

Manipulación de data.frame con tidyr

Luciano Selzer

21 September, 2016



Muchas veces necesitamos manipular nuestros datos entre el formato “ancho” y “largo” En el formato “largo”:

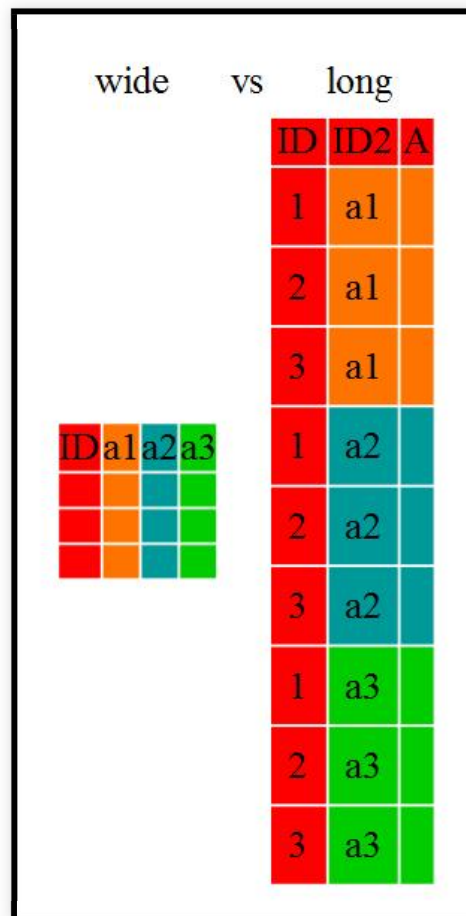
- cada columna es una variable
- cada fila es una observación

Una columna para la variable observada y las otras otras columnas son variables de identificación.

En el formato ancho cada fila es un paciente, sujeto, o sitio y hay muchas observaciones de cada uno.

- Múltiples observaciones en el tiempo
- Múltiples variables
- Una mezcla de ambas

Quizás sea más fácil entrar los datos en formato ancho.
Pero muchas funciones de R necesitan los datos en formato largo



Para los humanos es más fácil leer los datos en formato ancho, pero para las computadoras es más fácil el formato largo.

Las bases de datos están en formato largo, con cada fila una observación. En el formato largo todos los datos están en columnas, en cambio en el formato ancho parte de los datos está en los nombres de columna.

Empezando

Instala los paquetes si no los tienes

```
install.packages("tidyr")  
install.packages("dplyr")
```

Y cargalos

```
library("tidyr")  
library("dplyr")
```



Veamos como están dispuestos los datos:

```
str(gapminder)
```

```
'data.frame':   1704 obs. of  6 variables:
 $ country   : chr  "Afghanistan" "Afghanistan" " "
 $ year      : int   1952 1957 1962 1967 1972 1977
 $ pop       : num   8425333 9240934 10267083 1153
 $ continent: chr   "Asia" "Asia" "Asia" "Asia" .
 $ lifeExp   : num   28.8 30.3 32 34 36.1 ...
 $ gdpPercap: num    779 821 853 836 740 ...
```



Ejercicio 1

¿Gapminder está en formato largo, ancho o intermedio?



A veces, tenemos formatos intermedios:

ID:	Observaciones:
continent	pop
country	lifeExp
year	gdpPercap

Es un buen formato porque las tres observaciones tienen diferentes unidades

Nota: Algunas funciones de gráficos en R sí funcionan mejor con datos en formato ancho.

De formato ancho a largo gather()

Hasta ahora usamos una versión cómoda de gapminder, pero en la vida real vamos a tener datos en formato ancho.

Vamos a cargar una versión en este formato. Pero no queremos que `continent` y `country` sean factores así que lo leemos con `stringsAsFactors = FALSE`.



```
gap_wide <- read.csv("data/gapminder_wide.csv",
str(gap_wide)
```

```
'data.frame':    142 obs. of  38 variables:
 $ continent      : chr  "Africa" "Africa" "Africa"
 $ country        : chr  "Algeria" "Angola" "Benin"
 $ gdpPercap_1952: num   2449  3521 1063  851  543 .
 $ gdpPercap_1957: num   3014  3828  960  918  617 ..
 $ gdpPercap_1962: num   2551  4269  949  984  723 ..
 $ gdpPercap_1967: num   3247  5523 1036 1215  795
 $ gdpPercap_1972: num   4183  5473 1086 2264  855
 $ gdpPercap_1977: num   4910  3009 1029 3215  743
 $ gdpPercap_1982: num   5745  2757 1278 4551  807
 $ gdpPercap_1987: num   5681  2430 1226 6206  912
 $ gdpPercap_1992: num   5023  2628 1191 7954  932
 $ gdpPercap_1997: num   4797  2277 1233 8647  946
 $ gdpPercap_2002: num   5288  2773 1373 11004 103
 $ gdpPercap_2007: num   6223  4797 1441 12570 121
 $ lifeExp_1952   : num   43.1  30  38.2  47.6  32 ...
 $ lifeExp_1957   : num   45.7  32  40.4  49.6  34.9 .
 $ lifeExp_1962   : num   48.3  34  42.6  51.5  37.8 .
 $ lifeExp_1967   : num   51.4  36  44.9  53.3  40.7 .
 $ lifeExp_1972   : num   54.5  37.9  47  56  43.6 ...
 $ lifeExp_1977   : num   58  39.5  49.2  59.3  46.1 .
 $ lifeExp_1982   : num   61.4  39.9  50.9  61.5  48.1
 $ lifeExp_1987   : num   65.8  39.9  52.3  63.6  49.6
 $ lifeExp_1992   : num   67.7  40.6  53.9  62.7  50.3
 $ lifeExp_1997   : num   69.2  41  54.8  52.6  50.3 .
 $ lifeExp_2002   : num   71  41  54.4  46.6  50.6 ...
 $ lifeExp_2007   : num   72.3  42.7  56.7  50.7  52.3
 $ pop_1952       : num  9279525  4232095 1738315
 $ pop_1957       : num 10270856  4561361 1925173
 $ pop_1962       : num 11000948  4826015 2151895
 $ pop_1967       : num 12760499  5247469 2427334
 $ pop_1972       : num 14760787  5894858 2761407
 $ pop_1977       : num 17152804  6162675 3168267
 $ pop_1982       : num 20033753  7016384 3641603
 $ pop_1987       : num 23254956  7874230 4243788
```

```
$ pop_2007      : int  33333216 12420476 807831
```

wide format

continent	country	gdpPercap_1952	gdpPercap_1957	gdpPercap_...	lifeExp_1952	lifeExp_1957	lifeExp_...	pop_1952	pop_1957	pop_...
Africa	Algeria									
Africa	Angola									
...	...									

El primer paso hacia convertir nuestros datos a un formato intermedio es convertir de ancho a largo.

Con la función `gather()` de `tidyr` va a juntar las columnas de observaciones en una sola variable.

```
gap_long <- gap_wide %>%  
  gather(obstype_year, obs_values, starts_with('lifeExp'),  
         starts_with('lifeExp'), starts_with('lifeExp'),  
         str(gap_long))
```

```
'data.frame':   5112 obs. of  4 variables:  
 $ continent   : chr  "Africa" "Africa" "Africa"  
 $ country     : chr  "Algeria" "Angola" "Benin"  
 $ obstype_year: chr  "pop_1952" "pop_1952" "pop_1952"  
 $ obs_values  : num  9279525 4232095 1738315 44...
```

Dentro de `gather` el primer nombre es la nueva columna con la nueva variable de indentificación y la segunda con el nombre de la nueva columna que va a tener los valores.

Podríamos haber escrito todos los nombre, pero la función `starts_with()` nos ahorra trabajo. También podemos usar el signo `-` para indicar cuales columnas nos son variables.

long format

continent	country	obstype_year	obs_value
Africa	Algeria	gdpPercap_1952	
Africa	Algeria	gdpPercap_1957	
Africa	Algeria	gdpPercap_...	
Africa	Algeria	lifeExp_1952	
Africa	Algeria	lifeExp_1957	
Africa	Algeria	lifeExp_...	
Africa	Algeria	pop_1952	
Africa	Algeria	pop_1957	
Africa	Algeria	pop_...	
Africa	Angola	gdpPercap_1952	
Africa	Angola	gdpPercap_1957	
Africa	Angola	gdpPercap_...	
Africa	Angola	lifeExp_1952	
Africa	Angola	lifeExp_1957	
Africa	Angola	lifeExp_...	
Africa	Angola	pop_1952	
Africa	Angola	pop_1957	
Africa	Angola	pop_...	
Africa	...	gdpPercap_1952	
Africa	...	gdpPercap_1957	
Africa	...	gdpPercap_...	
Africa	...	lifeExp_1952	
Africa	...	lifeExp_1957	
Africa	...	lifeExp_...	
Africa	...	pop_1952	
Africa	...	pop_1957	
Africa	...	pop_...	


```
gap_long <- gap_wide %>% gather(obstype_year, obs  
str(gap_long)
```

```
'data.frame':    5112 obs. of  4 variables:  
 $ continent      : chr  "Africa" "Africa" "Africa"  
 $ country        : chr  "Algeria" "Angola" "Benin"  
 $ obstype_year   : chr  "gdpPercap_1952" "gdpPerca  
 $ obs_values     : num  2449 3521 1063 851 543 ...
```

Puede parecer trivial pero a veces tenes una variable identificatoria y 40 obsevaciones con nombres distintos.

Ahora `obstype_year` en verdad contiene dos cosas:

- la observación
- el año

Podemos usar la función `separate()` para dividir una cadena en múltiples variables.

```
gap_long <- gap_long %>%  
  separate(obstype_year,  
            into = c('obs_type', 'year'), sep = "_"  
gap_long$year <- as.integer(gap_long$year)
```



Ejercicio 2

Usando `gap_long`, calcula la expectativa de vida, población, `gdpPercap` para cada continente. **Pista:** usa las funciones `group_by()` y `summarize()` aprendidas en la lección de `dplyr`.



De formato largo a intermedio con `spread()`

Vamos a chequear que los datos sean iguales a los otros.

Vamos a usar el opuesto de `gather()` para extender nuestras variables observadas con la función `spread()`. Podemos hacerlo hasta el formato intermedio o el ancho.



```
gap_normal <- gap_long %>% spread(obs_type, obs_  
dim(gap_normal)
```

```
[1] 1704      6
```

```
dim(gapminder)
```

```
[1] 1704      6
```

```
names(gap_normal)
```

```
[1] "continent" "country"    "year"       "gdpPerCap"
```

```
names(gapminder)
```

```
[1] "country"    "year"       "pop"        "continent"
```

Ahora tenemos un formato intermedio `gap_normal`, con las mismas dimensiones pero el orden de las variables es distinto. Lo arreglamos antes de probar si son `all.equal()`.

```
gap_normal <- gap_normal[,names(gapminder)]  
all.equal(gap_normal, gapminder)
```

```
[1] "Component \"country\": 1704 string mismatch  
[2] "Component \"pop\": Mean relative difference  
[3] "Component \"continent\": 1212 string mismatch  
[4] "Component \"lifeExp\": Mean relative difference  
[5] "Component \"gdpPercap\": Mean relative difference"
```

```
head(gap_normal)
```

	country	year	pop	continent	lifeExp	gdpPer
1	Algeria	1952	9279525	Africa	43.077	2449.
2	Algeria	1957	10270856	Africa	45.685	3013.
3	Algeria	1962	11000948	Africa	48.303	2550.
4	Algeria	1967	12760499	Africa	51.407	3246.
5	Algeria	1972	14760787	Africa	54.518	4182.
6	Algeria	1977	17152804	Africa	58.014	4910.

```
head(gapminder)
```

	country	year	pop	continent	lifeExp	gdpPer
1	Afghanistan	1952	8425333	Asia	28.801	7
2	Afghanistan	1957	9240934	Asia	30.332	8
3	Afghanistan	1962	10267083	Asia	31.997	8
4	Afghanistan	1967	11537966	Asia	34.020	8
5	Afghanistan	1972	13079460	Asia	36.088	7
6	Afghanistan	1977	14880372	Asia	38.438	7

El original estaba ordenado por `country`, `continent`, luego `year`.

```
gap_normal <- gap_normal %>% arrange(country, con  
all.equal(gap_normal, gapminder)
```

```
[1] TRUE
```

Ahora convirtamos desde largo hasta ancho.

Vamos a conservar las variables identificatorias y extender todas las observaciones de las tres medidas

(`pop`, `lifeExp`, `gdpPercap`) y tiempo (`year`).

Primero necesitamos crear las etiquetas apropiadas para nuestras nuevas variables (tiempo*medida) y también unificar nuestras variables identificatorias pas simplificar el proceso:

```
gap_temp <- gap_long %>%  
  unite(var_ID, continent, country, sep = "_")
```

```
str(gap_temp)
```

```
'data.frame':   5112 obs. of  4 variables:
 $ var_ID      : chr  "Africa_Algeria" "Africa_Ang
 $ obs_type    : chr  "gdpPercap" "gdpPercap" "gdp
 $ year        : int   1952 1952 1952 1952 1952 195
 $ obs_values: num   2449 3521 1063 851 543 ...
```

```
gap_temp <- gap_long %>%  
  unite(ID_var, continent, country, sep = "_") %>%  
  unite(var_names, obs_type, year, sep = "_")  
str(gap_temp)
```

```
'data.frame':   5112 obs. of  3 variables:  
 $ ID_var      : chr  "Africa_Algeria" "Africa_Ang  
 $ var_names   : chr  "gdpPercap_1952" "gdpPercap_  
 $ obs_values  : num  2449 3521 1063 851 543 ...
```

Usando `unite()` ahora tenemos una sola columna ID que es combinación de `continent`, `country`, y definimos los nombres de las variables. ahora podemos entubar con `spread()`

```
gap_wide_new <- gap_long %>%
  unite(ID_var, continent, country, sep = "_")
  unite(var_names, obs_type, year, sep = "_")
  spread(var_names, obs_values)
str(gap_wide_new)
```

```
'data.frame': 142 obs. of 37 variables:
 $ ID_var      : chr  "Africa_Algeria" "Africa
 $ gdpPercap_1952: num  2449 3521 1063 851 543 .
 $ gdpPercap_1957: num  3014 3828 960 918 617 ..
 $ gdpPercap_1962: num  2551 4269 949 984 723 ..
 $ gdpPercap_1967: num  3247 5523 1036 1215 795
 $ gdpPercap_1972: num  4183 5473 1086 2264 855
 $ gdpPercap_1977: num  4910 3009 1029 3215 743
 $ gdpPercap_1982: num  5745 2757 1278 4551 807
 $ gdpPercap_1987: num  5681 2430 1226 6206 912
 $ gdpPercap_1992: num  5023 2628 1191 7954 932
 $ gdpPercap_1997: num  4797 2277 1233 8647 946
 $ gdpPercap_2002: num  5288 2773 1373 11004 103
 $ gdpPercap_2007: num  6223 4797 1441 12570 121
 $ lifeExp_1952  : num  43.1 30 38.2 47.6 32 ...
 $ lifeExp_1957  : num  45.7 32 40.4 49.6 34.9 .
 $ lifeExp_1962  : num  48.3 34 42.6 51.5 37.8 .
 $ lifeExp_1967  : num  51.4 36 44.9 53.3 40.7 .
 $ lifeExp_1972  : num  54.5 37.9 47 56 43.6 ...
 $ lifeExp_1977  : num  58 39.5 49.2 59.3 46.1 .
 $ lifeExp_1982  : num  61.4 39.9 50.9 61.5 48.1
 $ lifeExp_1987  : num  65.8 39.9 52.3 63.6 49.6
 $ lifeExp_1992  : num  67.7 40.6 53.9 62.7 50.3
 $ lifeExp_1997  : num  69.2 41 54.8 52.6 50.3 .
 $ lifeExp_2002  : num  71 41 54.4 46.6 50.6 ...
 $ lifeExp_2007  : num  72.3 42.7 56.7 50.7 52.3
 $ pop_1952      : num  9279525 4232095 1738315
 $ pop_1957      : num  10270856 4561361 1925173
 $ pop_1962      : num  11000948 4826015 2151895
 $ pop_1967      : num  12760499 5247469 2427334
 $ pop_1972      : num  14760787 5894858 2761407
 $ pop_1977      : num  17152804 6162675 3168267
 $ pop_1982      : num  20033753 7016384 3641603
 $ pop_1987      : num  23254956 7874230 4243702
```

```
$ pop_1997      : num  29072015  9875024  6066080
$ pop_2002      : num  31287142  10866106  7026111
$ pop_2007      : num  33333216  12420476  8078311
```



Ejercicio 3

Lleva esto un paso más lejos y crea

`gap_ridiculamente_ancho` extendiendo países, año y las tres medidas. **Pista** la nueva data.frame debería tener solo 5 filas



Ahora tenemos una dataframe ancho, pero la variable `ID_var` podría ser más usable, separemos en dos variables con `separate()`

```
gap_wide_betterID <- separate(gap_wide_new, ID_var,
                             c("continent", "country"),
gap_wide_betterID <- gap_long %>%
  unite(ID_var, continent, country, sep = "_")
  unite(var_names, obs_type, year, sep = "_")
  spread(var_names, obs_values) %>%
  separate(ID_var, c("continent", "country"),
```



```
str(gap_wide_betterID)
```

```
'data.frame':    142 obs. of  38 variables:
 $ continent      : chr   "Africa" "Africa" "Africa"
 $ country        : chr   "Algeria" "Angola" "Benin"
 $ gdpPercap_1952: num    2449  3521  1063  851  543  .
 $ gdpPercap_1957: num    3014  3828  960  918  617  .
 $ gdpPercap_1962: num    2551  4269  949  984  723  .
 $ gdpPercap_1967: num    3247  5523  1036  1215  795  .
 $ gdpPercap_1972: num    4183  5473  1086  2264  855  .
 $ gdpPercap_1977: num    4910  3009  1029  3215  743  .
 $ gdpPercap_1982: num    5745  2757  1278  4551  807  .
 $ gdpPercap_1987: num    5681  2430  1226  6206  912  .
```

```
all.equal(gap_wide, gap_wide_betterID)
```

```
[1] TRUE
```

¡Fuimos y volvimos!

Otros recursos útiles

- [Data Wrangling Cheat sheet](#)
- [Introduction to tidyr](#)

