Chapter 24: Pivot and unpivot with data.table

Parameter	Details
id.vars	tell `melt` which columns to retain
variable.name	tell `melt` what to call the column with category labels
value.name	tell `melt` what to call the column that has values associated with category labels
value.var	tell `dcast` where to find the values to cast in columns
formula	tell `dcast` which columns to retain to form a unique record identifier (LHS) and which one holds the category labels (RHS)
fun.aggregate	specify the function to use when the casting operation generates a list of values in each cell

Section 24.1: Pivot and unpivot tabular data with data.table - I

Convert from wide form to long form

Load `data USArrests` from `datasets`.

```
data("USArrests")
head(USArrests)

           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
```

Use `?USArrests` to find out more. First, convert to `data.table`. The names of states are row names in the original `data.frame`.

```
library(data.table)
DT <- as.data.table(USArrests, keep.rownames=TRUE)
```

This is data in the wide form. It has a column for each variable. The data can also be stored in long form without loss of information. The long form has one column that stores the variable names. Then, it has another column for the variable values. The long form of `USArrests` looks like so.

```
             State    Crime  Rate
  1:       Alabama   Murder  13.2
  2:        Alaska   Murder  10.0
  3:       Arizona   Murder   8.1
  4:      Arkansas   Murder   8.8
  5:    California   Murder   9.0
 ---
196:      Virginia     Rape  20.7
197:    Washington     Rape  26.2
198: West Virginia     Rape   9.3
199:     Wisconsin     Rape  10.8
200:       Wyoming     Rape  15.6
```

We use the `melt` function to switch from wide form to long form.

```
DTm <- melt(DT)
names(DTm) <- c("State", "Crime", "Rate")
```

By default, `melt` treats all columns with numeric data as variables with values. In `USArrests`, the variable `UrbanPop` represents the percentage urban population of a state. It is different from the other variables, `Murder`, `Assault` and `Rape`, which are violent crimes reported per 100,000 people. Suppose we want to retain `UrbanPop` column. We achieve this by setting `id.vars` as follows.

```
DTmu <- melt(DT, id.vars=c("rn", "UrbanPop" ),
             variable.name='Crime', value.name = "Rate")
names(DTmu)[1] <- "State"
```

Note that we have specified the names of the column containing category names (Murder, Assault, etc.) with `variable.name` and the column containing the values with `value.name`. Our data looks like so.

```
             State UrbanPop  Crime Rate
  1:       Alabama       58 Murder 13.2
  2:        Alaska       48 Murder 10.0
  3:       Arizona       80 Murder  8.1
  4:      Arkansas       50 Murder  8.8
  5:    California       91 Murder  9.0
```

Generating summaries with with split-apply-combine style approach is a breeze. For example, to summarize violent crimes by state?

```
DTmu[, .(ViolentCrime = sum(Rate)), by=State]
```

This gives:

```
        State ViolentCrime
1:    Alabama        270.4
2:     Alaska        317.5
3:    Arizona        333.1
4:   Arkansas        218.3
5: California        325.6
6:   Colorado        250.6
```

Section 24.2: Pivot and unpivot tabular data with data.table - II

Convert from long form to wide form

To recover data from the previous example, use `dcast` like so.

```
DTc <- dcast(DTmu, State + UrbanPop ~ Crime)
```

This gives the data in the original wide form.

```
             State UrbanPop Murder Assault Rate
 1:        Alabama       58   13.2     236 21.2
 2:         Alaska       48   10.0     263 44.5
 3:        Arizona       80    8.1     294 31.0
 4:       Arkansas       50    8.8     190 19.5
 5:     California       91    9.0     276 40.6
```

Here, the formula notation is used to specify the columns that form a unique record identifier (LHS) and the column containing category labels for new column names (RHS). Which column to use for the numeric values? By default, `dcast` uses the first column with numerical values left over when from the formula specification. To make explicit, use the parameter `value.var` with column name.

When the operation produces a list of values in each cell, `dcast` provides a `fun.aggregate` method to handle the situation. Say I am interested in states with similar urban population when investigating crime rates. I add a column `Decile` with computed information.

```
DTmu[, Decile := cut(UrbanPop, quantile(UrbanPop, probs = seq(0, 1, by=0.1)))]
levels(DTmu$Decile) <- paste0(1:10, "D")
```

Now, casting `Decile ~ Crime` produces multiple values per cell. I can use `fun.aggregate` to determine how these are handled. Both text and numerical values can be handle this way.

```
dcast(DTmu, Decile ~ Crime, value.var="Rate", fun.aggregate=sum)
```

This gives:

```
dcast(DTmu, Decile ~ Crime, value.var="Rate", fun.aggregate=mean)
```

This gives:

```
            State UrbanPop  Crime Rate Decile
 1:       Alabama       58 Murder 13.2     4D
 2:        Alaska       48 Murder 10.0     2D
 3:       Arizona       80 Murder  8.1     8D
 4:      Arkansas       50 Murder  8.8     2D
 5:    California       91 Murder  9.0    10D
```

There are multiple states in each decile of the urban population. Use `fun.aggregate` to specify how these should be handled.

```
dcast(DTmu, Decile ~ Crime, value.var="Rate", fun.aggregate=sum)
```

This sums over the data for like states, giving the following.

```
   Decile Murder Assault  Rape
1:     1D   39.4     808  62.6
2:     2D   35.3     815  94.3
3:     3D   22.6     451  67.7
4:     4D   54.9     898 106.0
5:     5D   42.4     758 107.6
```