Ejemplo de funciones en R: programación del sistema electoral d'Hondt

Este ejemplo fue el tema principal de una de las prácticas de laboratorio R de la asignatura Fundamentos de Programación del grado de Biotecnología. Por tanto, el siguiente contenido se obtuvo, en su mayoría, de ésta.

Introducción: sistema electoral d'Hondt

El sistema d'Hont es un método empleado para calcular el número de diputados que consigue cada partido tras la celebración de elecciones. Es utilizado en numerosos países en la actualidad, incluida España (en la que se hace separadamente en todas las circunscripciones). El método es el siguiente:

1. Se excluyen aquellos partidos con menos del 3% de los votos totales
2. Elaborar una tabla en la que cada fila corresponde cada partido, en la que la primera columna sean los votos de cada partido y las columnas posteriores sean estos votos divididos por el número de la columna.
3. Asignar escaños a las mayores cifras de la tabla de forma ordenada. El número de escaños de cada partido es igual al número de cifras con escaño en su correspondiente fila.

Ejemplo, para 8 escaños:

- Partido A: 40000 votos
- Partido B: 25000 votos
- Partido C: 8000 votos
- Partido D: 2000 votos (<3%=2250)

	/1	/2	/3	/4
Partido A	40000	20000	13333.333	10000
Partido B	25000	12500	8333.333	6250
Partido C	8000	4000	2666.667	2000

Partido A: 4 escaños; Partido B: 3 escaños; Partido C: 1 escaño

Objetivo

Nuestro objetivo es el de elaborar un código en R que calcule los escaños que corresponden a cada partido, dado el número de escaños totales, el número de partidos y los votos de cada uno. *Por comodidad omitiremos el criterio del 3%. Además, representaremos gráficamente los resultados de dos formas distintas.

Elaboración del código

En primer lugar, tenemos que elaborar una función, que dependa del número de partidos, número de escaños y votos por partido (ya que son los datos mínimos necesarios para aplicar el método d'Hondt); y que calcule los escaños.

Para ello, una de las múltiples opciones es la de buscar la mayor cifra a medida que añadimos columnas a la matriz del método d'Hondt. Esto resulta válido dado que el mayor elemento de una columna k siempre es mayor que el mayor elemento de la columna k+1. Visto de otro modo, en el caso más extremo, en el que un partido se lleva todos los escaños, en la iteración k le asignaremos un escaño al elemento k de la fila en cuestión. Por tanto, en el resto de casos, donde hay partidos con más votos que las fracciones de los votos de otros partidos, la forma en la que está estructurado el algoritmo sigue dando margen a que la asignación se realice de forma correcta. Esto lo podemos construir de la siguiente forma:

```
asignacion_escaños <- function ( votos, npart, nesc ){
k = 1   (número de la columna; coeficiente por el que dividiremos los votos)
cuenta_escan = 1    (variable para contar los escaños)
A <- matrix ( c ( 0 ) , nrow = npart , ncol = nesc )
escaños_unica <- c ( 0 )    (vector con los escaños de cada partido)
for ( i  in  1 : npart ) { escaños_unica [ i ] = 0 }    (asignar los ceros al vector)
while ( cuenta_escan <= nesc ) {
        A [ , k ] <- votos / k (creamos la columna k, que empieza en 1)
        max <- 0
        for ( i  in  1 : npart ) {
                for ( j  in  1 : k ) {
                        if ( A [ i , j ] >= max ) {
                                max <- A [i , j ]
                                imax <- i
                                jmax <- j
                        }
                }
        }   (buscamos el mayor elemento en lo que llevamos de tabla)
        A [ imax , jmax ] = 0    (lo fijamos a 0 para no volver a contarlo)
        escaños_unica [ imax ] <- escaños_unica [ imax ] + 1 (+1 voto al partido)
        cuenta_escan <- cuenta_escan + 1    (siguiente escaño)
        k <- k + 1    (pasamos de columna)
}
```

```
return ( escaños_unica )
}
```

Si no quedamos satisfechos con este modo de proceder, existen otras versiones igual de válidas. Como ya se habrá dado cuenta, lo más intuitivo sería crear la matriz entera y hacer un “buscador de mayores valores” a la par que fijamos a cero dichos valores. Sin embargo, esta forma del método consumiría una mayor capacidad de computación. Se escribiría tal que así:

```
asignacion_escaños<-function(votos,npart,nesc){
cuenta_escan = 1
A = matrix (c(0), nrow=npart, ncol=nesc)
escaños_unica = c(0)
for ( i  in  1 : npart ) { escaños_unica [ i ] = 0 }
for ( i  in  1 : nesc ) { A [ , i ] =votos/i}
while(cuenta_escan<=nesc){
        max<-0
        for ( i  in  1 : npart ) {
                for ( j  in  1 : nesc ) {
                        if ( A [ i , j ] >= max ) {
                                max = A [ i , j ]
                                imax = i
                                jmax = j
                        }
                }
        }
        A [ imax , jmax ] = 0
        escaños_unica [ imax ] = escaños_unica [ imax ] + 1
        cuenta_escan = cuenta_escan + 1
}
return ( escaños_unica )
}
```

En este caso, la variable k deja de ser necesaria dado que en la asignación de cada escaño no estamos añadiendo nuevas columnas a la matriz A (que actúa como la tabla del método de d’Hondt). Para los principiantes en la programación en R, la

estructura señalada con el color azul se trata de una estructura genérica para detectar “récords” o elementos de mayor valor numérico y sus posiciones en la matriz. Por otro lado, A[,i] es la forma de trabajar únicamente con la columna i de la matriz A.

Una vez tenemos nuestra función programada, podemos pasar a poner los datos con los que queremos trabajar:

```
npart <- 17

nesc <- 350

votos = c (6752983, 5019869, 3640063, 3097185, 1637540, 869934, 577055,
527375,377423, 276519, 244754, 226469, 123981, 119597, 98448, 68580,
19696)

Partido = c ('PSOE', 'PP', 'VOX', 'PODEMOS', 'Cs', 'ERC', 'MasPais', 'JxCAT',
'PNV', 'BILDU', 'CUP', 'PACMA', 'CC', 'BNG',
'NAVARRA_SUMA','PRC','TeruelExiste')
```

Una vez tenemos los datos, sólo tenemos que ejecutar el código:

```
escaños<-asignacion_escaños(votos,npart,nesc)

print ( escaños )
```

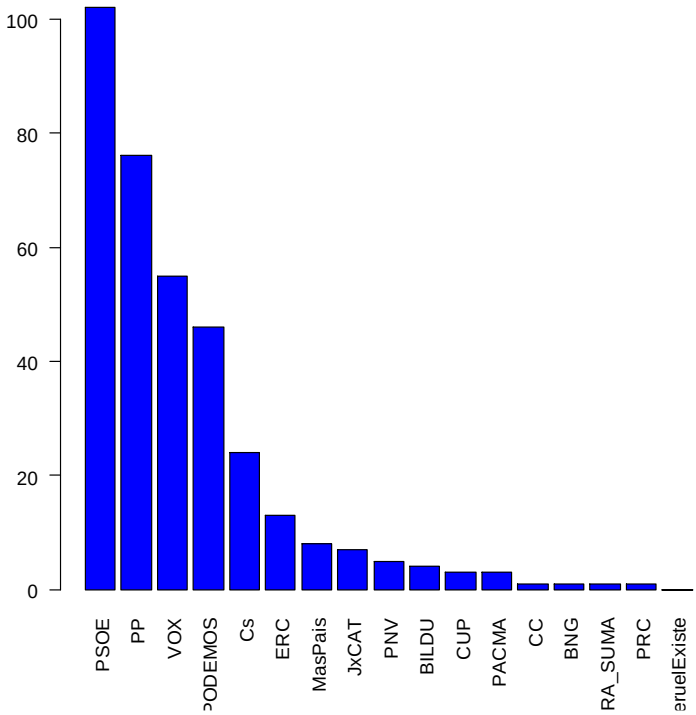
```
> escaños<-asignacion_escaños(votos,npart,nesc)
> escaños
 [1] 102  76  55  46  24  13   8   7   5   4   3   3   1   1   1   1   0
```

Por otro lado, si no sólo queremos calcular los escaños de los partidos, sino también representarlos gráficamente, podemos hacerlo, por ejemplo, por gráfico de barras, que en R se escribe:

```
*names(escaños)<-c ('PSOE', 'PP', 'VOX', 'PODEMOS', 'Cs', 'ERC', 'MasPais',
'JxCAT', 'PNV', 'BILDU', 'CUP', 'PACMA', 'CC', 'BNG', 'NAVARRA_SUMA',
'PRC', 'TeruelExiste')

barplot (escaños, beside=TRUE, col="blue", las=2)
```

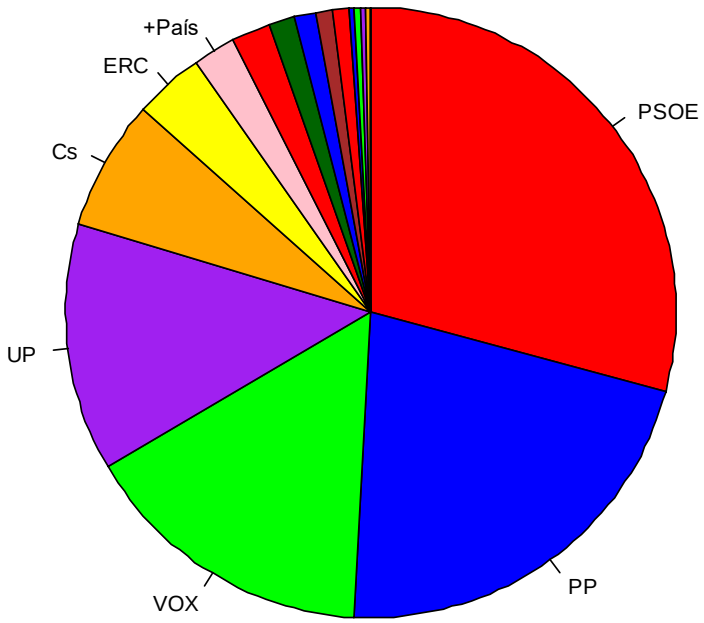
*Para que se vean los nombres de los partidos en el gráfico



o incluso por un gráfico circular, que en R se escribe:

```
names (escaños) <- c ("PSOE","PP","VOX","UP","Cs","ERC","+País")

Pie (escaños, clockwise=TRUE,
col=c('red','blue','green','purple','orange','yellow','pink','red','dark
green','blue','brown'), cex=0.8)
```



Por otro lado, existen otros sistemas electorales, como el método de Huntignton-Hill, utilizado en la Cámara de Representantes de los Estados Unidos. Según este sistema, los representantes obtenidos por los distintos partidos en cada provincia son los siguientes:

Provincia	Partido ganador	Nº de representantes
Madrid	PSOE	52
Barcelona	PSOE	44
Valencia	PSOE	21
Sevilla	PSOE	16
Alicante	PSOE	16
Málaga	PSOE	14
Murcia	VOX	13
Cádiz	PSOE	11
Vizcaya	PNV	11
Baleares	PSOE	11
Las Palmas	PSOE	10
La Coruña	PP	10
S.C.Tenerife	PSOE	10
Asturias	PSOE	10
Zaragoza	PSOE	9
Pontevedra	PSOE	9
Granada	PSOE	9
Tarragona	ERC	8
Córdoba	PSOE	8
Gerona	ERC	8
Gipúzcoa	PNV	7
Almería	PSOE	7
Toledo	PSOE	7
Badajoz	PSOE	7
Navarra	PSOE	7
Jaén	PSOE	7
Cantabria	PP	6
Castellón	PSOE	6
Huelva	PSOE	6
Valladolid	PSOE	6
Ciudad Real	PSOE	6
León	PSOE	5
Lérida	ERC	5
Cáceres	PSOE	5
Albacete	PSOE	5
Burgos	PSOE	5
Álava	PNV	4
Salamanca	PP	4
Lugo	PP	4
La Rioja	PSOE	4
Orense	PP	4
Guadalajara	PSOE	4
Huesca	PSOE	4
Cuenca	PSOE	3

Zamora	PP	3
Palencia	PP	3
Ávila	PP	3
Segovia	PP	3
Teruel	TERUEL EXISTE	3
Soria	PSOE	3
Melilla	PSOE	3
Ceuta	PSOE	3

Sumando los representantes tendríamos:

- PSOE: 353
- PP: 40
- VOX:13
- PNV: 22
- ERC: 21
- TERUEL EXISTE: 3

En este caso, el ganador de las elecciones sería el PSOE, por mayoría absoluta.