plyr: *divide et impera*

Hadley Wickham

May 3, 2008

plyr is a set of tools that solves a common set of problems: you need to break a big problem down into manageable pieces, operate on each pieces and then put all the pieces back together. This paper describes the components that make up plyr.

1 Introduction

This avoids any ambiguity about what you'll get back from one of these functions.
divide and conquer, Divide and marriage before conquest,

2 Input

(Keep brief as possible. Maybe this and output should be in one big table.)
The type of input determines the ways it can be split up:

- data.frame, by variables
- vector/matrix/array, by margins
- list, by itself

2.1 Data frames (d)

Split by variables. .() notation

2.2 Arrays (a)

By margins. Same as apply

2.3 Lists (l)

mply

3 Process

conditions on function
  how extra arguments are passed in
  explode/splat
  each colwise failwith
  progress bars

4 Output

4.1 Data frames (d)

Function needs to return atomic vector of fixed size or data.frame.
  Extra variable will be added according to splits.

4.2 Arrays (a)

Function needs to return atomic of vector/matrix/array of fixed size/shape, or a list.
  Each splitting criterion creates a new dimension. Dimensions from function are added onto the end. This ensures idempotency

4.3 Lists (l)

Not much to talk about. dl -¿ ld will be labelled the same as dd.

4.4 Ignored (_)

Sometimes you are operating on a list purely for the side effects (e.g. plots, caching, output to screen/file). A little more memory efficient than simply abandoning the output of *lply because it doesn't construct the intermediate storage.

5 Equivalence to existing R functions

- lapply → llply
- sapply → laply, llply
- apply → aaply, alply
- mapply → maply, mlply
- by → dlply
- aggregate → daply(, colwise(f))
- tapply
- ave

- sweep
- lapply(split(df, ...), f) → dlply
- do.call("rbind", lapply(split(df, ...), f)) → ddply

```
* aggregate(mtcars, list(mtcars$cyl), median)
  daply(mtcars, .(cyl), colwise(median))
  daply(mtcars, .(cyl), colwise(median, .if = is.numeric))

* p <- function(df) coef(lm(mpg ~ wt, data = df))
  do.call("rbind", lapply(split(mtcars, mtcars$cyl), p))
  ddply(mtcars, .(cyl), p)
```

The cast function in the reshape package (Wickham, 2005) is a special case of aaply, which provides a number of nice labelling features.

6 Strategy

Take a small dataset, that you can easily solve.

6.1 Case study: baseball data

Calculating stint.
Model for each player

References

H. Wickham. *reshape: Flexibly reshape data.*, 2005. URL http://had.co.nz/reshape/. R package version 0.7.1.