Reshaping Data

Contents

1 Reshaping Data

1.1 The goal is tidy data

1. Each variable forms a column
2. Each observation forms a row
3. Each table/file stores data about one kind of observation (e.g. people/hospitals).

1.2 Start with reshaping

```
library(reshape2)
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

1.3 Melting data frames

```
mtcars$carname <- rownames(mtcars)
carMelt <- melt(mtcars, id=c("carname","gear","cyl"), measure.vars=c("mpg","hp"))
head(carMelt)
```

carname	gear	cyl	variable	value
Mazda RX4	4	6	mpg	21.0
Mazda RX4 Wag	4	6	mpg	21.0

carname	gear	cyl	variable	value
Datsun 710	4	4	mpg	22.8
Hornet 4 Drive	3	6	mpg	21.4
Hornet Sportabout	3	8	mpg	18.7
Valiant	3	6	mpg	18.1

```
tail(carMelt)
```

	carname	gear	cyl	variable	value
59	Porsche 914-2	5	4	hp	91
60	Lotus Europa	5	4	hp	113
61	Ford Pantera L	5	8	hp	264
62	Ferrari Dino	5	6	hp	175
63	Maserati Bora	5	8	hp	335
64	Volvo 142E	4	4	hp	109

1.4 Casting data frames

```
cylData <- dcast(carMelt, cyl ~ variable)
```

```
## Aggregation function missing: defaulting to length
```

```
cylData
```

cyl	mpg	hp
4	11	11
6	7	7
8	14	14

```
cylData <- dcast(carMelt, cyl ~ variable, mean)
cylData
```

cyl	mpg	hp
4	26.66364	82.63636
6	19.74286	122.28571
8	15.10000	209.21429

The dcast() function in R is used to reshape data from long to wide format. It is a part of the 'reshape2' or 'data.table' package. Here is what it does in your specific context:

In this line, you're applying the dcast() function to the carMelt data frame. The argument cyl ~ variable indicates that you want to reshape your data so that you have one row for each unique value of cyl (cylinder), and one column for each unique value of variable.

The function that follows (mean) is applied to all cells in the data frame that correspond to a given (cyl, variable) pair. So, for instance, if your carMelt data frame has several rows for cyl = 4 and variable = 'mpg', then the dcast() function will take the mean of all these rows and put this value in the cell that corresponds to (cyl = 4, variable = 'mpg') in the new data frame.

1.5 Averaging values

```
head(InsectSprays)
```

count	spray
10	A
7	A
20	A
14	A
14	A
12	A

```
tapply(InsectSprays$count, InsectSprays$spray, sum)
```

```
##   A   B   C   D   E   F
## 174 184  25  59  42 200
```

1.6 Another way - split

```
spins <- split(InsectSprays$count, InsectSprays$spray)
spins
```

```
## $A
##  [1] 10  7 20 14 14 12 10 23 17 20 14 13
##
## $B
##  [1] 11 17 21 11 16 14 17 17 19 21  7 13
##
## $C
##  [1] 0 1 7 2 3 1 2 1 3 0 1 4
##
## $D
##  [1]  3  5 12  6  4  3  5  5  5  5  2  4
##
## $E
##  [1] 3 5 3 5 3 6 1 1 3 2 6 4
##
## $F
##  [1] 11  9 15 22 15 16 13 10 26 26 24 13
```

1.7 Another way - apply

```
sprCount = lapply(spins,sum)
sprCount
```

```
## $A
## [1] 174
##
## $B
## [1] 184
##
## $C
```

```
## [1] 25
##
## $D
## [1] 59
##
## $E
## [1] 42
##
## $F
## [1] 200
```

1.8 Another way - combine

```
unlist(sprCount)
```

```
##   A   B   C   D   E   F
## 174 184  25  59  42 200
```

```
sapply(spins, sum)
```

```
##   A   B   C   D   E   F
## 174 184  25  59  42 200
```

1.9 Another way - plyr package

```
ddply(InsectSprays,.(spray),summarize,sum=ave(count,FUN = sum))
```

```
## Error in ddply(InsectSprays, .(spray), summarize, sum = ave(count, FUN = sum)): could not find functi
```