Introducción a R

Luciano Selzer

21 September, 2016

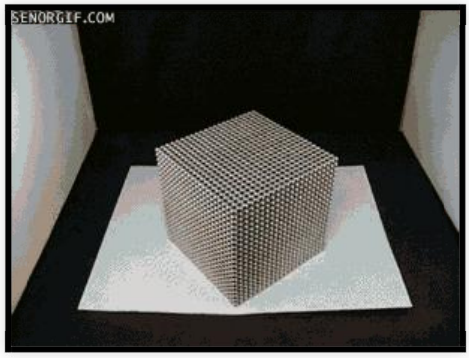
Anteriormente vimos como las funciones simplifican nuestro código.

Definimos la función `calcPBI` que calcula el PBI y le agregamos dos argumentos para poder calcular por año y/o país

```
# Toma el set de datos y multiplica la columna
# población por PBI per capita
calcPBI <- function(dat, year = NULL, country =
  if(!is.null(year)) {
    dat <- dat[dat$year %in% year, ]
  }
  if (!is.null(country)) {
    dat <- dat[dat$country %in% country,]
  }
  gdp <- dat$pop * dat$gdpPercap
```

Divide y vencerás

Muchas veces queremos hacer los cálculos u operaciones por grupo.
Arriba calculamos el PBI multiplicando dos columnas. ¿Y si quisieramos calcular el PBI medio por continente?



UNTDF ICPA

Podríamos ejecutar `calcGPD` y luego calcular la media de cada continente:

```
conPBI <- calcPBI(gapminder)
mean(conPBI[conPBI$continent == "Africa", "gdp"]
```

```
[1] 20904782844
```

```
mean(conPBI[conPBI$continent == "Americas", "gdp
```

```
[1] 379262350210
```

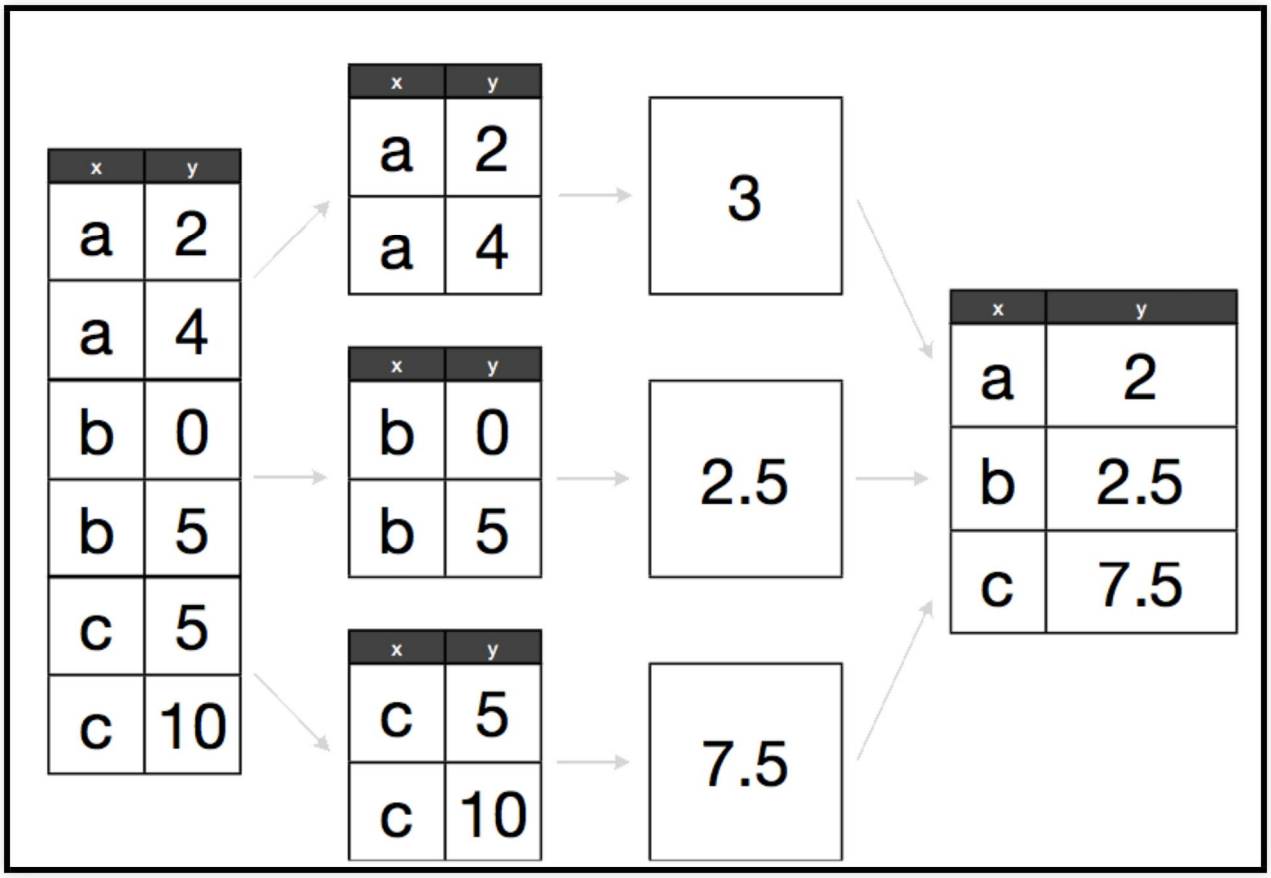
```
mean(conPBI[conPBI$continent == "Asia", "gdp"])
```

```
[1] 227233738153
```

Pero no es muy *lindo*. Usando una función disminuimos la repetición. Eso **está** bueno.
Pero hay mucha repetición: lleva tiempo, ahora y más adelante, y puede introducir errores.

Podríamos escribir una nueva función que sea flexible como `calcPBI`, el esfuerzo sería considerable y muchas pruebas para hacerlo bien.

El problema que tenemos se conoce como “divide-aplica-combina”:



El paquete `plyr`

Familia de funciones `apply`.

El paquete plyr provee un set de herramientas que hacen que sea más amigable lidiar con este problema.

```
library(plyr)
```

Plyr tiene funciones para operar en `listas`, `data.frames` y `arreglos` (matrices, o vectores n-dimensionales). Cada función: Plyr has functions for operating on `lists`, `data.frames` and `arrays` (matrices, or n-dimensional vectors). Each function performs:

1. Una operación de **división**.
2. **Aplica** una función en cada división.
3. Re**combina** las salidas como un solo objeto.

El nombre de la funciones depende de lo que esperan como entrada, y la estructura de salida.

	array	data frame	list	nothing
array	aaply	adply	alply	a_ply
data frame	daply	ddply	dlply	d_ply
list	laply	ldply	llply	l_ply
n replicates	raply	rdply	rlply	r_ply
function arguments	maply	mdply	mlply	m_ply

Cada función de **ply (`daply`, `ddply`, `llply`, `laply`, ...) tiene la misma estructura y las mismas 4 caraterísticas clave y estructura:

```
**ply(.data, .variables, .fun)
```

- La primera letra correponde al tipo de entrada y la seguna el tipo de salida
- .data - el objeto a ser procesado
- .variables - identifica las variables para dividir
- .fun - da la función a ser ejecutada en cada pedazo

Ahora podemos ejecutar rápidamente la media de PBI por continente:

```
ddply(
 .data = calcPBI(gapminder),
 .variables = "continent",
 .fun = function(x) mean(x$gdp)
)
```

```
  continent            V1
1    Africa  20904782844
2  Americas 379262350210
3      Asia 227233738153
4    Europe 269442085301
5   Oceania 188187105354
```

¿Qué tal si quisieramos otro tipo de salida?

```
dlply(
  .data = calcPBI(gapminder),
  .variables = "continent",
  .fun = function(x) mean(x$gdp)
)
```

```
$Africa
[1] 20904782844

$Americas
[1] 379262350210

$Asia
[1] 227233738153

$Europe
[1] 269442085301
```

Llamamos la misma función de nuevo, pero cambiamos la segunda letra a `l`, por lo que la salida es devuelta como una lista.

Podemos especificar varias columnas por grupo:

```
ddply(
 .data = calcPBI(gapminder),
 .variables = c("continent", "year"),
 .fun = function(x) mean(x$gdp)
)
```

```
   continent year           V1
1     Africa 1952   5992294608
2     Africa 1957   7359188796
3     Africa 1962   8784876958
4     Africa 1967  11443994101
5     Africa 1972  15072241974
6     Africa 1977  18694898732
7     Africa 1982  22040401045
8     Africa 1987  24107264108
9     Africa 1992  26256977719
10    Africa 1997  30023173824
11    Africa 2002  35303511424
12    Africa 2007  45778570846
13  Americas 1952 117738997171
14  Americas 1957 140817061264
15  Americas 1962 169153069442
16  Americas 1967 217867530844
17  Americas 1972 268159178814
18  Americas 1977 324085389022
19  Americas 1982 363314008350
20  Americas 1987 439447790357
21  Americas 1992 489899820623
22  Americas 1997 582693307146
23  Americas 2002 661248623419
24  Americas 2007 776723426068
25      Asia 1952  34095762661
26      Asia 1957  47267432088
```

```
28      Asia 1967  84648519224
29      Asia 1972 124385747313
30      Asia 1977 159802590186
31      Asia 1982 194429049919
32      Asia 1987 241784763369
33      Asia 1992 307100497486
34      Asia 1997 387597655323
35      Asia 2002 458042336179
36      Asia 2007 627513635079
37    Europe 1952  84971341466
38    Europe 1957 109989505140
```

```
daply(
 .data = calcPBI(gapminder),
 .variables = c("continent", "year"),
 .fun = function(x) mean(x$gdp)
)
```

```
           year
continent          1952          1957          196
  Africa     5992294608    7359188796    878487695
  Americas 117738997171 140817061264 16915306944
  Asia       34095762661   47267432088   6013686901
  Europe     84971341466 109989505140 13898469309
  Oceania    54157223944   66826828013   8233645324
           year
continent          1972          1977          198
  Africa     15072241974   18694898732   2204040104
  Americas 268159178814 324085389022 36331400835
```

Podemos llamar a estas funciones en lugar de bucles `for` (y generalmente es más rápido). Para reemplazar un bucle `for`, pon el código del cuerpo del bucle dentro una función anónima.

```
d_ply(
  .data = gapminder,
  .variables = "continent",
  .fun = function(x) {
    meanGDPperCap <- mean(x$gdpPercap)
    print(paste(
      "The mean GDP per capita for", unique(x$co
      "is", format(meanGDPperCap, big.mark = ","
   ))
  }
)
```

```
[1] "The mean GDP per capita for Africa is 2,193
[1] "The mean GDP per capita for Americas is 7,1
[1] "The mean GDP per capita for Asia is 7,902.1
[1] "The mean GDP per capita for Europe is 14,46
[1] "The mean GDP per capita for Oceania is 18,6
```

Tip: Imprimiendo números

La función `format` puede ser usada para hacer los números que quedn “bien” para imprimir mensajes.

Ejercicio 1

Calcula la expectativa de vida promedio por contiente. ¿Cuál es la mayor? ¿Cuál es la menor?

Ejercicio 2

Calcula la expectativa de vida promedio por continente y por año. ¿Cual tuvo la expectativa más corta y más larga en 2007? ¿Cual tuvo el mayor cambio entre 1952 y 2007?

Ejercicio Avanzado

Calcula la diferencia de medias entre la expectativa de vida en los años 1952 y 2007 usando la salida del ejercicio 2 usando una de las funciones de `plyr`.