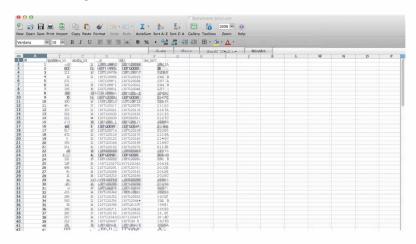


Reshaping data

Jeffrey Leek Johns Hopkins Bloomberg School of Public Health

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).

Ihttp://vita. had .co. nz/papers/tidy-data. pd

!Leek, Taub, and Pineda 2011 PLoS One

Start with reshaping

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

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

Melting data frames

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

```
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
```

html

Casting data frames

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

```
cyl mpg hp
1 4 11 11
2 6 7 7
3 8 14 14
```

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

```
cyl mpg hp

1 4 26.66 82.64

2 6 19.74 122.29

3 8 15.10 209.21
```

Averaging values

```
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,surn)
```

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

Another way - split

```
spins = split(InsectSprays$count,InsectSprays$spray)
spins
```

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

Another way - apply

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

```
$A
   174
$B
    184
$C
   25
$D
    59
$E
[1]
    42
$F
                                                                                                               8/12
    200
```

Another way - combine

```
unlist(sprCount)
```

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

```
sapply(spins,surn)
```

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

Another way - plyr package

```
ddply(InsectSprays,.(spray),sumrnarize,sum-sum(count))
```

```
spray sum

1  A 174

2  B 184

3  C 25

4  D 59

5  E 42

6  F 200
```

Creating a new variable

```
spraySums <- ddply(InsectSprays,.(spray),sumrnarize,sum=ave(count,FUN=sum))
dim(spraySums)</pre>
```

```
[1] 72 2
```

```
head(spraySums)
```

More information

- A tutorial from the developer of plyr http://plyr.had.co.nz/09-u
- · A nice reshape tutorial http://www.slideshare.net/jeffreybreen/reshaping-data-in-
- · A good plyr primer ffi!p://www.r-bloggers.com/a-guick-primer-on-split-apply-combine-proble
- See also the functions
 - acast for casting as multi-dimensional arrays
 - arrange for faster reordering without using order() commands
 - mutate adding new variables