


# Ley D'Hont

Para llevar a cabo este ejercicio se debe seguir al pie de la letra las instrucciones detalladas en Moodle.

Universidad Politécnica de Madrid Grado en Biotecnología - ETSIAAB

---

## Pseudo-código

**PRIMER PASO:**

1. PROGRAMAR: `asignacion_escaños<-function(votos,npart,nesc)` Y SALIDA: `escaños_unica` con los siguientes pasos:

- Inicializar los contadores: `k=1; cuenta_escan=1`
- Inicializar a 0 una matriz `A` de `npart` filas y `nesc` columnas.
- Inicializar el vector `escaños_unica = 0`. Además, hacer: `escaños_unica[i]=0, (i=1,...,npart)`
- Mientras (`cuenta_escan<nesc`) hacer:
  - Almacenar en la columna `k` de `A` la cantidad: `votos[i]/k (i=1,...,npart)`
  - Encontrar el mayor valor de la matriz `A` (filas `1:npart` ; columnas `1:k`) almacenando el valor en la variable `max`, su fila en `imax` y su columna en `jmax`.
  - Hacer `A(imax,jmax)=0` para no volverlo a utilizar.
  - Hacer `escaños_unica [imax]=escaños_unica [imax]+1; k=k+1; cuenta_escan=cuenta_escan+1`

Fin del bucle  
`return (escaños_unica)`  
Fin function

**DATOS:**

```
nesc=350; npart=17
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');
escaños_reales=c(120, 89, 52, 35, 10, 13, 8, 8, 6, 5, 2, 0, 2, 1, 2, 1, 1);
```

Fundamentos de Programación: Práctica 4 (A. Hidalgo, A. Fidalgo) 20

## 1. Función

Comenzamos programando una función a la que llamaremos **asignación\_escaños** (es recomendable llamarla de alguna manera más fácil de escribir). La función dependerá de las variables **votos**, **npart** (número de partidos), y **nesc** (número de escaños).

```
ae = function(votos,npart,nesc){
```

Primero igualamos las variables **k** y **cuenta\_escan** a 1 (De nuevo, recomendamos usar nombres más sencillos).

```
k=1; ce=1
```

Ahora inicializamos una matriz **A** de **npart** filas y **nesc** columnas en 0, utilizando el comando **matrix()** de la segunda práctica.

```
A = matrix(0, nrow = npart, ncol = nesc)
```

Una forma de hacer el siguiente paso, que consiste en generar un vector de **npart** elementos que valgan 0 es hacer:

```
eu=c(1:npart)*0
```

Así nos ahorramos tener que programar un bucle for. Aunque no es un método universal para todos los programas, trucos como este pueden ser encontrados fácilmente buscando en Google y empleando sitios como [StackOverflow](https://stackoverflow.com), así que te recomendamos que indagues siempre un poco más.

Ahora se aplica lo explicado en el contenido teórico de esta práctica, ya que el uso de la palabra **Mientras** nos delata que tenemos que crear un bucle **while**, que establece que ocurran una serie de procesos cuando **ce** es menor que **nesc**.

```
while (cuenta_escan<nesc){
```

Lo primero que programamos dentro de este bucle while es un bucle **for**, para poder introducir en la columna **k** de la matriz **A** el número de votos de cada partido, dividido por el número de la columna (así la **1ª** presentará los votos **totales**, la **2ª** la **mitad**, la **3ª** un **tercio** y así sucesivamente).

```
for (i in 1:npart){  
    A[i,k]=votos[i]/k  
}
```

Ahora se debe utilizar el proceso para calcular el **mayor** valor de una matriz, que fue visto en clase y se puso a prueba durante las prácticas de **Vectores y Matrices**. La forma en la que nosotros lo resolvimos fue primero creando **3** variables que contuvieran el **valor numérico** máximo (**max**) y su posición en **filas** (**imax**) y **columnas** (**jmax**). A estas variables se las asigna el valor de el elemento que se encuentra en la primera fila y la primera columna.

```
max=A[1,1]  
imax=1  
jmax=1
```

Y mediante dos bucle **for** y una estructura condicional **if** comparamos este valor asignado a **max** con todos los demás de la matriz. Si otro elemento de la matriz es mayor que el valor **max** ya asignado, **cambiamos** éste por el nuevo valor mayor.

```
for (i in 1:npart){
    for (j in 1:k){
        if (A[i,j]>max){
            max=A[i,j]
            imax=i
            jmax=j
        }
    }
}
```

Tras esto, hacemos que el valor de la matriz con coordenadas **imax** y **jmax** sea igual a **0**, para que cuando el programa **vuelva** a buscar el valor máximo de la matriz no elija el **mismo** otra vez.

```
A[imax,jmax]=0
```

Ahora sumamos 1 al valor de la componente **imax** del vector **escaños\_unica**. Esto representa que el número de escaños de un partido en concreto ha aumentado en 2. Después sumamos **1** a los valores de **k** y **cuenta\_escan**. Al primero se los sumamos para que actúe como **dividendo** de la siguiente columna de votos (como ya dijimos antes, la segunda columna se divide por **2**, la tercera por **3**, etc.). Al segundo le sumamos 1 porque es la **variable contador**, y el bucle **while** que hemos generado depende de que esta variable vaya **cambiando** su valor hasta llegar al **límite** del bucle.

```
escaños_unica[imax]=escaños_unica[imax]+1
k=k+1
cuenta_escan=cuenta_escan+1
```

Por último, empleamos el comando **return()** para que la función devuelva el valor de **escaños\_unica** calculado. (No olvides cerrar tanto el bucle **while** como la función con sus respectivas **llaves**).

```
} (<- cierre del while)
return(escaños_unica)
} (cierre de la función)
```



## Pseudo-código

### En el programa principal:

1. Introducir los DATOS.
  2. Llamar a la función: `escaños_unica<-asignacion_escaños(votos,npart,nesc)`
  3. Construir la tabla: `T = data.frame(Partido,escaños_unica,escaños_reales)`  
y Escribir: T
- POST-PROCESO:**
6. `par(mfrow=c(2,1),mar=c(5,4,0.01,1))` Para colocar 2 gráficos distribuidos en 2 filas y 1 columna.
  7. `names(escaños_unica) <- c('PSOE','PP','VOX','UP','Cs','ERC','+País')`
  8. `pie(escaños_unica, clockwise=TRUE,col=c('red','blue','green','purple','orange','yellow','pink','red','dark green','blue','brown'),cex=0.8)` *#PARA GRÁFICO SECTORES CIRCULARES*
  9. `names(escaños_reales) <- c('PSOE','PP','VOX','UP','Cs','ERC','+País','JxC','PNV','Bildu','CUP','PACMA','CC','BNG','N+','PRC','TERUEL')`
  10. `escanos<-rbind(escaños_reales,escaños_unica)`
  11. `barplot(escanos, beside=TRUE, col=c(4,5), las=2)` *#PARA GRÁFICO BARRAS SUPERPUESTAS*
  12. `legend(x='top', legend=rownames(escanos), col=1:17,fill = c(4,5))`

## 2. Programa

En primer lugar debemos introducir los datos que aparecen en la primera diapositiva del enunciado. Para mayor facilidad a la hora de copiar y pegarlos aquí se dan:

**nesc=350**

**npart=17**

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

**Partidos=c('PSOE','PP','VOX','PODEMOS','Cs','ERC','MasPais','JxCAT','PNV','BILDU','CUP','PACMA','CC','BNG','NAVARRA\_SUMA','PRC','TeruelExiste')**

**escaños\_reales=c(120, 89, 52, 35, 10, 13, 8, 8, 6, 5, 2, 0, 2, 1, 2, 1, 1)**

Después se programa la función paso a paso como se ha explicado en el paso 1.

```
asignacion_escaños = function(votos,npart,nesc){  
...  
}
```

Ahora llamamos a la función **asignación\_escaños** con los datos copiados de la **presentación**, e introducimos su valor en la variable **escaños\_unica**.

```
escaños_unica=asignacion_escaños(Votos,npart,nesc)
```

Por último creamos una tabla **T** empleando la función `data.frame()` en la que se muestre el nombre de los partidos (**Partidos**), sus escaños calculados con la función (**escaños\_unica**), y sus escaños reales (**escaños\_reales**), y decimos al programa que nos muestre la tabla.

```
T= data.frame(Partidos,escaños_unica,escaños_reales)
```

```
T
```

### 3. Representación Gráfica

Para esta última parte basta con copiar el programa que aparece en la presentación. Aquí se presenta para facilitar su copia:

```
par(mfrow=c(2,1),mar=c(5,4,0.01,1))
```

```
names(escaños_unica)=c('PSOE','PP','VOX','UP','Cs','ERC','+País')
```

```
pie(escaños_unica, clockwise=TRUE,col=c('red', 'blue', 'green', 'purple', 'orange',  
'yellow', 'pink', 'red', 'darkgreen', 'blue', 'brown'),cex=0.8)
```

```
names(escaños_reales)=c('PSOE','PP','VOX','UP','Cs','ERC','+País','JxC','PNV','Bildu',  
'CUP','PACMA','CC','BNG','N+','PRC','TERUEL')
```

```
escanos=rbind(escaños_reales,escaños_unica)
```

```
barplot(escanos, beside=TRUE, col=c(4,5), las=2)
```

```
legend(x='top', legend=rownames(escanos), col=1:17,fill = c(4,5))
```

### 4. Resultado

Debería quedarnos la siguiente **tabla** y **2 gráficos**:

	Partidos	escaños_unica	escaños_reales
1	PSOE	102	120
2	PP	75	89
3	VOX	55	52
4	PODEMOS	46	35
5	Cs	24	10
6	ERC	13	13
7	MasPais	8	8
8	JxCAT	7	8
9	PNV	5	6
10	BILDU	4	5
11	CUP	3	2
12	PACMA	3	0
13	CC	1	2
14	BNG	1	1
15	NAVARRA_SUMA	1	2
16	PRC	1	1
17	TeruelExiste	0	1

