# 数据拆分和合并

# 本节内容

- data.table的函数melt 和 dcast 是增强包reshape2里同名函数的扩展。
- 在这一讲，我们会：
  - 复习一下原先的函数 melt 和 dcast，它们是如何reshaping一个data.table。
  - 然后，了解一下当前的功能是如何变得冗长而且低效。
  - 最后，学习一下改进之后的函数 melt 和 dcast 如何同时处理多个列。

# 原生的melt / dcast

# 数据集

```
setwd("C:\\Users\\lenovo\\Documents\\软件学院\\大数据班\\R语言基础课件") #改变工作目录到csv文件所在目录
DT = fread("melt_default.csv")
#DT = fread("melt_enhanced.csv") #稍后使用
str(DT)
```

# 函数melt（1/2）

- DT

| | family_id | age_mother | dob_child1 | dob_child2 | dob_child3 |
|---|---|---|---|---|---|
| 1: | 1 | 30 | 1998-11-26 | 2000-01-29 | NA |
| 2: | 2 | 27 | 1996-06-22 | NA | NA |
| 3: | 3 | 26 | 2002-07-11 | 2004-04-05 | 2007-09-02 |
| 4: | 4 | 32 | 2004-10-10 | 2009-08-27 | 2012-07-21 |
| 5: | 5 | 29 | 2000-12-05 | 2005-02-28 | NA |

- DT.m1 = **melt**(DT, **id.vars =** c("family_id", "age_mother"), **measure.vars =** c("dob_child1", "dob_child2", "dob_child3"))

| | family_id | age_mother | variable | value |
|---|---|---|---|---|
| 1: | 1 | 30 | dob_child1 | 1998-11-26 |
| 2: | 2 | 27 | dob_child1 | 1996-06-22 |
| 3: | 3 | 26 | dob_child1 | 2002-07-11 |
| 4: | 4 | 32 | dob_child1 | 2004-10-10 |
| 5: | 5 | 29 | dob_child1 | 2000-12-05 |
| 6: | 1 | 30 | dob_child2 | 2000-01-29 |
| 7: | 2 | 27 | dob_child2 | NA |
| 8: | 3 | 26 | dob_child2 | 2004-04-05 |
| 9: | 4 | 32 | dob_child2 | 2009-08-27 |
| 10: | 5 | 29 | dob_child2 | 2005-02-28 |
| 11: | 1 | 30 | dob_child3 | NA |
| 12: | 2 | 27 | dob_child3 | NA |
| 13: | 3 | 26 | dob_child3 | 2007-09-02 |
| 14: | 4 | 32 | dob_child3 | 2012-07-21 |
| 15: | 5 | 29 | dob_child3 | NA |

简写成：

DT.m1 = melt(DT, id.vars = c("family_id", "age_mother"))

或

DT.m1 = melt(DT, measure.vars = c("dob_child1", "dob_child2", "dob_child3"))

也可以

# 函数melt（2/2）

- 分别将 variable列和 value列重命名为 child 和 dob

```
DT.m1 = melt(DT, measure.vars =
c("dob_child1", "dob_child2", "dob_child3"),
variable.name = "child", value.name = "dob")
```

| | family_id | age_mother | child | dob |
|---|---|---|---|---|
| 1: | 1 | 30 | dob_child1 | 1998-11-26 |
| 2: | 2 | 27 | dob_child1 | 1996-06-22 |
| 3: | 3 | 26 | dob_child1 | 2002-07-11 |
| 4: | 4 | 32 | dob_child1 | 2004-10-10 |
| 5: | 5 | 29 | dob_child1 | 2000-12-05 |
| 6: | 1 | 30 | dob_child2 | 2000-01-29 |
| 7: | 2 | 27 | dob_child2 | NA |
| 8: | 3 | 26 | dob_child2 | 2004-04-05 |
| 9: | 4 | 32 | dob_child2 | 2009-08-27 |
| 10: | 5 | 29 | dob_child2 | 2005-02-28 |
| 11: | 1 | 30 | dob_child3 | NA |
| 12: | 2 | 27 | dob_child3 | NA |
| 13: | 3 | 26 | dob_child3 | 2007-09-02 |
| 14: | 4 | 32 | dob_child3 | 2012-07-21 |
| 15: | 5 | 29 | dob_child3 | NA |

# 函数cast

- 将刚刚分拆的 DT.m1 还原成 DT

```
dcast(DT.m1, family_id + age_mother ~ child,
value.var = "dob")
```

```
   family_id age_mother dob_child1 dob_child2 dob_child3
1:         1         30 1998-11-26 2000-01-29         NA
2:         2         27 1996-06-22         NA         NA
3:         3         26 2002-07-11 2004-04-05 2007-09-02
4:         4         32 2004-10-10 2009-08-27 2012-07-21
5:         5         29 2000-12-05 2005-02-28         NA
```

# 如何对两组不同类型的属性分别melt成两列呢？

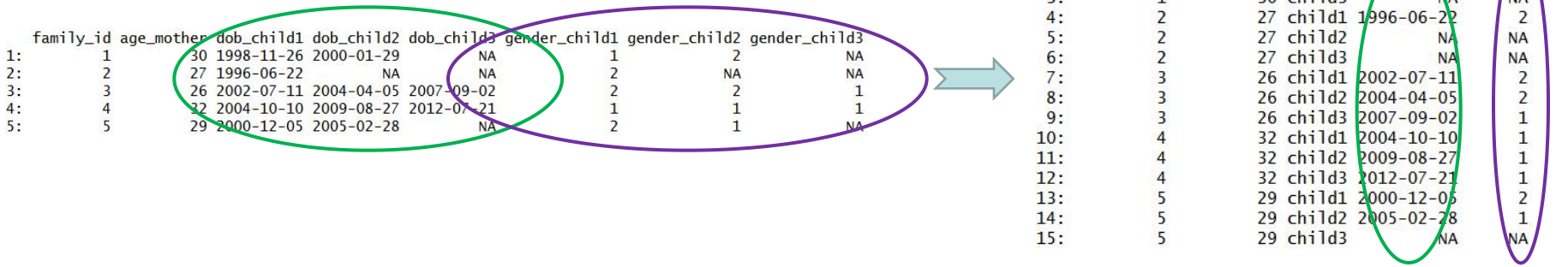
- 例：
  - 数据：melt_enhanced.csv #见左下图
  - 问题：希望将三列"dob_child1", "dob_child2", "dob_child3"，melt成一列dob；三列gender_child1，gender_child1，gender_child1，melt成一列gender。

| | family_id | age_mother | dob_child1 | dob_child2 | dob_child3 | gender_child1 | gender_child2 | gender_child3 |
|---|---|---|---|---|---|---|---|---|
| 1: | 1 | 30 | 1998-11-26 | 2000-01-29 | NA | 1 | 2 | NA |
| 2: | 2 | 27 | 1996-06-22 | NA | NA | 2 | NA | NA |
| 3: | 3 | 26 | 2002-07-11 | 2004-04-05 | 2007-09-02 | 2 | 2 | 1 |
| 4: | 4 | 32 | 2004-10-10 | 2009-08-27 | 2012-07-21 | 1 | 1 | 1 |
| 5: | 5 | 29 | 2000-12-05 | 2005-02-28 | NA | 2 | 1 | NA |

| | family_id | age_mother | child | dob | gender |
|---|---|---|---|---|---|
| 1: | 1 | 30 | child1 | 1998-11-26 | 1 |
| 2: | 1 | 30 | child2 | 2000-01-29 | 2 |
| 3: | 1 | 30 | child3 | NA | NA |
| 4: | 2 | 27 | child1 | 1996-06-22 | 2 |
| 5: | 2 | 27 | child2 | NA | NA |
| 6: | 2 | 27 | child3 | NA | NA |
| 7: | 3 | 26 | child1 | 2002-07-11 | 2 |
| 8: | 3 | 26 | child2 | 2004-04-05 | 2 |
| 9: | 3 | 26 | child3 | 2007-09-02 | 1 |
| 10: | 4 | 32 | child1 | 2004-10-10 | 1 |
| 11: | 4 | 32 | child2 | 2009-08-27 | 1 |
| 12: | 4 | 32 | child3 | 2012-07-21 | 1 |
| 13: | 5 | 29 | child1 | 2000-12-05 | 2 |
| 14: | 5 | 29 | child2 | 2005-02-28 | 1 |
| 15: | 5 | 29 | child3 | NA | NA |

# 用目前为止学过的知识解答

```
DT = fread("melt_enhanced.csv")
DT.m1 = melt(DT, id = c("family_id", "age_mother"))
DT.m1[, c("variable", "child") := tstrsplit(variable, "_", fixed = TRUE)]
DT.c1 = dcast(DT.m1, family_id + age_mother + child ~ variable, value.var = "value")
DT.c1
```



| | family_id | age_mother | dob_child1 | dob_child2 | dob_child3 | gender_child1 | gender_child2 | gender_child3 |
|---|---|---|---|---|---|---|---|---|
| 1: | 1 | 30 | 1998-11-26 | 2000-01-29 | NA | 1 | 2 | NA |
| 2: | 2 | 27 | 1996-06-22 | NA | NA | 2 | NA | NA |
| 3: | 3 | 26 | 2002-07-11 | 2004-04-05 | 2007-09-02 | 2 | 2 | 1 |
| 4: | 4 | 32 | 2004-10-10 | 2009-08-27 | 2012-07-21 | 1 | 1 | 1 |
| 5: | 5 | 29 | 2000-12-05 | 2005-02-28 | NA | 2 | 1 | NA |

| | family_id | age_mother | child | dob | gender |
|---|---|---|---|---|---|
| 1: | 1 | 30 | child1 | 1998-11-26 | 1 |
| 2: | 1 | 30 | child2 | 2000-01-29 | 2 |
| 3: | 1 | 30 | child3 | NA | NA |
| 4: | 2 | 27 | child1 | 1996-06-22 | 2 |
| 5: | 2 | 27 | child2 | NA | NA |
| 6: | 2 | 27 | child3 | NA | NA |
| 7: | 3 | 26 | child1 | 2002-07-11 | 2 |
| 8: | 3 | 26 | child2 | 2004-04-05 | 2 |
| 9: | 3 | 26 | child3 | 2007-09-02 | 1 |
| 10: | 4 | 32 | child1 | 2004-10-10 | 1 |
| 11: | 4 | 32 | child2 | 2009-08-27 | 1 |
| 12: | 4 | 32 | child3 | 2012-07-21 | 1 |
| 13: | 5 | 29 | child1 | 2000-12-05 | 2 |
| 14: | 5 | 29 | child2 | 2005-02-28 | 1 |
| 15: | 5 | 29 | child3 | NA | NA |

# 原生的melt / dcast的局限

- 先把所有的东西都拆分开了，再将它们合并。很容易看出，这太过迂回和低效了。
- 需要被整合的列可能是不同的类型，使用函数melt 的时候，这些列被硬塞到结果里面。
- 整合数据时引入很多操作。特别是，必须要计算等式中变量的顺序，代价太大。

# 增强的新功能

# 增强的melt（1/2）

- 上一个例子的解答：用data.table的melt 同时拆分多个列。
- 我们给参数 measure.vars **传递一个列表，这个列表的每个元素包含需要被合并的列**。

```
#DT = fread("melt_enhanced.csv")
colA = paste("dob_child", 1:3, sep = "")
colB = paste("gender_child", 1:3, sep = "")
DT.m2 = melt(DT, measure = list(colA, colB), value.name = c("dob", "gender"))
DT.m2
```

| | family_id | age_mother | dob_child1 | dob_child2 | dob_child3 | gender_child1 | gender_child2 | gender_child3 |
|---|---|---|---|---|---|---|---|---|
| 1: | 1 | 30 | 1998-11-26 | 2000-01-29 | NA | 1 | 2 | NA |
| 2: | 2 | 27 | 1996-06-22 | NA | NA | 2 | NA | NA |
| 3: | 3 | 26 | 2002-07-11 | 2004-04-05 | 2007-09-02 | 2 | 2 | 1 |
| 4: | 4 | 32 | 2004-10-10 | 2009-08-27 | 2012-07-21 | 1 | 1 | 1 |
| 5: | 5 | 29 | 2000-12-05 | 2005-02-28 | NA | 2 | 1 | NA |

| | family_id | age_mother | variable | dob | gender |
|---|---|---|---|---|---|
| 1: | 1 | 30 | 1 | 1998-11-26 | 1 |
| 2: | 2 | 27 | 1 | 1996-06-22 | 2 |
| 3: | 3 | 26 | 1 | 2002-07-11 | 2 |
| 4: | 4 | 32 | 1 | 2004-10-10 | 1 |
| 5: | 5 | 29 | 1 | 2000-12-05 | 2 |
| 6: | 1 | 30 | 2 | 2000-01-29 | 2 |
| 7: | 2 | 27 | 2 | NA | NA |
| 8: | 3 | 26 | 2 | 2004-04-05 | 2 |
| 9: | 4 | 32 | 2 | 2009-08-27 | 1 |
| 10: | 5 | 29 | 2 | 2005-02-28 | 1 |
| 11: | 1 | 30 | 3 | NA | NA |
| 12: | 2 | 27 | 3 | NA | NA |
| 13: | 3 | 26 | 3 | 2007-09-02 | 1 |
| 14: | 4 | 32 | 3 | 2012-07-21 | 1 |
| 15: | 5 | 29 | 3 | NA | NA |

# 增强的melt（2/2）

- 函数 patterns()

  通常，我们想整合的这些列的列名都有共通的格式。我们可以用函数patterns()指定正则表达式，让语法更简洁。上页的操作还可以这样写：

  ```
  DT.m2 = melt(DT, measure = patterns("^dob", "^gender"), value.name = c("dob", "gender"))
  DT.m2
  ```

- 这个功能是用C实现的，因此效率高，节省内存，而且简洁。

# 增强的dcast

- 同时合并多个 value.vars
  - 我们可以对函数dcast()**指定多个 value.var参数**，这样操作就在内部进行，而且高效。

```
DT.c2 = dcast(DT.m2, family_id + age_mother ~ variable, value.var = c("dob", "gender"))
DT.c2
```

```
    family_id age_mother variable        dob gender
 1:         1         30        1 1998-11-26      1
 2:         2         27        1 1996-06-22      2
 3:         3         26        1 2002-07-11      2
 4:         4         32        1 2004-10-10      1
 5:         5         29        1 2000-12-05      2
 6:         1         30        2 2000-01-29      2
 7:         2         27        2         NA     NA
 8:         3         26        2 2004-04-05      2
 9:         4         32        2 2009-08-27      1
10:         5         29        2 2005-02-28      1
11:         1         30        3         NA     NA
12:         2         27        3         NA     NA
13:         3         26        3 2007-09-02      1
14:         4         32        3 2012-07-21      1
15:         5         29        3         NA     NA
```

```
    family_id age_mother      dob_1      dob_2      dob_3 gender_1 gender_2 gender_3
 1:         1         30 1998-11-26 2000-01-29         NA        1        2       NA
 2:         2         27 1996-06-22         NA         NA        2       NA       NA
 3:         3         26 2002-07-11 2004-04-05 2007-09-02        2        2        1
 4:         4         32 2004-10-10 2009-08-27 2012-07-21        1        1        1
 5:         5         29 2000-12-05 2005-02-28         NA        2        1       NA
```