  373991
10: Snohomish   1   deaths   2233   2322    2301    2463
11: Snohomish   2      pop 357377 360577  366013  371922
12: Snohomish   2   deaths   2237   2353    2375    2409
```

Reshaping wide

```
> wide <- dcast(long, cnty + year + sex ~ variable, value.var = "value
> head(wide, 20)
          cnty year sex     pop deaths
 1:       King 2010   1  965486   5770
 2:       King 2010   2  971999   5988
 3:       King 2011   1  983391   6012
 4:       King 2011   2  987922   6082
 5:       King 2012   1 1001169   6154
 6:       King 2012   2 1006405   6142
 7:       King 2013   1 1021389   6219
 8:       King 2013   2 1023060   6252
 9:     Pierce 2010   1  393265   2902
10:     Pierce 2010   2  402231   2785
11:     Pierce 2011   1  397685   2941
12:     Pierce 2011   2  405708   2812
13:     Pierce 2012   1  402480   3014
14:     Pierce 2012   2  409575   2672
15:     Pierce 2013   1  407307   3047
16:     Pierce 2013   2  412436   2972
17: Snohomish 2010   1  358067   2233
18: Snohomish 2010   2  357377   2237
19: Snohomish 2011   1  361939   2322
20: Snohomish 2011   2  360577   2353
```

Reshaping wide

And you can reshape wide by multiple variables at the same time:

```
> wide <- dcast(long, cnty + variable ~ sex + year, value.var = "value
> wide
        cnty variable 1_2010 1_2011  1_2012  1_2013 2_2010
1:      King      pop 965486 983391 1001169 1021389 971999
2:      King   deaths   5770   6012    6154    6219   5988
3:    Pierce      pop 393265 397685  402480  407307 402231
4:    Pierce   deaths   2902   2941    3014    3047   2785
5: Snohomish      pop 358067 361939  366949  373991 357377
6: Snohomish   deaths   2233   2322    2301    2463   2237
   2_2011  2_2012  2_2013
1: 987922 1006405 1023060
2:   6082    6142    6252
3: 405708  409575  412436
4:   2812    2672    2972
5: 360577  366013  371922
6:   2353    2375    2409
```

Reshaping wide

```
> wide <- dcast(long, variable ~ cnty + sex + year, value.var = "value
> wide
   variable King_1_2010 King_1_2011 King_1_2012 King_1_2013
1:      pop      965486      983391     1001169     1021389
2:   deaths        5770        6012        6154        6219
   King_2_2010 King_2_2011 King_2_2012 King_2_2013
1:      971999      987922     1006405     1023060
2:        5988        6082        6142        6252
   Pierce_1_2010 Pierce_1_2011 Pierce_1_2012 Pierce_1_2013
1:        393265        397685        402480        407307
2:          2902          2941          3014          3047
   Pierce_2_2010 Pierce_2_2011 Pierce_2_2012 Pierce_2_2013
1:        402231        405708        409575        412436
2:          2785          2812          2672          2972
   Snohomish_1_2010 Snohomish_1_2011 Snohomish_1_2012
1:           358067           361939           366949
2:             2233             2322             2301
   Snohomish_1_2013 Snohomish_2_2010 Snohomish_2_2011
1:           373991           357377           360577
2:             2463             2237             2353
   Snohomish_2_2012 Snohomish_2_2013
1:           366013           371922
2:             2375             2409
```

Reshaping

In practice, the usual approach to reshaping is to first melt your data into a totally long format (i.e., just one column for actual data) and then cast it wide to the final desired format:

```
> long <- melt(data, id.vars = c("cnty", "year", "sex"))
> wide <- dcast(long, year + sex ~ variable + cnty, value.var = "value
```

```
> head(data, 3)
   cnty year sex    pop deaths
1: King 2010   1 965486   5770
2: King 2010   2 971999   5988
3: King 2011   1 983391   6012
>
> head(long, 3)
   cnty year sex variable  value
1: King 2010   1      pop 965486
2: King 2010   2      pop 971999
3: King 2011   1      pop 983391
>
> head(wide, 3)
   year sex pop_King pop_Pierce pop_Snohomish deaths_King
1: 2010   1   965486     393265        358067        5770
2: 2010   2   971999     402231        357377        5988
3: 2011   1   983391     397685        361939        6012
   deaths_Pierce deaths_Snohomish
1:          2902             2233
2:          2785             2237
3:          2941             2322
```