

# Melt and Cast The Shape of Your Data-Frame: Exercises



Data-sets often arrive to us in a form that is different from what we need for our modeling or visualization functions, which, in turn, don't necessarily require the same format.

Reshaping data.frames is a step that all analysts need, but many struggle with. Practicing this meta-skill will, in the long-run, result in more time to focus on the actual analysis.

The solutions to this set will rely on `data.table`, mostly `melt()` and `dcast()`, which are originally from the `reshape2` package. However, you can also get practice out of it using your favorite base-R, tidy-verse or any other method, then compare the results.

Solutions are available [here](#).

## Exercise 1

Take the following data.frame *from* this form:

```
df <- data.frame(id = 1:2, q1 = c("A", "B"), q2 = c("C", "A"),
```

```
stringsAsFactors = FALSE)
df
  id q1 q2
1  1  A  C
2  2  B  A
```

to this:

```
  id question value
1  1         q1    A
2  2         q1    B
3  1         q2    C
4  2         q2    A
```

## Exercise 2

Do the opposite; return the data.frame back to its original form.

## Exercise 3

Set up the data.frame in terms of questions, as follows:

```
  question id_1 id_2
1         q1    A    B
2         q2    C    A
```

## Exercise 4

The data entry behind this data.frame went a little bit wrong. Get all the C and B entries into their corresponding columns:

```
df2 <- data.frame(
  A = c("A1", "A12", "A31", "A4"),
  B = c("B4", "C7", "C3", "B9"),
  C = c("C3", "B16", "B3", "C4")
)
```

## Exercise 5

Get this data.frame:

```
df3 <- data.frame(
  Join_ID = rep(1:3, each = 2),
  Type    = rep(c("a", "b"), 3),
  v2      = c(8, 9, 7, 6, 5, 4)*10
)
```

To look like this:

	Join_ID	a_v2	b_v2
1	1	80	90
2	2	70	60
3	3	50	40

## Exercise 6

Revisiting a data-set used in an earlier [exercise set on data exploration](#),

load the AER package and run the command `data("Fertility")`, which loads the data-set `Fertility` to your work space.

Melt it into the following format, with one row per child.

```
head(ferl)
  morekids age afam hispanic other work mother_id order gender
1      no  27  no      no      no    0         1     1  male
2      no  30  no      no      no   30         2     1 female
3      no  27  no      no      no    0         3     1  male
4      no  35 yes      no      no    0         4     1  male
5      no  30  no      no      no   22         5     1 female
6      no  26  no      no      no   40         6     1  male
```

## Exercise 7

Take this:

```
d1 = data.frame(
  ID=c(1,1,1,2,2,4,1,2),
  medication=c(1,2,3,1,2,7,2,8)
)
```

```
d1
  ID medication
1  1           1
2  1           2
3  1           3
4  2           1
5  2           2
6  4           7
7  1           2
8  2           8
```

to this form:

```
  ID medications
1:  1  1, 2, 3, 2
2:  2    1, 2, 8
3:  4           7
```

Note: the solution doesn't use `melt()` nor `dcast()`, so you might look at other options.

## Exercise 8

Get this:

```
dfs <- data.frame(
  Name = c(rep("name1",3),rep("name2",2)),
  MedName = c("atenolol 25mg","aspirin 81mg","sildenafil
100mg", "atenolol 50mg","enalapril 20mg")
)
```

```
dfs
  Name          MedName
1 name1  atenolol 25mg
```

```
2 name1      aspirin 81mg
3 name1 sildenafil 100mg
4 name2      atenolol 50mg
5 name2      enalapril 20mg
```

Into the following format:

```
      Name medication_1 medication_2 medication_3
1: name1 atenolol 25mg aspirin 81mg sildenafil 100mg
2: name2 atenolol 50mg enalapril 20mg
```

### Exercise 9

Get the following data.frame organized in standard form:

```
df7 <- data.table(
  v1 = c("name1, name2", "name3", "name4, name5"),
  v2 = c("1, 2", "3", "4, 5"),
  v3 = c(1, 2, 3)
)
df7
```

```
      v1      v2 v3
1: name1, name2 1, 2 1
2:      name3   3 2
3: name4, name5 4, 5 3
```

Expected output:

```
      v1 v2 v3
1: name1 1 1
2: name2 2 1
3: name3 3 2
4: name4 4 3
5: name5 5 3
```

The solution doesn't use melt() nor dcast() and can be suprisingly hard.

### Exercise 10

Convert this:

```
df <- data.frame(
  Method = c("10.fold.CV Lasso", "10.fold.CV.1SE", "BIC",
"Modified.BIC"),
  n = c(30, 30, 50, 50, 50, 50, 100, 100),
  lambda = c(1, 3, 1, 2, 2, 0, 1, 2),
  df = c(21, 17, 29, 26, 25, 32, 34, 32) )
> df
```

	Method	n	lambda	df
1	10.fold.CV Lasso	30	1	21
2	10.fold.CV.1SE	30	3	17
3	BIC	50	1	29
4	Modified.BIC	50	2	26
5	10.fold.CV Lasso	50	2	25
6	10.fold.CV.1SE	50	0	32
7	BIC	100	1	34
8	Modified.BIC	100	2	32

Into:

	Method	lambda_30	lambda_50	lambda_100	df_30	df_50
df_100						
1	10.fold.CV Lasso	1	2		21	25
2	10.fold.CV.1SE	3	0		17	32
3	BIC		1	1		29
34						
4	Modified.BIC		2	2		26
32						

(Image by [Joe Alterio](#))