Biotecnología Programación 2023-2024 Dificultad: alta
1º curso Grupo M10

# PRÁCTICA 6

## SISTEMA ELECTORAL D'HONDT

En esta práctica conoceremos el sistema D'Hondt. Este sistema fue ideado en 1878 por el jurista y profesor belga Victor D'Hondt, y en España se aplica desde 1977. El procedimiento a seguir es el siguiente:

Una vez celebradas las elecciones, se cuentan los votos obtenidos por cada partido y se ordenan de mayor a menor, descartando las candidaturas que hayan obtenido un porcentaje de votos inferior al 3%, de esta manera se evita que la Cámara se fragmente demasiado.

El primer escaño se otorga al partido que ha obtenido más votos. A continuación, se divide el número de votos entre dos, y el escaño se asigna al siguiente partido con más votos (excluyendo al primero). Para asignar el tercer escaño se divide el número de votos entre tres, y el escaño se otorga al partido con más votos. Se continuará con este proceso hasta que todos los escaños estén asignados.

En España este sistema se aplica a cada una de las 52 circunscripciones separadamente, de acuerdo con el número de escaños que tienen asignados.

## ENUNCIADO

En este ejercicio, determinaremos el reparto de escaños entre los partidos políticos como si toda España fuese una misma circunscripción. Utilizaremos los resultados de las elecciones del 23 de julio de 2023.
No aplicaremos el criterio del 3% (710.521 VOTOS).

| PARTIDO | VOTOS | ESCAÑOS |
|---|---|---|
| BNG | 152327 | 1 |
| PNV | 275782 | 5 |
| ERC | 462883 | 7 |
| PP | 8091840 | 137 |
| PSOE | 7760970 | 121 |
| EH-Bildu | 333362 | 6 |
| VOX | 3033744 | 33 |
| Sumar | 3014006 | 31 |
| JxCat | 392634 | 7 |
| UPN | 51764 | 1 |
| Cca | 114718 | 1 |
| **TOTAL** | 23684030 | 350 |

## DATOS

En primer lugar introduciremos los datos que nos proporciona el enunciado.
Para ello crearemos 3 vectores diferentes:
.- 1º con los nombres de los partidos, al que denominaremos **Partidos**.
.- 2º con los votos obtenidos por cada partido, al que denominaremos **votos.**
.- 3º con los escaños obtenidos por cada partido en la vida real, al que llamaremos **escanos_reales.**
A continuación introduciremos el número de partidos, podremos hacerlo simplemente colocando el nº total de partidos, en este caso 11, o bien indicándole al programa que ese dato es la longitud del vector **Partidos** ( npart=length(Partidos) ).
Introducimos el número de escaños totales, y por último el número de votos.

```
Partidos=c("BNG", "PNV", "ERC", "PP", "PSOE", "EHBildu", "Vox", "Sumar",
"JxCat", "UPN", "Cca")
votos=c(152327,275782,462883,8091840,7760970,333362,3033744,3014006,3926
34,51764,114718)
escaños_reales=c(1,5,7,137,121,6,33,31,7,1,1)
npart=lenght(Partidos) ; nesc=350
nvotos=lenght(votos)
```

**FUNCIONES**

En primer lugar vamos a programar una función que llamaremos **"asignación_escanos()"**, que dependerá del nº de votos, nº de partidos y del nº de escaños. El nombre de esta función no importa siempre y cuando luego la llamemos por el nombre que le hemos dado.

A continuación inicializamos los contadores que vamos a utilizar.
Crearemos una matriz A cuyo nº de filas será el nº de partidos, y el nº de columnas será el nº de escaños.

Crearemos un vector llamado **escanos_unica**, que contendrá el nº de escaños de cada partido, y nos servirá para ir añadiendo el nº de escaños que irá obteniendo cada partido. Inicializaremos tanto escanos_unica como cada componente a cero, utilizando un bucle ***for.***
Mediante un bucle ***while***, impondremos la condición de que mientras el contador de escaños sea menor al nº de escaños;

.- para cada elemento del vector votos[i] (i=1,…,npart), lo divida entre **k** lo almacene en la columna de la matriz.

.- para encontrar el mayor valor de la matriz y así repartir el escaño al partido correspondiente, llamaremos **max** al nº de votos que hay en la matriz en ese momento, **imax** a su fila y **jmax** a su columna.

A continuación al elemento con mayor nº de votos (imax, jmax), le asignamos valor cero, para evitar que se repita en el siguiente ciclo como valor máximo.
Para sumar el escaño al partido correspondiente; escanos_unica [imax]=escanos_unica [imax]+1. Esto es como apuntar en una lista cuantos escaños se ha llevado cada partido ciclo a ciclo. Por último sumaremos 1 al contador de columnas y a cuenta_escanos.

Este proceso se repetirá mientras se mantenga la condición dictada por ***while***, es decir hasta que cuenta_escanos sea igual al nº de escaños.

Finalmente empleamos el comando ***return()*** para que la función nos devuelva el vector con el nº de escaños que ha obtenido cada partido.

```
asignacion_escanos=function(votos,npart,nesc){
     k=1; cuenta_escan=1
     A=matrix(c(0),nrow=npart,ncol=nesc)
     escanos_unica=0
     for (i in 1:npart){
          escanos_unica[i]=0
     }
     while(cuenta_escan<=nesc){
          max=0
          for(i in 1:npart){
               A[i,k]=votos[i]/k
               for(j in 1:k){
                    if (A[i,j]>max){
                         max=A[i,j]
                         imax=i
```

```
                              jmax=j
                        }
                  }
            }
            A[imax,jmax]=0
            escanos_unica[imax]=escanos_unica[imax]+1
            k=k+1
            cuenta_escan=cuenta_escan+1
      }
 return(escanos_unica)
 }
```

## TABLA DE DATOS

Vamos a crear una tabla que compare el nº de escaños obtenidos aplicando el sistema suponiendo España como circunscripción única, frente a los escaños reales.
Para ello daremos al vector escanos_unica el valor obtenido en la función a partir de nuestros datos. Emplearemos el comando ***data.frame***, para crear una tabla con los nombres de los partidos y los escaños obtenidos según cada sistema. Mediante el comando ***T***, el programa nos devuelve la tabla.

```
escanos_unica=asignacion_escanos(votos,npart,nesc)
T=data.frame(Partido,escanos_unica,escanos_reales)
T
```

## ORDENACIÓN DE DATOS

Vamos a ordenar los datos de mayor a menor nº de votos obtenidos, para ello utilizaremos un "algoritmo burbuja". Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado.

Los datos que van a hacer posible intercambiar la información sin que se pierda nada (por tanto intercambiar las posiciones de dos partidos), se van a almacenar en la variable auxiliar ***ext***. Con ayuda de esta variable, iremos imponiendo las condiciones para que no solo se intercambien los datos en el vector de los votos, sino también en el de Partidos, escanos_reales y escanos_unica.

```
nvotos=length(votos)
      for(i in 1:nvotos){
            for(j in 1:(nvotos-1)){
                  if(votos[j]<votos[j+1]){
                        ext=votos[j]
                        votos[j]=votos[j+1]
                        votos[j+1]=ext
                        ext=Partido[j]
                        Partido[j]=Partido[j+1]
                        Partido[j]=ext
                        ext=escanos_reales[j]
                        escanos_reales[j]=escanos_reales[j+1]
                        escanos_reales[j+1]=ext
                        ext=escanos_unica[j]
                        escanos_unica[j]=escanos_unica[j+1]
                        escanos_unica[j+1]=ext
                  }
            }
}
O=data.frame(Partido,escanos_unica,escanos_reales)
O
```

Antes del ordenamiento están todos descolocados:

| | Partidos | escanos_unica | escanos_reales |
|---|---|---|---|
| 1 | BNG | 2 | 1 |
| 2 | PNV | 4 | 5 |
| 3 | ERC | 6 | 7 |
| 4 | PP | 121 | 137 |
| 5 | PSOE | 116 | 121 |
| 6 | EHBildu | 5 | 6 |
| 7 | Vox | 45 | 33 |
| 8 | Sumar | 45 | 31 |
| 9 | JxCat | 5 | 7 |
| 10 | UPN | 0 | 1 |
| 11 | Cca | 1 | 1 |

Tras el reordenamiento, los partidos con más votos se colocan arriba y luego bajan en orden descendente, pero cada partido conserva sus votos y sus escaños:

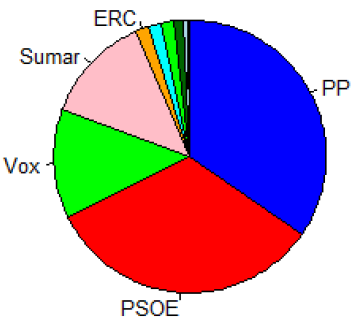
| | Partidos | escanos_unica | escanos_reales |
|---|---|---|---|
| 1 | PP | 121 | 137 |
| 2 | PSOE | 116 | 121 |
| 3 | Vox | 45 | 33 |
| 4 | Sumar | 45 | 31 |
| 5 | ERC | 6 | 7 |
| 6 | JxCat | 5 | 7 |
| 7 | EHBildu | 5 | 6 |
| 8 | PNV | 4 | 5 |
| 9 | BNG | 2 | 1 |
| 10 | Cca | 1 | 1 |
| 11 | UPN | 0 | 1 |

## GRÁFICOS

Vamos a crear dos gráficas una de sectores circulares y otra de barras superpuestas. Utilizaremos el comando ***par(mfrow)***, para que las dos gráficas se muestren juntas, este comando debe incluir un vector con nº de fila y nº de columnas que indique como se mostrarán las gráficas, también podemos usar la instrucción ***mar***, para poner los márgenes.

### SECTORES CIRCULARES

El gráfico mostrará los nombres de cada partido, identificado con un determinado color, así como el nº de escaños obtenidos. Debemos ser cuidadosos a la hora de asignar los colores al vector de color, tendremos que poner el color deseado para cada partido en la misma posición que dicho partido ocupe en su vector. Por tanto, las dimensiones de ambos vectores deben ser las mismas.




Programaremos nuestro gráfico con el vector Partidos, utilizaremos el comando ***pie***, ya que se trata de un gráfico circular (o “de tarta”), indicaremos los elementos que debe contener, que los muestre en sentido horario (***clockwise***), y por último el parámetro ***cex*** para el tamaño de la letra.

```
par(mfrow=c(2,1),mar=c(5,4,0.01,1)) names(escanos_unica)=c("PP", "PSOE", "Vox", "Sumar", "ERC") pie(escanos_unica, clockwise=TRUE, col=c("blue", "red", "pink", "orange", "cyan", "green", "dark green", "light blue", "yellow"), cex=0.8)
```

**BARRAS**

Realizaremos un gráfico en el que aparecerán los nombres de los partidos, los escaños reales y los datos del vector escanos_unica, de manera que podamos compararlos. Utilizaremos el comando ***rbind***, que creará una matriz escaños cuyas filas serán los vectores escaños reales, y escaños única. Utilizaremos ***barplot***, ya que queremos crear una gráfica de barras. El parámetro ***beside=TRUE***, hará que las barras aparezcan una junto a la otra.

```
names(escanos_reales)=c("PP", "PSOE", "Vox", "Sumar", "ERC", "EHBildu",
"JxCat", "PNV", "BNG", "Cca", "UPN")
escanos=rbind(escanos_reales,escanos_unica)
barplot(escanos,beside=TRUE,col=c(1, 10),las=2) , ,
legend(x='top',legend=rownames(escanos),col=1:11,fill=c(1,10))
```