

Aquisição de Arquivos e Datasets

Reshiping Data

Delermendo Branquinho Filho

O objetivo é os dados arrumados

1. Cada variável forma uma coluna
 2. Cada observação forma uma linha
 3. Cada tabela / arquivo armazena dados sobre um tipo de observação (por exemplo, pessoas / hospitais).
-

Comece com a remodelação

```
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

Melting data frames

```
mtcars$carname <- rownames(mtcars)
carMelt <- melt(mtcars,id=c("carname","gear","cyl"),measure.vars=c("mpg","hp"))
head(carMelt,n=3)
```

##		carname	gear	cyl	variable	value
## 1		Mazda RX4	4	6	mpg	21.0
## 2		Mazda RX4 Wag	4	6	mpg	21.0
## 3		Datsun 710	4	4	mpg	22.8

```
tail(carMelt,n=3)
```

##		carname	gear	cyl	variable	value
## 62		Ferrari Dino	5	6	hp	175
## 63		Maserati Bora	5	8	hp	335
## 64		Volvo 142E	4	4	hp	109

Modelos de Molduras de Dados

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

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

```
cylData
```

```
##   cyl mpg hp
```

```
## 1   4  11 11
```

```
## 2   6   7  7
```

```
## 3   8  14 14
```

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

```
cylData
```

```
##   cyl      mpg      hp
```

```
## 1   4 26.66364 82.63636
```

```
## 2   6 19.74286 122.28571
```

```
## 3   8 15.10000 209.21429
```

Valores medios

```
head(InsectSprays)
```

```
##   count spray
```

```
## 1    10     A
```

```
## 2     7     A
```

```
## 3    20     A
```

```
## 4    14     A
```

```
## 5    14     A
```

```
## 6    12     A
```

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

```
##   A  B  C  D  E  F
```

```
## 174 184  25  59  42 200
```

Outra maneira - dividir

```
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

sprCount = lapply(spIns,sum)
sprCount

## $A
## [1] 174
##
## $B
## [1] 184
##
## $C
## [1] 25
##
## $D
## [1] 59
##
## $E
## [1] 42
##
## $F
## [1] 200
```

Outra maneira - 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
```

Outra maneira - plyr package

```
library("plyr")
```

```
## Warning: package 'plyr' was built under R version 3.3.3
```

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

```
##   spray sum
## 1     A 174
## 2     B 184
## 3     C  25
```

```
## 4      D  59
## 5      E  42
## 6      F 200
```

Criando uma nova variável

```
spraySums <- ddply(InsectSprays,.(spray),summarize,sum=ave(count,FUN=sum))
dim(spraySums)
```

```
## [1] 72  2
```

```
head(spraySums)
```

```
##   spray sum
## 1     A 174
## 2     A 174
## 3     A 174
## 4     A 174
## 5     A 174
## 6     A 174
```