Reforma de datos

Carlos Ivan Espinosa

7 de mayo de 2019

Contents

Pueden descargar este documento en pdf haciendo clic aquí

Prologo

Una de las principales preocupaciones de los investigadores durante los últimos años es como poder preservar los datos. Aunque algunos sistemas de procesamiento de datos como excel ofrece unas interesantes herramientas, uno de los principales problemas que tiene excel es la deficiente trazabilidad (el no saber todos los procesos que se siguieron para obtener los datos) que ofrece. Esto es un grave problema ya que atenta justamente con la preservación de los datos.

De esta forma podemos utilizar excel o cualquier otro software para sistematizar y almacenar los datos brutos, pero el procesamiento de datos debería hacérselo en sistemas que permitan mantener la trazabilidad. `R` ofrece una interesante oportunidad ya que permite mantener una alta trazabilidad y reducir la generación de archivos intermedios.

En el presente documento intentaremos dar unos primeros consejos sobre la reforma de datos, como a partir de datos brutos se puede reformar los datos para desarrollar análisis.

Reforma de datos

Nos referimos a reformar o manipular datos cuando generamos procesos de reestructuración. Muchas veces los datos brutos están en formatos conocidos como **datos de formato largos** y lo que nosotros necesitamos es obtener **datos de formato ancho**.

Table 1: Ejemplo de datos de formato largo (long data)

cod	especie	parcela	subparcela	area.basal	dap.a	no.fustes
S1_A_1	Tabebuia.chrysantha	S1	A	870.66975	33.295173	1
S1_A_11	Eriotheca.ruizii	S1	A	363.64991	21.517722	1
S1_A_12	Bursera.graveolens	S1	A	128.60036	12.796042	1
S1_A_13	Cochlospermum.vitifolium	S1	A	34.42839	6.620838	1

cod	especie	parcela	subparcela	area.basal	dap.a	no.fustes
S1_A_14	Erythroxylum.glaucum	S1	A	42.09648	7.321118	1
S1_A_15	Cynophalla.mollis	S1	A	65.09118	9.103652	1
S1_A_16	Tabebuia.chrysantha	S1	A	58.44248	8.626187	1
S1_A_17	Cynophalla.mollis	S1	A	30.57048	6.238866	1
S1_A_18	Geoffroea.spinosa	S1	A	69.25229	9.390130	1
S1_A_19	Cynophalla.mollis	S1	A	52.56012	8.180554	1

Los datos de formato largo tienen una columna para los posibles tipos de variables y una columna para los valores de esas variables, en el caso del ejemplo tenemos varias especies con datos para cada una de esas especies. Los datos de formato largo no son necesariamente solo dos columnas y pueden tener múltiples variables, como en el caso del ejemplo. Sin embargo, si por ejemplo con estos datos necesito desarrollar un estudio de comunidad, me interesa que la variable **especies**, se convierta en varias variables, tantas como especies hay. De esta manera, necesitamos transformar el formato largo a un formato ancho como en la siguiente tabla.

Table 2: Ejemplo de datos con formato ancho (wide data)

subparcela	Armatocereus.sp	Bursera.graveolens	Caesalpinia.glabrata	Ceiba.trichistandra
A	0	5	0	0
B	0	1	0	0
C	0	3	0	0
D	0	5	0	0
E	0	1	0	0
F	0	0	0	0
G	2	0	0	0
H	0	1	0	0
I	0	0	0	0
J	0	0	0	0

Como vemos tenemos unos casos que se denominan subparcela, y unas variables que son las especies.

A continuación, vamos a mostrar cómo realizar los cambios de formatos largos a anchos.

Datos de formato largo a formato ancho

Para este poder reformar los datos vamos a utilizar el paquete `reshape2`, este paquete tiene varias funciones que permiten reformar los datos.

El cambio de formato largo a ancho lo realizaremos a través de la función **dcast()**. Esta función está compuesta por varios argumentos que se debe proveer para realizar la reforma de los datos; *dcast(datos, fórmula, variable de valor)*

El primer argumento *datos* se refiere a la matriz que voy a reformar. El segundo argumento es una explicación de cómo quiero transformar los datos, en el caso del ejemplo quiero poner como casos las subparcelas y como variables las especies, así la formula sería *subparcelas~especies*, la *variable valor* es optativa, y se refiere a los datos que quiero que se muestren en la tabla ancha. Si no incluyo esa variable, los valores que asoman en cada caso corresponderán a la frecuencia de esa variable.

```
library(readxl)
library(reshape2)
```

```
dta <- read_excel("REA.datos.xlsx")
```

Cargamos los datos, estos datos corresponden a una parcela permanente de bosque seco. Puede revisar la estructura de esta matriz de datos. Ahora bien, queremos transformar estos datos a datos con formato ancho, manteniendo las variables de diseño (parcela y subparcela) y tener como variables a las diferentes especies.

```
dtaT <- dcast(dta, parcela+subparcela~especie)
knitr::kable(dtaT[1:10,1:5], caption= "Subconjunto de los datos
             con formato ancho de frecuencia")
```

Table 3: Subconjunto de los datos con formato ancho de frecuencia

parcela	subparcela	Achatocarpus.pubescens	Armatocereus.sp	Aster.desconocida
S1	A	0	0	0
S1	B	0	0	0
S1	C	0	0	0
S1	D	0	0	0
S1	E	0	0	0
S1	F	0	0	0
S1	G	0	2	0
S1	H	0	0	0
S1	I	0	0	0
S1	J	0	0	0

Como vemos ahora tenemos las variables de diseño frente a las variables especie, como datos aparece la frecuencia en la que cada especie aparece en la subparcela, puede ver la matriz completa usando la función **view(dtaT)**

Bien ahora es posible que me interese medir la ocurrencia de las especies como un valor de biomasa, puesto que tengo un valor de área basal usaré esta medida como indicador de biomasa.

```
dtaTab <- dcast(dta, parcela+subparcela~especie, value.var = "area.basal", fun.aggregate = sum)
knitr::kable(dtaTab[1:10,1:5], caption= "Subconjunto de los datos
             con formato ancho de biomasa")
```

Table 4: Subconjunto de los datos con formato ancho de biomasa

parcela	subparcela	Achatocarpus.pubescens	Armatocereus.sp	Aster.desconocida
S1	A	0	0.00000	0
S1	B	0	0.00000	0
S1	C	0	0.00000	0
S1	D	0	0.00000	0
S1	E	0	0.00000	0
S1	F	0	0.00000	0
S1	G	0	47.63348	0
S1	H	0	0.00000	0
S1	I	0	0.00000	0
S1	J	0	0.00000	0

En este caso la matriz resultante tiene datos de la suma del área basal. A diferencia del primer código, en esta ocasión hemos incrementado dos argumentos; *value.var* (variable de valor) que corresponde a la variable que debería poner en la matriz y *fun.aggregate* que corresponde a la función que `R` usa para agregar (juntar) los datos, en este caso la suma. Podríamos usar otras medidas como la media, esto dependerá del uso que le quiera dar a los datos y de la interpretación biológica.

Datos de formato ancho a formato largo

Bien ahora vamos a ver un ejemplo opuesto, donde tenemos datos disgregados como varias variables y realmente queremos que sea una sola variable. En este caso usaremos la función **melt()**. Esta función está compuesta por al menos dos argumentos; **melt(datos, id.vars)**. Id.vars se refiere a las variables que deseo se mantengan en formato largo.

```
dtaL <- melt(dtaT, id.vars=c("parcela", "subparcela"),
             variable.name = "especie",
             value.name ="frecuencia")

knitr::kable(dtaL[1:6,], caption= "Subconjunto de los datos
             con formato largo de frecuencia")
```

Table 5: Subconjunto de los datos con formato largo de frecuencia

parcela	subparcela	especie	frecuencia
S1	A	Achatocarpus.pubescens	0
S1	B	Achatocarpus.pubescens	0
S1	C	Achatocarpus.pubescens	0
S1	D	Achatocarpus.pubescens	0
S1	E	Achatocarpus.pubescens	0
S1	F	Achatocarpus.pubescens	0

Como vemos ahora especies volvió a ser una sola variable, aunque no podemos recuperar el formato inicial debido a que los casos en este caso están a nivel de subparcela.

Agregación de datos

La agregación de los datos nos permite colapsar los datos en unidades superiores, así, por ejemplo, podríamos colapsar los datos que tenemos a nivel de subparcelas a nivel de parcelas, en otras palabras, dejar como casos las parcelas y no las subparcelas. Para realizar la agregación usaremos la función **aggregate()** la cual necesita al menos tres argumentos. **aggregate(datos, list(agregación), función de agregación)**.

Usaremos nuestra matriz en formato ancho de frecuencia para colapsar los datos a nivel de parcela.

```
dtaA <- aggregate(dtaT[,-(1:2)], list(par=dtaT$parcela), sum)

knitr::kable(dtaA[,1:6], caption= "Subconjunto de los datos agregados
             a parcela")
```

Table 6: Subconjunto de los datos agregados a parcela

par	Achatocarpus.pubescens	Armatocereus.sp	Aster.desconocida	Bursera.graveolens	Byttneria.flexuosa
S1	0	8	0	18	0
S2	9	265	0	8	2
S3	2	14	0	12	0
S4	20	38	0	12	1
S5	32	21	0	13	0
S6	9	9	2	23	0
S7	0	12	0	14	0
S8	0	16	0	5	0
S9	0	8	0	8	0

En el ejemplo, queremos agregar los datos de especies, sin embargo, la 1era y 2da variables no son especies por lo que las excluyo de los datos (dtaT[,-(1:2)]), uso una lista del vector *parcela* a la cual le he asignado el nombre *par* (list(par=dtaT$parcela)), finalmente le he dicho que para realizar el colapso a nivel de parcela haga una suma de los elementos de la subparcela.

Ahora podríamos necesitar hacer agregaciones, pero bajo una estructura un poco más compleja. Vamos a usar los datos brutos (dta) para colapsar los datos a nivel de subparcela.

```
dtaA2 <- aggregate(dta[, c("area.basal","dap.a", "no.fustes")],
                   list(par=dta$parcela,
                        subpar=dta$subparcela,
                        especie=dta$especie),
                   sum)

knitr::kable(dtaA2[1:7,], caption= "Subconjunto de los datos agregados
             a parcela/subparcela/especie")
```

Table 7: Subconjunto de los datos agregados a parcela/subparcela/especie

par	subpar	especie	area.basal	dap.a	no.fustes
S2	A	Achatocarpus.pubescens	28.96620	6.072958	3
S2	B	Achatocarpus.pubescens	30.67711	6.249738	5
S4	B	Achatocarpus.pubescens	44.98355	7.568005	5
S5	B	Achatocarpus.pubescens	264.01733	36.230280	13
S3	C	Achatocarpus.pubescens	75.84369	13.895647	6
S4	C	Achatocarpus.pubescens	87.29848	18.174422	8
S5	C	Achatocarpus.pubescens	275.79161	44.905386	17

Lo que en este ejemplo hemos hecho es colapsar los datos de área basal, dap y número de fustes a nivel de especies, por subparcela y por parcela. Hemos usado la función *sum* para colapsar los datos. El cambio implica que antes tenía los datos a nivel de individuo y ahora los tengo a nivel de especie.

Aunque este procedimiento puede servir en muchas casos, cabe la posibilidad que queremos que la función de agregación no sea una sino varias. Vamos a ver como podemos hacer esto.

```
##Creamos una función que contenga varias funciones de resumen
```

```r
funvar<- function(v){
  c(mean=mean(v, na.rm = T),
    range= max(v, na.rm = T)-min(v, na.rm = T),
    sd=sd(v, na.rm = T)
    )
}

#Ahora usamos la función aggregate
dtaA3 <- data.frame(aggregate(dta[ c("area.basal","dap.a", "no.fustes")],
                  list(par=dta$parcela,
                       especie=dta$especie),
                  FUN= funvar))

#Nos arroja una lista como resultado
mode(dtaA3)
```

```
## [1] "list"
```

```r
##Transformamos a dta.frame
dtaA3c <- cbind.data.frame(dtaA3$especie, dtaA3$area.basal,
                           dtaA3$dap.a,dtaA3$no.fustes)
#ponemos los nombres de las variables
colnames(dtaA3c) <- c(names(dtaA3)[2],
                paste(rep(names(dtaA3)[-(1:2)],each=3),
                      colnames(dtaA3$area.basal), sep="."))

knitr::kable(dtaA3c[1:5,1:5], caption= "Subconjunto de los datos
             agregados a parcela/especie con varias funciones de agregación")
```

Table 8: Subconjunto de los datos agregados a parcela/especie con varias funciones de agregación

especie	area.basal.mean	area.basal.range	area.basal.sd	dap.a.mean
Achatocarpus.pubescens	27.98324	12.985450	3.647384	5.956888
Achatocarpus.pubescens	37.92184	2.312521	1.635199	6.947824
Achatocarpus.pubescens	47.12016	87.046585	28.684197	7.459515
Achatocarpus.pubescens	46.43758	76.458822	21.051361	7.505835
Achatocarpus.pubescens	92.41623	434.648123	138.232639	9.424807

Ejercicios

Con los datos de amebiasis reportados en la provincia de Loja que puede descargar haciendo clic aquí, reforme de la siguiente manera los datos:

1. Genere un nuevo vector categórico que corresponda con la edad de los afectados. Así tendremos las siguientes categorías; infantes de 1 a 5 años, jóvenes de 5 a 18 años, adultos de 18 a 65 años y mayores a personas mayores a 65 años.
2. Obtenga dos matrices. La primera matriz debería contener la frecuencia de afectación a nivel de categoría de edad y Parroquia. La segunda matriz frecuencia de afectación a nivel de género y parroquia.

3. Obtenga dos matrices la primera transforme de datos con formato largo a formato ancho usando el género como variable a disgregar y la segunda matriz con la categoría de edad como variable a disgregar obtenga la frecuencia de ocurrencia de la amebiasis. Mantenga los datos a nivel de parroquia.