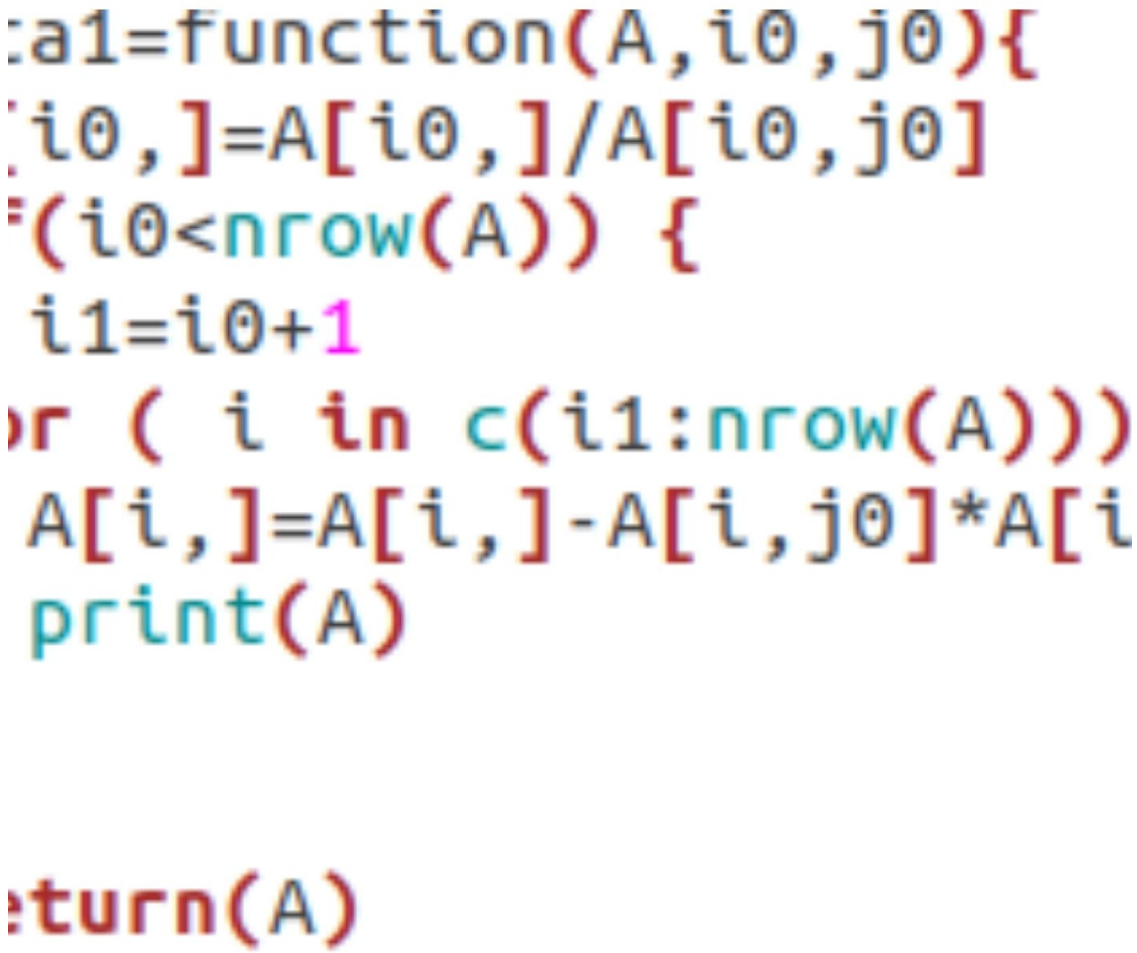
Informática y modelización



Fernando Blasco

Antonia González

POLITÉCNICA

Montes

Capítulo 1

Primeros pasos con R

1. El entorno de trabajo

El sistema R es un entorno y lenguaje de programación abierto, libre y gratuito. Funciona bajo licencia GNU (General Public License, en inglés). R se ideó para manejar datos estadísticos de una forma sencilla y efectiva y también posee capacidades gráficas.

La ventaja de aprender a usar R es que se pueden encontrar muchos programas ya implementados en este lenguaje. Esos programas se aplican en la gestión del medio natural y se utilizarán en asignaturas posteriores.

Si bien R no sería la elección que haría un programador, con él se pueden cubrir los objetivos mínimos que nos proponemos mostrar en esta introducción a la programación. Daremos las ideas básicas sobre programación y podremos implementar algunos modelos sencillos y, lo que puede ser más importante, entender qué hacen los programas que han escrito otras personas y que podemos encontrar por la red, modificándolos para nuestras necesidades.

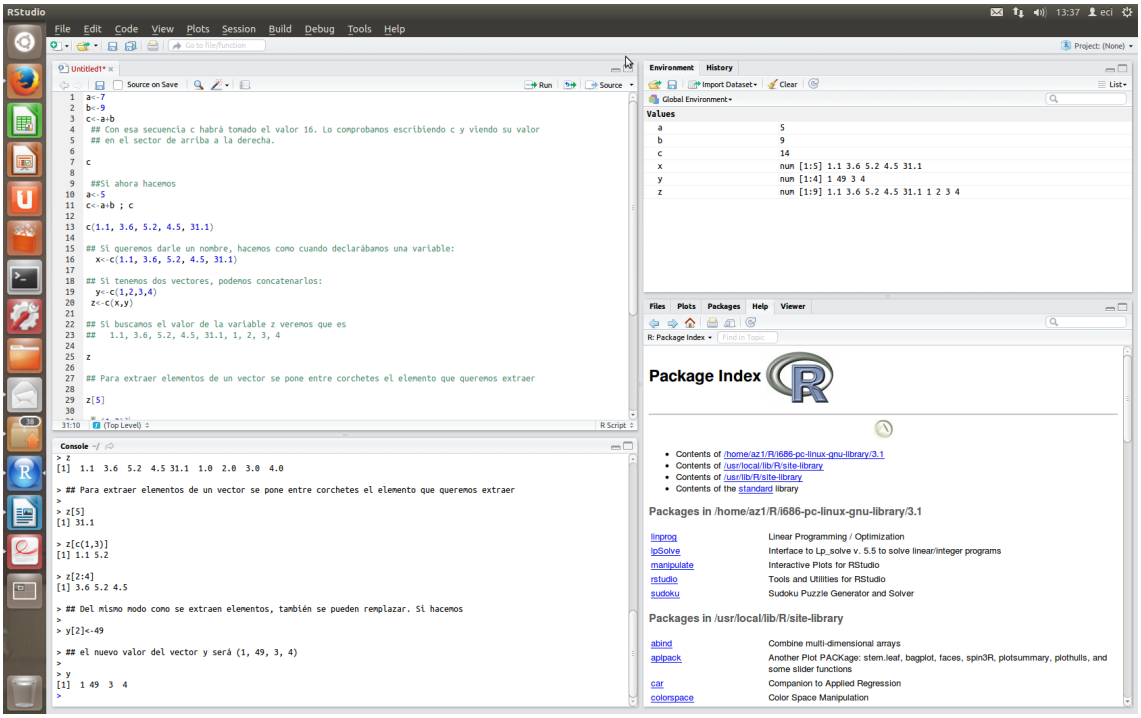
R es el motor que va a ejecutar nuestros programas. Podríamos usarlo a golpe de instrucciones desde una ventana de comandos, pero es mucho más cómodo utilizar un paquete que contenga integrado un editor de texto (con el que escribiremos los programas) y la posibilidad de compilarlos. Nosotros utilizaremos R-studio, cuya versión de escritorio es gratuita. Su editor es cómodo de usar y ofrece, además, un listado de las variables que se están utilizando, una ventana de gráficos y una configuración sencilla.

1.1. Instalación de R-Studio.

1. Instalar la última versión de R. Para windows se encuentra en este enlace. Para Linux o macOS se puede encontrar en este otro enlace.
2. Descargar e instalar R-studio de su página. Debe elegirse la versión correspondiente al sistema operativo que estemos utilizando.

Para abrir el programa hay que buscar el icono que lo representa y hacer click sobre él. 

La pantalla de R-Studio está dividida en 4 sectores. Al abrirlo por primera vez uno de ellos aparece minimizado. Hay que encoger la pestaña `Console` para que aparezca el editor de texto.



1.2. Uso de la consola. La **consola** es el espacio donde se ejecutan las órdenes, mientras que el **editor** es el espacio donde escribiremos los programas. Habitualmente los dos se encuentran en la columna de la izquierda, el editor encima de la consola.

Para ejecutar una orden se puede hacer de diferentes modos

- Desde la consola pulsando `Enter`
- Desde el editor pulsando `Control+Enter` o haciendo click en la pestaña `Run`

Se pueden utilizar operaciones, funciones matemáticas y constantes habituales:

```
+   *   -   /   ^   abs()   log()   sin()   exp()   sqrt()   asin()   tan()   pi
```

Ejercicio 1.1

a) Calcula el valor de las siguientes expresiones:

$345{,}87 + 451{,}336$, $231 * 24$, $231/24$, 231^3, $231^{(1/3)}$, $231^{0{,}5}$, $\sqrt{231}$

e, $\sin(\pi/4)$, $\log(e^2)$, $4^{(2/3)}$, $4^2/3$, $2/0$, $\tan(\pi/2)$

$\log(0)$, $\text{arcsen}(1)$, 10^{308}, 10^{309}

b) Ejecuta los siguiente comandos en R:

> 8 %%2

> 7 %%2

> 5 %/%3

> 17 %/%5

> 13 %%4

¿Podrías explicar lo que hace los operadores %% y %/% ?

En el caso en que no podamos acceder a un ordenador con R instalado podemos utilizarlo online en algunos servidores, como por ejemplo `https://www.mycompiler.io/es/new/r`.

R está especialmente pensado para trabajar con funciones estadísticas. No vamos a entrar en ello puesto que no es objeto de esta asignatura, pero sí destacaremos cómo generar *números aleatorios*.

`runif(n)` da como resultado un vector con `n` números entre 0 y 1 generados aleatoriamente con una distribución uniforme.

La función `sample` extrae una muestra de una colección dada. Hay que tener cuidado porque la muestra puede ser con reemplazamiento o sin reemplazamiento.

Por ejemplo,

`sample(1:6,1,replace=TRUE)` simularía el resultado del lanzamiento de un dado.

`sample(1:6,4,replace=TRUE)` simularía el resultado de lanzar un dado 4 veces.

Sin embargo, si lo que queremos es simular 10 números obtenidos en el bingo deberíamos usar la orden

```
sample(1:90,10,replace=FALSE)
```

puesto que las bolas que salen no vuelven a entrar hay que programar la muestra *sin remplazamiento*.

Ejercicio 1.2

a.- Ejecuta esta sentencia `sample(1:6,8,replace=FALSE)`. ¿Por qué da error?

b.- Escribe una orden que te proporcione el resultado de un sorteo de lotería primitiva (6 números de entre 49).

c.- Escribe una orden que proporcione aleatoriamente un punto dentro del cuadrado $[0,1] \times [0,1]$.

2. Variables.

R admite variables numéricas, lógicas y de texto.

Asignación de variables. Es muy cómodo dar nombre a números, textos y funciones para referirnos a ellos dentro de un programa. Por ejemplo, las siguientes líneas lo que hacen es llamar `a` al número 7, `b` al número 9 y `c` al resultado de sumar `a` y `b`.

La versión actual de R permite asignar valores a las variables directamente con el signo =. Se puede seguir usando el símbolo heredado de versiones anteriores " < −".

```
a<-7        a=7
b<-9        b=9
c<-a+b      c=a+b
```

Con esa secuencia `c` habrá tomado el valor 16. Lo comprobamos escribiendo `c` y viendo su valor en el sector de arriba a la derecha (`Environment`)

Si ahora hacemos

```
a<-5
c<-a+b
```

El valor de `c` habrá cambiado a 14.

Observa que el valor de las variables de texto se escribe entre comillas.

```
x <- "abcdef"
```

Si no recordamos qué tipo de variable tenemos, podemos usar el comando `class`.

```
class(x)
class(a)
```

3. Vectores

Un **vector** se define con la letra `c` y separando sus componentes con comas. Por ejemplo

```
 c(1.1, 3.6, 5.2, 4.5, 31.1)
c("sábado","domingo")
```

Si queremos darle un nombre, hacemos como cuando declarábamos una variable:

```
x<-c(1.1, 3.6, 5.2, 4.5, 31.1)
finsemana<-c("sábado","domingo")
```

Si tenemos dos vectores, podemos **concatenarlos**:

```
y<-c(1,2,3,4)
z=c(x,y)
```

Si buscamos el valor de la variable `z` veremos que es

```
 1.1, 3.6, 5.2, 4.5, 31.1, 1, 2, 3, 4
```

Si los componentes de un vector son números consecutivos podemos no utilizar la letra `c`. Así, por ejemplo, las tres expresiones siguientes, $v1, v2$ y $v3$ producen el mismo resultado: el vector $(3, 4, 5, 6, 7, 8)$

```
v1<- c(3,4,5,6,7,8)
v2<- c(3:8)
v3<- 3:8
v4<-  4:9-1
```

Cuidado con los prioridades de las operaciones al escribir las sentencias.

La función `seq()` de R se puede usar para generar una sucesión de números. Su sintaxis es la siguiente:

```
seq(from=**, to=**, by=**, length.out=NULL, along.with=NULL)
```

donde

`from` es el valor inicial de la secuencia, `to` es el valor final, `by` es el incremento, `length.out` la longitud de la secuencia `along.with` la longitud deseada que concuerda con la de otro objeto

```
a1=seq(20)
a2=seq(-3,to=20)
a3=seq(-3, to=20, by=3)
a4=seq(-3, length.out=20)
b1=1:4
a5=seq(-3, by=3, along.with=b1)
```

La función `rep()` de R se puede usar para generar una sucesión de un número repetido. Su sintaxis es la siguiente:

```
rep(x, r, each=**)
```

```
r1=rep(2, 3)
r2=rep(c(1:3),2)
r3=rep(c(1:3), each=3)
r4=rep(seq(-3, to=3, by=3), each=2)
```

Para **extraer** elementos de un vector se pone entre corchetes la posición que ocupa el elemento que queremos extraer Así, volviendo al vector `z` con el que habíamos trabajado antes,

```
z[5]     31.1
z[c(1,3)]   [1.1 , 5.2]
z[c(1:3)]   [1.1 , 3,6 , 5.2]
z[c(2:5)]   [3.6, 5.2, 4.5, 31.1]
z[c(2,5)]   [3.6 , 31.1]
z[2:5]   [3.6, 5.2, 4.5, 31.1]
```

Esta orden presenta mucha versatilidad, lo veremos en próximas prácticas.

Del mismo modo como se extraen elementos, también se pueden **reemplazar**

Si hacemos `y[2]<-49` el nuevo valor del vector `y` será `(1, 49, 3, 4)`

```
v<-seq(1,10,0.1)
v[-90]
v1<-seq(2,length(v),2)
v2<-seq(2,by=2,along.with=v)
w1<-v[-v1]
w1<-v[-v2]
```

```
w1  #explicar qué ocurre
```

Ejercicio 1.3

a) Genera un vector `v` cuyas primeras entradas sean los números pares mayores que dos y menores que 14.

b) Genera el vector `w` cuyas entradas sean los múltiplos de 3 del 1 al 12.

c) Genera el vector `vw` cuyas entradas son el vector `v` y el vector `w`.

d) A partir del vector `vw`, genera los siguientes vectores

d1) Un vector, a partir de `vw`, `vw1` que en la quinta entrada tenga un cero.

d2) Un vector `vw2` que no contenga el elemento que en `vw` ocupa la posición octava.

d3) Un vector `vw3` cuyas entradas 6, 7 y 8 sean -1.

d4) Un vector `vw4` cuyas entradas 1, 4 y 7, sean -1

4. Matrices

Una **matriz** se introduce con el vector donde están las entradas de la matriz, seguido del número de filas y el número de columnas. Por defecto, la matriz se escribe por columnas

```
A <- matrix(c(1,2,3,4,5,6), nrow=2, ncol=3)
```

Produce la matriz

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

Si queremos introducir la matrix por filas hay que indicarlo con la orden `byrow=TRUE`

```
B <- matrix(c(1,2,3,4,5,6), nrow=2, ncol=3, byrow=TRUE)
```

La matriz que produce es

$$B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Podemos **extraer elementos** de una matriz como de un vector, utilizando dos índices.

`B[2,2]` da como resultado `5`.

`A[1,2]` da como resultado `3`.

Si queremos **extraer una fila entera** de una matriz usamos solo el índice de fila:

`A[1,]` da como resultado `[1, 3, 5]`.

Si queremos **extraer una columna entera** de una matriz usamos solo el índice de columna:

`A[,2]` da como resultado `[3, 4]`.

Se pueden "pegar" matrices una encima de la otra

`M<-rbind(A,B)` (r proviene de *fila, row*)

O pueden pegarse una a continuación de la otra

`D<-cbind(A,B)` (c proviene de *columna*)

Podemos **extraer submatrices** de diferentes maneras:

```
M1<-M[2:3,2:3]
M2<-M[c(1,3),c(2,4)]
```

Hay formas alternativas, como eliminar filas o columnas con el signo −

```
M3<-M[-c(1,4),-1]
```

La matriz `M3` es la misma que `M1`, aunque expresada de formas distintas.

Para **redefinir una entrada** de una matriz simplemente se asigna el nuevo valor a esa entrada.

```
A[1,1]<-100
```

Podemos comprobar que efectivamente está bien redefinido mirando el valor de `A` en la pestaña `Environment` o pidiendo su valor por consola, escribiendo simplemente `A` y pulsando `Enter`

También podemos **redefinir filas o columnas** de golpe. Por ejemplo, la nueva primera fila de la matriz `B` es la antigua tercera fila de la matriz `A`

```
B[1,]<-A[3,]
```

Para realizar **operaciones elementales** en una matriz el lenguaje es muy natural:

`A[,2]<-3*A[,2]` multiplica por 3 la segunda columna de `A` (ojo, el valor de `A` cambia)

`A[,2]<-A[,2]-5*A[,1]` la segunda columna de `A` es la antigua segunda columna de `A` menos 5 veces la primera.

`A<-A[,c(2,1,3)]` permuta las columnas 1 y 2 en la matriz `A`

La **matriz identidad** de tamaño $n \times n$ se escribe `diag(1,n)`

Podemos también calcular **determinantes**, **multiplicar** matrices, calcular matrices **inversas** y matrices **traspuestas**.

`det(A[1:2,1:2])`

`A[1:2,1:2] %*% B` (esta es la multiplicación usual de matrices)

`solve(A[1:2,1:2])` (calcula la matriz inversa de esa submatriz 2×2 de `A`)

`A1<-t(A)` (hace la traspuesta de `A` y lo guarda en `A1`)

Es importante que cuando tenemos dos vectores o matrices *de la misma longitud,* se pueden hacer operaciones con las componentes de uno de ellos y la correspondiente del otro y lo almacena en un nuevo vector o matriz.

```
A <- matrix(1:6, nrow=2, ncol=3)
B <- matrix(1:6, nrow=2, ncol=3, byrow=TRUE)
A/B #divide elementos de A entre elementos de B
A*B #multiplica elementos de A por elementos de B
A%/%B #hace división entera de elementos de A entre elementos de B
A ^B #eleva cada elemento de A al correspondiente de B
```

Ya nos hemos referido a algunas *funciones* que vienen incorporadas en R. Podemos aplicar funciones no solo a números sino, de golpe, a todos los elementos de un vector o una matriz.

Si aplicamos una función a un vector, el resultado es el de aplicar la función *a cada uno de los elementos del vector.* Por ejemplo si ejecutamos las dos órdenes siguientes

```
a<- 1:5
b<- a^3
```

el resultado es que el vector `b` contiene los cubos de los números del 1 al 5. En efecto, se puede comprobar que `b = (1, 8, 27, 64, 125)`

Lo mismo ocurre al aplicar una función a una matriz.

Ejercicio 1.4

a) Define la matriz B, de tamaño 3×3 en la que, por filas, están los números del 1 al 9.

b) Multiplica la matriz B por sí misma, utilizando producto de matrices. Guarda ese resultado en una matriz C.

c) Ejecuta en R la orden `D<- B^2`

d) Compara las matrices C y D. Explica qué ocurre.

Ejercicio 1.5

Escribe un programa en R para crear una matriz con 3 filas y 5 columnas en la que aparezcan correlativamente los números pares mayores que 50.

$$\begin{bmatrix} 52 & 58 & 64 & 70 & 76 \\ 54 & 60 & 66 & 72 & 78 \\ 56 & 62 & 68 & 74 & 80 \end{bmatrix}$$

5. Funciones

5.1. Funciones numéricas. En R será muy útil definir funciones más generales. Iremos completando esto más adelante, cuando nos adentremos en la *programación*.

Cuando un proceso va a tener que ser utilizado varias veces, conviene definir una **función**. Las funciones pueden depender de varios argumentos, los cuales pueden ser tanto numéricos como vectoriales, matriciales, de texto o lógicos. Por ejemplo,

```
p<- function(x){
       3*x^2-5*x+10
}
```

es una función que nos permite, entre otras cosas, evaluar el polinomio $3x^2 - 5x + 10$ y mostrar el valor en pantalla. Para hallar los valores del polinomio en 3, −4 y 2'5 debemos ejecutar

```
p(3)     p(-4)     p(2.5)
```

Si lo que queremos es, por ejemplo, hacer una tabla de valores de ese polinomio y que nos muestre los valores de $p(1)$, $p(2)$, ..., $p(10)$, una vez definida la función `p` podemos usar la orden `p(1:10)`.

Para hallar los valores de `p` entre 0 y 1, con diferencia de una décima, podríamos usar una secuencia: `seq(0,1,by=0.1)`.

Seguidamente ejecuta las siguientes órdenes y observa la tabla de valores que hemos construido:

```
coordX<- seq(0,1,by=0.1)
coordY<- p(coordX)
tabla<- cbind(coordX,coordY)
```

Eso nos da una tabla de coordenadas. Si quisiésemos dibujar la gráfica correspondiente a esa tabla, simplemente hay que usar

`plot(tabla)`

¿Qué harías para que la gráfica tenga mejor resolución?

En realidad, para que la gráfica de una función tenga más resolución puede añadirse en la orden `plot` la instrucción de estilo `type="l"`. También podríamos cambiar el color añadiendo `col="green"`.

Se pueden encontrar más usos de la función `plot` en `https://www.datamentor.io/r-programming/plot-function/`

5.2. Funciones más generales. Las funciones pueden tomar cualquier tipo de argumento. Hemos visto que cuando las funciones numéricas se aplican a un vector el resultado es la función aplicada a cada uno de los elementos del vector, pero también podríamos definir funciones cuyos argumentos son vectores o matrices.

R trae predifinidas algunas funciones que actúan sobre vectores o matrices, como

`sum` que calcula la suma de todos los elementos de un vector o matriz

`prod` que calcula el producto de todos los elementos de un vector o matriz

Por ejemplo,

`sum(1:4)` da como resultado 10

`prod(1:4)` da como resultado 4! = 24

Podemos, por ejemplo, aprovechar esto para definir una nueva función que nos de la suma de los cuadrados de los elementos de un vector.

```
sumacuadrados = function(v){
sum(v^2)
}
```

Lo bueno que tiene utilizar funciones es que, una vez definidas, podemos utilizarlas siempre que queramos.

Partiendo de una función simple como

```
  producto=function(x,y){
  x*y
}
```

Podemos multiplicar elemento a elemento las coordenadas de dos vectores de la misma longitud. Por ejemplo, si

```
u=1:4
v=c(2,1,3,5)
```

al ejecutar `producto(u,v)`

nos va a dar como resultado

```
 2  2  9 20
```

EJEMPLO 1.1. Supongamos que tenemos árboles distribuidos en 4 clases de edad: la clase 1 comprende árboles de 0 a 5 años, la clase 2 de 5 a 10, la clase 3 de 10 a 15 y la clase 4 los árboles con más de 15 años.

Imaginemos que tenemos 14 árboles de clase 1, 23 de clase 2, 18 de clase 3 y 31 de clase 4.

Podemos representar nuestro "bosque" como

```
bosque=c(14,23,18,31)
```

Además, sabemos que cada árbol de la clase 1 se valora en 30€, cada unoi de la clase 2 en 80€, los de la clase 3 en 150€ y los de la clase 4 en 250€.

Podemos almacenar estos precios en otro vector

```
precio=c(30,80,150,250)
```

Utilizando la función `producto` que hemos definido antes, tenemos que el valor de nuestro bosque es

```
sum(producto(bosque,precio))
```

Esta misma idea podemos aprovecharla para hacer operaciones sobre matrices. Por ejemplo, las *operaciones elementales* que se utilizan en álgebra lineal se pueden agilizar enormemente si usamos funciones. Ejemplo:

Pretendemos que mediante la orden `mult(A,i,r)` se multiplique la fila `i` de la matriz `A` por el número `r`,

```
mult<- function(A,i,r) {
    A[i,]<- r*A[i,]
    return(A)
}
```

Observa que hemos puesto al final de la función `return(A)`. Eso es para que nos *devuelva* el valor de la función. Si solo hubiésemos puesto `A` lo veríamos por pantalla,

pero no quedaría asociado a la función. Este es un primer contacto con esa idea, que al programar en muchas ocasiones es necesaria. Lo veremos más adelante.

Ejercicio 1.6

a) Escribe la matriz A y analiza su estructura.

$$\begin{pmatrix} 1 & 2 & 3 & 2 & 4 & 6 \\ 4 & 5 & 6 & 8 & 10 & 12 \\ 3 & 9 & 15 & 1 & 2 & 3 \\ 6 & 12 & 18 & 5 & 7 & 9 \end{pmatrix}$$

b) Escribe una función tal que, al introducir un vector $\vec{v}$ con 6 entradas, escriba la siguiente matriz por bloques:

$$C = \left[\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right]$$

donde

- A_1 es la matriz 2×3 cuyas entradas son las de $\vec{v}$ escritas por filas
- A_2 es el doble de A_1
- A_3 se hace a partir de $3\vec{v}$, escribiendo el vector por columnas
- A_4 tiene la primera fila igual que A_1 pero su segunda fila es la suma de las dos filas de A_1.

Ejercicio 1.7

a) Diseña la función `sumar(A,i,j,r)` de modo que sume a la fila `i` de la matriz `A` la fila `j` multiplicada por `r`.

b) Construye la función `cambiar(A,i,j)` que permute las filas `i` y `j` de la matriz `A`. *Cuidado: en este apartado va a ser necesario que al definir la función "guardemos" una de las filas en otra variable.*

c) Finalmente, comprueba que, efectivamente, esas funciones hacen lo que se les pedía.

d) Resuelve el sistema de ecuaciones

$$\begin{aligned} x &+ y &&= 1 \\ x &- y &+ z &= -1 \\ x &&- z &= 1 \end{aligned}$$

OBSERVACIÓN 1.1. La función `det` que calcula el determinante de una matriz es una función, predefinida en R, que tiene como argumento una matriz. Da como resultado un número (o error, si la matriz no es cuadrada).

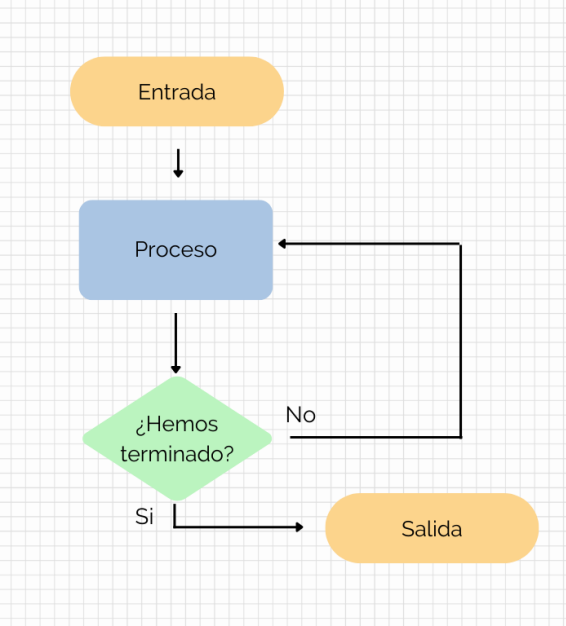
La función `inv` calcula la inversa de una matriz es una función, predefinida en R, que tiene como argumento una matriz. Da como resultado otra matriz del mismo tamaño (o error, si el argumento es una matriz no invertible).

Capítulo 2

Bucles

1. Procesos iterativos: bucles

Los **bucles** van a aparecer cada vez que hagamos un programa. El objeto de un bucle es realizar, repetidas veces, una serie de operaciones o instrucciones concretas. R nos da la posibilidad de utilizar instrucciones simplificadas para lo que en otros lenguajes de programación se requiere un bucle. Comenzamos con ejemplos parecidos a otros que ya hemos realizado, pero en esta vez pensados como un bucle.



EJEMPLO 2.1. Calcula los cubos de los 5 primeros números naturales.

```
for (i in 1:5){      #el bucle se repetirá 5 veces
   print( i^3)
    }
```

EJEMPLO 2.2. Calcula la suma de los 100 primeros números naturales

```
sumaparcial=0
for (i in 1:100){     #el bucle se repetirá 100 veces
   sumaparcial=sumaparcial+i
    }
print(sumaparcial)
```

Observa que sale por la consola el valor último de la suma. Si dentro del bucle hubiésemos escrito `print(sumaparcial)` habrían salido por pantalla todas las sumas parciales que se van realizando.

Ejercicio 2.1

En este ejercicio vas a programar una función que hace lo mismo que las órdenes `sum` y `prod` ya conocidas, por tanto no puedes usar esas instrucciones para realizar el ejercicio.

a) Define una función que aplicada a un vector $\vec{v}$ nos proporcione la suma de todas sus componentes.

b) Define una función que aplicada a un vector $\vec{v}$ nos proporcione el producto de todas sus componentes.

Ejercicio 2.2

Dado un número n define un bucle que calcule su factorial, $n!$

Ejercicio 2.3

Define una función que te permita calcular la potencia n de una matriz A.

EJEMPLO 2.3. Escribe los 50 primeros términos de la sucesión $\{1/n\}$.

```
for (i in 1:50){
print(1/i)
}
```

Para trabajar con sucesiones definidas por recurrencia (se define un término en función de los anteriores) es muy cómodo utilizar bucles.

EJEMPLO 2.4. La sucesión definida recurrentemente como

$$x_{n+1} = \frac{1}{2}\left(\frac{N}{x_n} + x_n\right)$$

es muy útil porque converge a $\sqrt{N}$. Calcula sus 10 primeros términos.

```
N=8 #número del que vamos a calcular la raíz cuadrada
x=6 #número por el que empezamos a calcular
for (i in 1:10){
x=0.5*(N/x+x)
print(x) #ojo: aquí en cada paso cambia el valor de x
}
```

EJEMPLO 2.5. Sucesión de Fibonacci. La sucesión de Fibonacci (que seguro que la habéis encontrado ya) es

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

Cada término es la suma de los dos anteriores. Esta serie aparece en la naturaleza en diferentes configuraciones biológicas, como por ejemplo en la disposición de las ramas en los árboles, las hojas en los tallos y las hojas de ciertas flores.

Vamos a calcular 15 términos de esta sucesión.

```
x1=1
x2=1
for (i in 1:15){
  x1_nuevo=x2  #tenemos que hacer esto para no machacar x1
  x2_nuevo=x1+x2
  x1=x1_nuevo
  x2=x2_nuevo
  print(x2)
}
```

Aunque habíamos introducido ya las funciones, por ejemplo, para dibujar gráficas, podemos usarlas para pasar argumentos en un procedimiento.

EJEMPLO 2.6. Calcula el término n de la sucesión de Fibonacci.

```
fibonacci=function(n){
x1=1
x2=1
for (i in 1:n-1){
  x1_nuevo=x2  #tenemos que hacer esto para no machacar x1
  x2_nuevo=x1+x2
  x1=x1_nuevo
  x2=x2_nuevo
      }
      return(x1)
}
```

Podemos también incluir bucles relacionados con matrices. Vamos a poner como ejemplo un bucle que pivota una matriz bajo el elemento (1,1). Pivotar implica convertir esa entrada en 1 y las que están por debajo en 0, haciendo operaciones elementales en matrices. Pivotar sirve, por ejemplo, para resolver sistemas de ecuaciones por el método de Gauss. De momento no tenemos mecanismos para *decidir* si el sistema se puede resolver o no. Eso lo veremos en la siguiente lección.

EJEMPLO 2.7. Dada la matriz $A = \begin{pmatrix} 2 & 1 & 5 \\ 4 & 5 & 3 \\ 1 & 6 & 4 \end{pmatrix}$ pivótala bajo el elemento (1,1). (En este caso puede hacerse porque ese elemento no es nulo)

```
A=matrix(c(2,4,1,1,5,6,5,3,4), nrow=3)
A[1,]=A[1,]/A[1,1]
A[2,]=A[2,]-A[2,1]*A[1,]
A[3,]=A[3,]-A[3,1]*A[1,]
  print(A)
```

Eso mismo hecho con un bucle sería

```
A=matrix(c(2,4,1,1,5,6,5,3,4), nrow=3)
A[1,]=A[1,]/A[1,1]
for (i in 2:3){
```

```
  A[i,]=A[i,]-A[i,1]*A[1,]
    }
    print(A)
```

Y esto mismo me sirve para cualquier tamaño de matriz

```
A=matrix( sample.int(20,60,replace=TRUE),nrow=10)
#En la línea anterior generamos una matriz aleatoria.
#Explicaremos más adelante cómo se hace

A[1,]=A[1,]/A[1,1]
  for (i in 2:nrow(A)){  #nrow nos indica el número de filas
  A[i,]=A[i,]-A[i,1]*A[1,]
    }
    print(A)
```

Ejercicio 2.4

Consideremos la matriz $A = \begin{pmatrix} 1 & -2 & 4 & -1 \\ 1 & -1 & 1 & 2 \\ 1 & 1 & 1 & 2 \\ 3 & -2 & 4 & 3 \end{pmatrix}$

a) Haz un bucle que pivote la matriz A en el elemento (1,1). Llama a esa nueva matriz $A1$.

b) Haz un bucle que pivote $A1$ en (2,2). Llama a la matriz resultante $A2$.

c) Pivota la matriz $A2$ en (3,3). Llama a la matriz resultante $A3$

d) Escribe un bucle que haciendo operaciones elementales en la matriz $A3$, queden ceros encima del elemento (4,4).

OBSERVACIÓN 2.1. El proceso anterior se puede automatizar de manera muy simple para matrices cuadradas que no planteen problemas. Implica usar un bucle dentro de otro y se profundizará en ello más adelante.

```
for(j in 1:ncol(A)){
   A[j,]=A[j,]/A[j,j]
       for (i in (j+1):nrow(A)){
```

```
    A[i,]=A[i,]-A[i,j]*A[j,]
}
print(A)
}
```

Ese procedimiento plantea problemas cuando el elemento $A(j,j) = 0$ y, para hacer correctamente el programa, habría que incluir una orden que detecte si ese elemento es 0 y, en ese caso cambiar de orden las filas.

Ejercicio 2.5

Una persona ingresa en un plan de pensiones 1000 euros el 1 de enero de cada año.

A este dinero se añade un interés anual del 2 % sobre la cantidad que en ese momento tiene ahorrada. El abono de intereses se hace el 31 de diciembre. ¿Qué cantidad tendrá en el plan de pensiones al final del 30^o año?

2. Bucles con while

Una instrucción que nos permite hacer bucles de otra forma es `while`, que permite que se ejecute un bucle hasta que se dé una determinada condición. En vez de ejecutar el bucle una serie de veces, este sigue ejecutándose hasta que se cumpla una determinada condición.

Por ejemplo, si queremos conocer todos los números naturales cuyos cubos son menores que 10000 podemos hacerlo de este modo

```
n=1
while(n^3<10000) {
    print(n)
    n=n+1
}
```

Es bien conocido que si $|x| < 1$ entonces $\lim_{n\to\infty} x^n = 0$. Podemos *corroborarlo* con este otro ejemplo con *while*:

```
x= 0.7
h= 10^(-5)
p=1
```

```
while(abs(p)>h) {
    p<- p*x
}
p
```

Mejoremos lo anterior escribiéndolo como función de modo que, además, nos informe de cuántas veces ha multiplicado el número x por sí mismo:

```
poten= function(x,h) {
    n<- 1
    p=1
    while(abs(p)>h) {
        p<- p*x
        n<-n+1
    }
    c(p,n)
}
```

Ejercicio 2.6

Suma los primeros 1000 términos de la sucesión $a_n = \frac{1}{n}$. Encuentra el número de términos de la sucesión que hay que sumar para que dicha suma sea al menos 5.

Ejercicio 2.7

Usa la sentencia *while* para averiguar a partir de qué posición los términos de la sucesión de Fibonacci son mayores que 1000.

Ejercicio 2.8

Escribe un programa que simule el lanzamiento de un dado y cuente cuántas veces hay que lanzarlo hasta que salga un 6.

Capítulo 3

Condicionales

1. Decisiones lógicas: condicionales

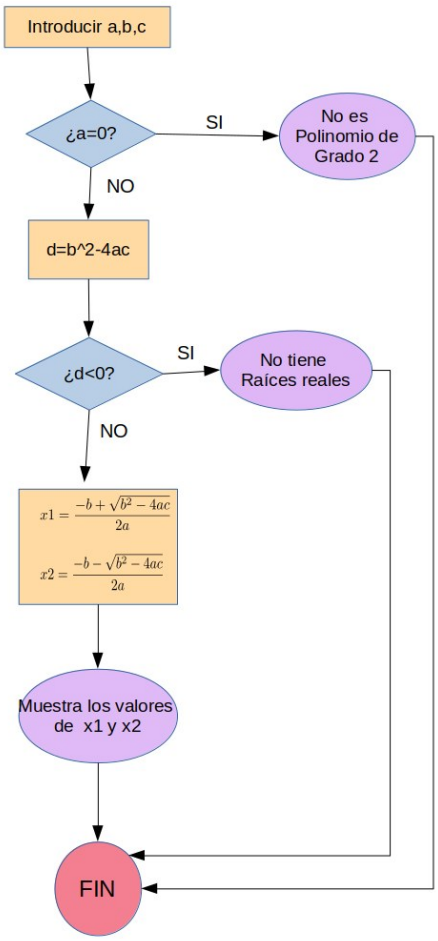
Es interesante combinar un bucle con una *decisión lógica*. Se utiliza cuando se quiere comprobar si se verifica una determinada condición, bien para que el programa se detenga o bien para que continúe por un camino diferente.

En este ejemplo calcularemos las raíces de un polinomio $ax^2 + bx + c$. Cuando el discriminante es negativo sabemos que no hay solución:

```
raices<-function(a,b,c){
  d<-b^2-4*a*c
    if (d<0) {print("la ecuación no tiene solución real")}
    else {print((-b+sqrt(b^2-4*a*c))/(2*a));
    print((-b-sqrt(b^2-4*a*c))/(2*a))}
}
```

Podríamos haber mejorado el programa para que detectara si, por ejemplo, es $a = 0$ y en realidad no se trata de un polinomio de segundo grado. Esto también se hace con instrucciones `if` o `else if`.

```
raices2<-function(a,b,c){
  if (a==0) print{"no es un polinomio de grado 2"}
    else {
        d<-b^2-4*a*c;
        if (d<0) {print("la ecuación no tiene solución real")};
        else {print((-b+sqrt(b^2-4*a*c))/(2*a));
          print((-b-sqrt(b^2-4*a*c))/(2*a))
        }
    }
}
```



En las cláusulas condicionales utilizaremos operadores lógicos. Ésta es su sintaxis:

|	O
&	Y
!	NO
==	identidad
<	menor que
>	mayor que

Es frecuente que en un condicional aparezcan más de dos opciones. Analiza este ejemplo:

```
ejem1<-function(z){
    if ( z<0 & z!=-2 ) { print(z^2 + 3) }
```

```
      else if ( z<=3 | z==10) { print(-4*z) ; print(c(z,2*z,4,z^3)) }
      else if ( z^2<50 ) {  print(matrix(c(z,0,0,-z),2)) }
    else { sqrt(z) }
}
```

Conviene resaltar que solo se ejecutará una acción: la que acompaña a la primera condición que se satisfaga. Apreciemos la importancia del orden en el que escribimos las condiciones:

```
ejem2<-function(z){
    if ( z<0 & z!=-2 ) { print(z^2 + 3) }
      else if ( z^2<50 ) { print(matrix(c(z,0,0,-z),2)) }
      else if ( z<=3 | z==10 ) { print(-4*z) ; print(c(z,2*z,4,z^3)) }
    else { sqrt(z) }
}
```

Ejercicio 3.1

La orden `sample(1:6,1)` nos proporciona una simulación del lanzamiento de un dado. Haz un programa que efectúe 20 lanzamientos del dado, imprima el número que ha salido y diga si ese número es par o es impar.

Ejercicio 3.2

Se pueden generar n números aleatorios entre a y b con la orden `runif(n,a,b)`. En este ejercicio se pide que generes 1000 puntos aleatorios en el cuadrado $[-1,1] \times [-1,1]$. Debes pintar el punto de rojo si está dentro de la circunferencia $x^2 + y^2 = 1$ y de azul si está fuera. *Indicación: primero hazlo con un punto y un condicional, después usa un bucle para que te repita eso 1000 veces*

Ejercicio 3.3

Si ejecutamos los siguientes programas en R, ¿qué número se escribirá en la consola?

a)

```
condicion = 1
i <- 3
while(condicion<10){
  if (i > 2) {
    condicion = i
    i <- i + 1
  }
} # end while
i
```

b)

```
condicion = 12
i <- 3
while(condicion<10){
  if (i > 2) {
    condicion = i
    i <- i + 1
  }
} # end while
i
```

c)

```
condicion = 1
i <- 1
while(condicion<10){
  if (i > 2) {
    condicion = i
    i <- i + 1
  }
} # end while
i
```

Ejercicio 3.4

Definimos una función en R como

```
fun <- function(a, b){
    res <- a
    if (b < a) {
       res <- b
    }
    return(res)
}
```

¿Qué aparece en pantalla tras ejecutar `fun(4,15)`?

Ejercicio 3.5

Imagina que ejecutas en R el siguiente código:

```
v=c(2:5)
A=matrix(v,2)
B=A%*%A
print(B[,2])
```

¿Cuál es el resultado?

Ejercicio 3.6

¿Qué hace esta función? ¿Qué aparece en pantalla cuando ejecutamos `fun2(9,8,7)` ?

```
fun2=function(a,b,c){
if (a > b) {
  if (b > c) {
  print(c)
  }
 else {
   print(b)
 }
}
else if (a > c) {
  print(c)
}
 else{
  print(a)
 }
}
```

Ejercicio 3.7

El siguiente sistema modeliza unas ecuaciones predador-presa discretas:

$$\begin{cases} x_{n+1} = & (a+1)x_n - bx_n y_n \\ y_{n+1} = & (1-c)y_n + dx_n y_n \end{cases}$$

donde $a, b, c, d > 0$ y x_n e y_n representan las densidades de población de la presa y el predador, respectivamente, en el intervalo de tiempo n.

En el caso en que las cuentas del modelo den una población negativa debe sustituirse por 0.

Haz un programa que analice la evolución del sistema para a=1.3, b=0.002, c=1.1, d=0.005. Partiendo de una población inicial $x = 100,\ y = 30$.

Estudia qué ocurre con la población para otros valores de los parámetros y para una situación inicial diferente.

EJEMPLO 3.1. En el capítulo anterior hemos estudiado un bucle que pivotaba debajo de un elemento. Pero si ese elemento era 0 daba error. Podemos mejorar ese procedimiento haciendo que busque una fila en la que el elemento correspondiente es distinto de 0 y poniéndola en su lugar:

```
pivota1=function(A,i0,j0){
  A[i0,]=A[i0,]/A[i0,j0]
  if(i0<nrow(A)) {
    i1=i0+1
  for ( i in c(i1:nrow(A))){
    A[i,]=A[i,]-A[i,j0]*A[i0,]
    print(A)
  }
  }
  return(A)
}
```

Ahí puede darnos errores si hay algún valor 0 y pedimos dividir por ese valor. Con instrucciones `if` y `while` podemos evitar este error.

```
cambia=function(A,i,j){
  guardado=A[i,]
  A[i,]=A[j,]
  A[j,]=guardado
  return(A)
  }
```

```
pivota2=function(A,i0,j0){
  i=i0
    while(A[i,j0]==0){
      if(i<nrow(A)){
        i=i+1
      }
      else {
        return(A)
        #print(A)
        break
      }
    }
  A=cambia(A,i0,i)
 A=pivota1(A,i0,j0)
  return(A)
}
```

Con eso, podemos fácilmente hacer una función que nos proporcione la forma escalonada de una matriz:

```
pivota3=function(A){
  for(i in (1:nrow(A))){
   A= pivota2(A,i,i)
  }
  return(A)
}
```

Ejercicio 3.8

Explica con tus palabras en lo que cambia la función `pivota2` con respecto a `pivota1`.

Ejercicio 3.9

Haz una función (dependiente de una matriz A) que vea si la matriz es cuadrada o no. En caso de ser cuadrada, que mire si es invertible y, en caso de que lo sea, que proporcione A^{-1}.

2. Bucles anidados

También será útil combinar varios procesos iterativos. Hay muchos problemas que se tienen que resolver "por fuerza bruta", esto es, haciendo que el ordenador trabaje diferentes posibilidades y busque una solución. En cuanto estudiemos ecuaciones diferenciales veremos que muy habitualmente esto es así.

EJEMPLO 3.2. Escribe dos bucles que reproduzcan esta matriz

$$\begin{bmatrix} 1 & 0{,}5 & 0{,}3333333 & 0{,}25 & 0{,}2 \\ 2 & 1 & 0{,}6666667 & 0{,}5 & 0{,}4 \\ 3 & 1{,}5 & 1 & 0{,}75 & 0{,}6 \\ 4 & 2 & 1{,}333333 & 1 & 0{,}8 \\ 5 & 2{,}5 & 1{,}666667 & 1{,}25 & 1 \end{bmatrix}$$

Observa que $a_{ij} = i/j$.

Ejercicio 3.10

La orden `sample(1:100,25 )` nos proporciona 25 números enteros aleatorios, comprendidos entre 1 y 100.

a) Escribe una matriz A de tamaño 5×5 cuyas entradas son números aleatorios entre 1 y 100.

b) Escribe una matriz B de tamaño 5×5 en la que el elemento $b_{ij} = 0$ cuando a_{ij} es par y $b_{ij} = 1$ cuando a_{ij} es impar. Hazlo con dos bucles anidados (aunque habría otras formas sencillas de hacerlo).

c) Escribe una matriz C de tamaño 5×5 que coincida con A pero en la que los números menores que 10 se han sustituido por 0.

Pensemos en el siguiente problema: "Calcula todos los números de tres cifras que coincidan con la suma de los cubos de las cifras que los componen". Nos aproximaremos a él de dos formas diferentes.

```
for (i in 0:9){   #genera la cifra de las unidades
 for (j in 0:9){   #genera las decenas
   for (k in 1:9){ #genera las centenas
if (100*k+10*j+i==k^3+j^3+i^3) print (k^3+j^3+i^3)
   }
 }
}
```

Otra posibilidad para calcular esos números pasaría por generar con un bucle todos los números de tres cifras y verificar si cumplen la condición impuesta. Para extraer las cifras de centenas y decenas hay que usar la *división entera* `%/%`.

```
for (i in 100:999){
 centenas<- i %/% 100
 decenas<-(i-centenas*100) %/% 10
 unidades <- (i-centenas*100-decenas * 10)
 if (i==centenas ^3 + decenas ^3 + unidades ^3) {
  print (i)
 }
}
```

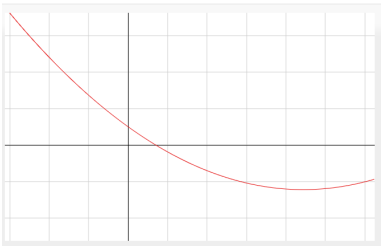
Capítulo 4

Automatizando procesos: programación

1. Aproximación de los ceros de una función

Es bien conocido el

TEOREMA DE BOLZANO:
Si $f : [a, b] \longrightarrow \mathbb{R}$ *es una función continua tal que* $f(a) \cdot f(b) < 0$, *entonces existe* $x_0 \in (a, b)$ *tal que* $f(x_0) = 0$.



Con la ayuda de un ordenador, y este teorema, podemos obtener valores aproximados, con la cota de error que queramos, de las raíces de una función continua.

Ejercicio 4.1

a) Construye una rutina que nos proporcione una raíz de una función f, con un error menor que una cota dada $\varepsilon > 0$, dentro de un intervalo $[a, b]$ cuando $f(a) \cdot f(b) \leq 0$.

"Resolución" Posibles pasos/consideraciones:

- hay, al menos, una raíz de f en el intervalo $[a, b]$
- si $f(a) = 0$ o $f(b) = 0$ hemos acabado. En otro caso, seguimos
- la longitud de $[a, b]$ es $b - a$
- tomemos el punto medio de $[a, b]$, $\frac{a+b}{2}$
- calculemos $f(\frac{a+b}{2})$
- si $f(\frac{a+b}{2}) = 0$ ya tenemos una raíz, y hemos acabado
- si $f(\frac{a+b}{2}) \neq 0$ su signo será contrario bien al de $f(a)$ bien al de $f(b)$
- en cualquier caso, disponemos de un intervalo, $[a, \frac{a+b}{2}]$ o $[\frac{a+b}{2}, b]$, de longitud $\frac{b-a}{2}$, en el que f cambia de signo y, por el teorema de Bolzano, posee al menos una raíz
- repetimos lo hecho sobre el intervalo $[a, b]$, ahora, con su escogido subintervalo
- . . . . .
- en general, tras repetir esta acción n veces tendremos un intervalo de longitud $\frac{b-a}{2^n}$ en el que residirá una raíz de f
- como $\lim_{n\to\infty} \frac{b-a}{2^n} = 0$, a partir de un n adecuado tendremos *localizada* una raíz de f con un error inferior a la cota impuesta $\varepsilon > 0$

Comienza a escribir la rutina. Observa que dependerá de cuatro parámetros (f, a, b, ε).

b) Comprueba que tu rutina funciona correctamente con la función $f : \mathbb{R} \longrightarrow \mathbb{R}$, $f(x) = x^4 - x^2 - 2$, sobre el intervalo $[0, 2]$, donde tiene una raíz que es $\sqrt{2}$.

Ejercicio 4.2

a) Elabora una rutina tal que dada una función f y un intervalo $[M, N]$ (donde $M, N \in \mathbb{Z}$) muestre un intervalo $[a, b]$ ($a, b \in \mathbb{Z}$), contenido en $[M, N]$ y de longitud 1, en el que la función f cambie de signo, esto es, $f(a) \cdot f(b) \leq 0$, siempre y cuando haya algún intervalo con esas características.

b) Comprueba que tu rutina funciona correctamente con la función $g : \mathbb{R} \longrightarrow \mathbb{R}$, $g(x) = \cos x$, sobre el intervalo $[-4, 3]$.

2. Listas en R

Hasta ahora, en R, para ordenar, coleccionar o agrupar objetos hemos usado *vectores* o *matrices*. En ambos casos sus elementos deben ser de la misma *clase* (números, caracteres, ...). Por ello no es posible, por ejemplo, que un elemento de una matriz sea un par de números (e.d. un elemento de $\mathbb{R}^2$). Podemos evitar esta rigidez si empleamos *listas*. En este momento no vamos a ahondar en esta herramienta, nos limitaremos a presentarla con unos ejemplos:

```
A<- matrix(6:9,2) ; A
v<- c(-2,0,5); v
w<- c(4,-11); w
L<- list(v,A,w,c(v,w),17) ; L    # la lista L consta de 5 objetos
L[[1]]            # primer objeto de L
L[[2]]            # segundo objeto de L
L[[4]]            # cuarto objeto de L
L[[5]]            # quinto objeto de L
L[[2]][1,2]       # del segundo objeto de L, su componente [1,2]
L[[3]][2]         # del tercer objeto de L, su componente [2]
```

Al igual que ocurre con vectores y matrices, podemos definir una lista *vacía* y la vamos rellenando según nos convenga:

```
L2<- list()              # creación de la lista vacía L2
L2
L2[[1]]<- matrix(1:8,2,byrow=T)  # imposición del primer objeto de L2
L2
L2[[1]]
L2[[3]]<- "nombre"          # imposición del tercer objeto de L2
L2                          # vemos que L2 carece de segundo elemento
L2[[2]]<- c(-3,5)           # imposición del segundo objeto de L2
L2                          # objetos de la lista L2
```

Ejercicio 4.3

a) Prepara una rutina tal que dada una función f y un intervalo $[M, N]$ (con $M, N \in \mathbb{Z}$) ofrezca una lista en la que estén todos los intervalos $[a, b]$ contenidos en $[M, N]$, de longitud 1 y extremos enteros, en los que f cambia de signo.

b) Comprueba que tu rutina funciona con la función $h : \mathbb{R} \longrightarrow \mathbb{R}$, $h(x) = x \cdot \operatorname{sen} x$, sobre el intervalo $[1, 15]$.

Ejercicio 4.4

a) Desarrolla una rutina tal que dada una función f y un intervalo $[M, N]$ (con $M, N \in \mathbb{Z}$) proporcione una lista en la que estén todos los intervalos de longitud $1/2$, y con algún extremo entero, contenidos en $[M, N]$ en los que f cambia de signo.

b) Comprueba que tu rutina funciona con la función $r : \mathbb{R} \longrightarrow \mathbb{R}$, $r(x) = \frac{x+1}{x^2+1} + \cos x$, sobre el intervalo $[0, 10]$.

Ejercicio 4.5

a) Construye una rutina tal que dada una función f, un intervalo $[M, N]$ (con $M, N \in \mathbb{Z}$) y un número $n \in \mathbb{N}$ ofrezca una lista en la que estén "todos" los intervalos $[a, b]$ contenidos en $[M, N]$, de longitud $1/n$, en los que f cambia de signo.

b) Emplea esta última rutina que has creado para mejorar las aproximaciones de las raíces que hemos detectado en los ejemplos precedentes. Podemos dibujar, con  , las gráficas de esas funciones, sobre los intervalos adecuados, para corroborar que no nos dejamos ninguna raíz.

3. Ejemplos básicos que se deben saber programar

1. Definir una función ORDEN tal que, dado un vector ordene sus componentes de menor a mayor utilizando un bucle. No es válido usar la orden SORT.
2. Dada una matriz A, con $A[1, 1] \neq 0$, elabora una función de forma que pivote en torno al elemento (1,1).

3. Dada una matriz A tal que $A[i, j] \neq 0$, proporciona una función que pivote en torno a A[i,j]
4. Dados una matriz A y un número n, indica cuantas de sus entradas son iguales a n.
5. Dada una matriz A y un número n, indica las posiciones de las entradas que son iguales a n. Si no hay ninguna, que lo diga.
6. Dados dos vectores, haz un programa que determine en qué posiciones hay valores coincidentes.
7. Dados dos vectores, haz un programa que diga si el primer elemento del primero aparece en alguna posición del segundo.
8. Dados dos vectores y un elemento del primero, haz un programa que diga siese elemento aparece en alguna posición del segundo. Que indique si aparece en su misma posición o no.
9. Dados dos vectores escribe un programa que digan qué elementos del primero aparecen en el segundo. Y en qué posiciones.
10. Dada una matriz A, escribe un programa que forme la matriz adjunta de A.
11. Calcula el término 523 de la sucesión definida por recurrencia $x_{n+1} = 11 + 2x_n$, con $x_1 = 1$.
12. Calcula la composición de una población acorde al modelo e Leslie después de 111 periodos.
13. Programar el método de Ruffini de división de polinomios
14. Programa el método de bipartición (de nuevo)
15. Dado un polinomio de grado n (por medio de sus coeficientes, como vector) haz un programa que proporcione su derivada.
16. Dado un polinomio de grado n (por medio de sus coeficientes, como vector) haz un programa que proporcione su derivada segunda.
17. Dado un polinomio de grado n (por medio de sus coeficientes, como vector) haz un programa que nos de una primitiva de ese polinomio.
18. Dada una combinación jugada de 6 números entre 1 y 49 y la combinación ganadora (6 números entre 1 y 49 más uno complementario), haz un programa que diga si te ha correspondido algún premio en la lotería primitiva.
19. Haz un programa que calcule el rango de una matriz. Además el programa debe ofrecer tantas filas independientes como indica ese rango.
20. Dada una matriz cuadrada M, haz un programa que proporcione una matriz simétrica S y una matriz antisimétrica A tales que M=S+A

21. Determina qué números entre 1 y 10000 son suma de cubos
22. Genera todos los números capicúas menores que 1000.
23. Dada una cantidad menor de 5 euros, que nos diga cuántas monedas de 2€, 1€, 50c, 20c, 10c, 5c, 2c y 1c tenemos que usar para obtener esa cantidad usando la menor cantidad posible de monedas.
24. Haz un programa que genere las 24 permutaciones posibles de los números 1,2,3,4

4. Ejemplos extraidos de exámenes

Ejercicio 4.6

El tamaño de una población evoluciona con el tiempo del siguiente modo: el número de individuos de una población en la etapa siguiente es el número de individuos de la etapa actual más el doble de los que había en la etapa anterior.

$$x_{n+1} = x_n + 3x_{n-1}$$

Supongamos que inicialmente la población es de 120 individuos.
(La sucesión asociada al tamaño de la población, en estas condiciones, sería 120, 120, 480, 840, 2280, 4800, ...)

1 Define una función (dependiente de n) que indique el número de individuos que habrá en la etapa n.

2 Indica cuántas etapas han de transcurrir para que la población supere el millón de individuos.

Ejercicio 4.7

a) Describe qué hace este procedimiento.

```
F1=function(A,n){
  for (i in 1:nrow(A)){
    for (j in 1:ncol(A)){
      if (A[i,j]==n){
        print(c(i,j))
      }
    }
  }
}
```

b) Indica qué daría como resultado ejecutar $F1\left(\begin{pmatrix}1&2&3\\4&3&-1\\1&0&2\end{pmatrix},3\right)$

Ejercicio 4.8

Escribe un procedimiento o función que proporcione la suma

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots + \frac{1}{999}$$

Ejercicio 4.9

Escribe una función `pares` que, al actuar sobre un vector de tamaño arbitrario, formado por números enteros, nos diga cuántas entradas pares tiene.

Por ejemplo,

$$\texttt{pares}(1,1,2,3,4,1,1,8) = 3$$
$$\texttt{pares}(0,0,0,0) = 4$$
$$\texttt{pares}(1,1,2,1,1,18,15) = 2$$

Comprueba que tu programa funciona. Para ello utiliza la orden `v=sample(1:100, size=10)` para generar diferentes vectores `v` de tamaño 10, cuenta manualmente la cantidad de números pares que hay en v y comprueba que la función que has definido es correcta.

Ejercicio 4.10

¿Qué resultado proporciona la ejecución de este código?

```
m<-matrix(1:6,3,2)
m
```

Ejercicio 4.11

a) Describe qué hace este procedimiento.

```
F1=function(v)
{sump=0
  for (i in 1:length(v)){
    if ((i%%2)==0 ){
   sump=sump+v[i]
    }
  }
sump
}
```

b) Indica cuál sería el resultado de aplicar la función $F1$ descrita en el apartado anterior a los vectores siguientes:

```
v1=1:5
v2=2*v1+1
```

Ejercicio 4.12

¿Qué aparecerá en pantalla al ejecutar este código?

```
i <- 1
n <- 5
while (i <= n) {
  print(i)
  i = i + 1
}
```

Ejercicio 4.13

Escribe un programa en R para crear una matriz con 30 filas y 50 columnas en la que aparezcan correlativamente los números pares mayores que 50.

$$\begin{bmatrix} 52 & 112 & \cdots & \cdots & \cdots \\ 54 & 114 & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \cdots & \cdots \end{bmatrix}$$

Ejercicio 4.14

a) Escribe un programa que decida si, dados dos vectores de longitud 10, v y w, el primer elemento de v aparece en alguna posición de w.

b) Escribe un programa que decida si, dados dos vectores de longitud 10, v y w, algún elemento de v aparece en alguna posición de w.

c) Escribe un programa que, dados dos vectores de longitud 10 v y w, proporcione una matriz $A = (a_{ij})$ de modo que

$$a_{ij} = \begin{cases} 0 & \text{si } v_i \neq w_j \\ 1 & \text{si } v_i = w_j \end{cases}$$

d) Comprueba en diferentes situaciones que el programa que ofreces funciona. Por ejemplo, si $v = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ y $w = (2, 0, 1, 3, 4, 9, 0, 1, 2, 2)$ la matriz resultante debería ser

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Capítulo 5

Modelos discretos

"Las matemáticas son el lenguaje con el que Dios ha escrito el universo." Galileo Galilei

La modelización matemática es una técnica cuyo objetivo es obtener ecuaciones (modelos) que describan el comportamiento y propiedades de un fenómeno real. Ningún modelo es una representación perfecta de la realidad.

1. Cadenas de Markov

Ejercicio 5.1

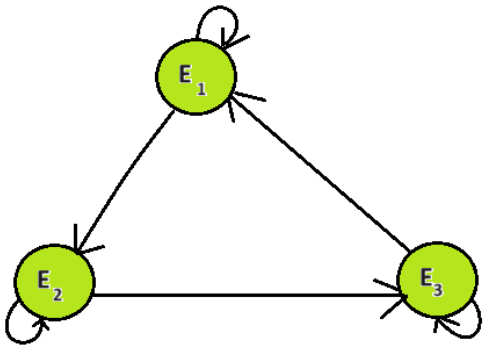
En cierto lugar se ha producido un incendio en un encinar. Después de la extinción del incendio, se ha clasificado el terreno en zonas de arbusto, quemadas y de regeneración del encinar. Tras un estudio se ha concluido que, después de un año, el 5 % de las zonas de las encinas regenerados vuelven a quemarse, el 30 % de las zonas de los arbustos se convierte en zonas de encinas regeneradas y el 40 % de las zonas quemadas se transforma en zonas de arbustos. Si la situación posteriormente a un incendio es 60 zonas quemadas, 20 de arbusto y 10 de árboles, cuál será la situación que se puede esperar del encinar cuando transcurran 20 años?

Las cadenas de Markov finitas es un modelo matemático que se utiliza para modelizar diferentes fenómenos en numerosas campos como por ejemplo el de la ingeniería, la informática, la epidemiología, etc. Permiten describir la evolución a la larga de un modelo utilizando para ello la probabilidad.

Supuestos para el modelo:

- La población está dividida en n estados o sucesos $\{E_1, E_2, \cdots, E_n\}$. En el ejercicio 5.1 hay tres estados o sucesos: $\{E_1 =$ " zona quemada", $E_2 =$ " zona de arbustos ", $E_3 =$ " zona regenerada"$\}$
- La probabilidad de que ocurra un suceso depende únicamente del suceso inmediatamente anterior y la denotaremos $p_{ij} = p(E_i E_j)$, probabilidad de pasar del estado E_j al estado E_i.

P_{ij}	E_1	E_2	E_3
E_1	$p_{11} = 0.6$	$p_{12} = 0$	$p_{13} = 0.05$
E_2	0.4	0.7	0
E_3	0	0.3	0.95



- Denotamos por $x_(k)(t)$ a la población que hay en el estado o suceso E_k en el momento t y por $\vec{x}(t)$ al vector cuyas componentes son la distribución de cada estado o suceso en el instante t.

$$\vec{x}(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{pmatrix}$$

Se supone conocido el vector $\vec{x}(0)$, es decir, al vector cuyas componentes son la distribución de cada estado o suceso al comienzo del estudio (instante $t = 0$).

$$\vec{x}(0) = \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ \vdots \\ x_{n-1}(0) \\ x_n(0) \end{pmatrix} \quad \text{en nuestro ejercicio 5.1} \quad \vec{x}(0) = \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{pmatrix} = \begin{pmatrix} 60 \\ 20 \\ 10 \end{pmatrix}$$

En nuestro ejercicio podemos calcular la distribución de las zonas pasado un año ($t = 1$, $\vec{x}(1)$) La cantidad de zonas quemadas después de un año será:

$$x_1(1) = 0{,}6\, x_1(0) + 0\, x_2(0) + 0{,}05\, x_3(0) = \begin{pmatrix} 0{,}6 & 0 & 0{,}05 \end{pmatrix} \begin{pmatrix} 60 \\ 20 \\ 10 \end{pmatrix} = 36{,}5$$

La cantidad de zonas de arbustos y árboles regenerados

$$x_2(1) = 0{,}4\, x_1(0) + 0{,}7\, x_2(0) + 0\, x_3(0) = \begin{pmatrix} 0{,}4 & 0{,}7 & 0 \end{pmatrix} \begin{pmatrix} 60 \\ 20 \\ 10 \end{pmatrix} = 38$$

$$x_3(1) = 0\, x_1(0) + 0{,}3\, x_2(0) + 0{,}95\, x_3(0) = \begin{pmatrix} 0 & 0{,}3 & 1 \end{pmatrix} \begin{pmatrix} 60 \\ 20 \\ 10 \end{pmatrix} = 15{,}5$$

Es decir,

$$\vec{x}(1) = \begin{pmatrix} x_1(1) \\ x_2(1) \\ x_3(1) \end{pmatrix} = \underbrace{\begin{pmatrix} 0{,}6 & 0 & 0{,}05 \\ 0{,}4 & 0{,}7 & 0 \\ 0 & 0{,}3 & 0{,}95 \end{pmatrix}}_{M} \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{pmatrix} = \mathrm{M}\,\vec{x}(0) = \begin{pmatrix} 36{,}5 \\ 38 \\ 15{,}5 \end{pmatrix}$$

Ejercicio 5.2

¿Cuál es la situación del encinar pasados dos años, tres o 10?

En general si suponemos que la población al cabo de k periodos viene dada por el vector $\vec{x}(k)$, en el periodo siguiente, la distribución de la población por estados o sucesos será

$$\begin{aligned} x_1(k+1) &= p_{11}x_1(k) + p_{12}x_2(k) + p_{13}x_3(k) + \cdots + p_{1n}x_n(k) \\ x_2(k+1) &= p_{12}x_1(k) + p_{22}x_2(k) + p_{23}x_3(k) + \cdots + p_{2n}x_n(k) \\ \vdots \quad &\quad \vdots \qquad\qquad\qquad \vdots \\ x_n(k+1) &= p_{n1}x_1(k) + p_{n2}x_2(k) + p_{n3}x_3(k) + \cdots + p_{nn}x_n(k) \end{aligned}$$

Matricialmente podemos escribir estas ecuaciones como

$$\vec{x}(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{pmatrix} = \mathrm{M}\,\vec{x}(k)$$

De manera general, a la matriz con la estructura de la matriz M se le conoce con el nombre de matriz de **Markov.**

Tenemos que

$$\vec{x}(k+1) = \mathrm{M}\,\vec{x}(k) = \mathrm{M}^2\,\vec{x}(k-1) = \cdots = \mathrm{M}^{k+1}\vec{x}(0).$$

EJEMPLO 5.1. Una empresa de autocares dispone de tres conductores (A, B, C) para hacer la ruta de Madrid a Sevilla. Se sabe que si A hace la ruta, la siguiente ruta la hace el conductor B y si la ruta la hace B, la siguiente ruta la hará C, pero si C hace la ruta la siguiente ruta la puede hacer A o B.

Este ejemplo se describe matricialmente de la siguiente manera

$$\vec{x}(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 1/2 \\ 1 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix}}_{\text{matriz de Markov, M}} \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{pmatrix} = \mathrm{M} \cdot \vec{x}(k)$$

Si la primera la ruta Madrid-Sevilla la hace el conductor A, ¿cuál es la probabilidad de que A haga la cuarta ruta $\vec{x}(3)$)?

$$\vec{x}(1) = \mathrm{M}\cdot\vec{x}(0) = \begin{pmatrix} 0 & 0 & 1/2 \\ 1 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \text{El conductor A hará la segunda ruta con probabilidad } 0$$

$$\vec{x}(2) = \mathrm{M}\cdot\vec{x}(1) = \begin{pmatrix} 0 & 0 & 1/2 \\ 1 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \text{El conductor A hará la tercera ruta con probabilidad } 0$$

$$\vec{x}(3) = \mathrm{M} \cdot \vec{x}(2) = \begin{pmatrix} 0 & 0 & 1/2 \\ 1 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix}$$

$$\vec{x}(3) = \mathrm{M}^3 \cdot \vec{x}(0) = \begin{pmatrix} 1/2 & 0 & 1/4 \\ 1/2 & 1/2 & 1/4 \\ 0 & 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix}$$

Por tanto, la probabilidad de que el conductor A haga la cuarta ruta es de 1/2.

2. Propiedades de la matriz de Markov

Supongamos que M es una matriz de Markov de tamaño n.

- Todos los elementos de la matriz verifican que son mayores o iguales a cero.

$$p_{ij} \geq 0 \text{ para todo } \; i, j = 1, \cdots, n$$

- la suma de todos los elementos o entradas de cualquier columna de M suman uno:

$$\sum_{i=1}^{n} pij = 1 \;\text{ para todo }\; j = 1, \cdots, n$$

- M tiene a $\lambda = 1$ como autovalor.

$$|\mathrm{M} - \mathrm{I}| = \left| \begin{pmatrix} p_{11} - 1 & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} - 1 & p_{23} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} - 1 - 1 \end{pmatrix} \right| = \left| \begin{pmatrix} \overbrace{\sum_{i=1}^{n} p_{i1}}^{1} - 1 & \overbrace{\sum_{i=1}^{n} p_{i2}}^{1} - 1 & \overbrace{\sum_{i=1}^{n} p_{i3}}^{1} - 1 & \cdots & \overbrace{\sum_{i=1}^{n} p_{in}}^{1} - 1 \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} \end{pmatrix} \right|$$

$$= \left| \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ p_{21} & p_{22} - 1 & p_{23} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} - 1 - 1 \end{pmatrix} \right| = 0 \Longrightarrow \quad \lambda_0 = 1 \;\text{ es un autovalor de M}$$

- Cualquier otro autovalor λ_r de una matriz de Markov M verifica que

$$|\lambda_\mathrm{r}| \leq \lambda_0 = 1$$

- Si M es una matriz de Markov y existe un $k \in \mathbb{N}$ tal que M^k es una matriz con todos sus elementos positivos (**regular**) entonces $\lambda_0 = 1$ es un autovalor de multiplicidad algebraica 1 y **dominante**, es decir, para cualquier otro autovalor λ_r se verifica que

$$|\lambda_\mathrm{r}| < \lambda_0 = 1$$

Ejercicio 5.3

$$M = \begin{pmatrix} 0{,}6 & 0 & 0{,}05 \\ 0{,}4 & 0{,}7 & 0 \\ 0 & 0{,}3 & 0{,}95 \end{pmatrix} \quad M^2 = \begin{pmatrix} 0{,}36 & 0{,}015 & 0{,}0775 \\ 0{,}52 & 0{,}490 & 0{,}0200 \\ 0{,}12 & 0{,}495 & 0{,}9025 \end{pmatrix} \Longrightarrow M \text{ es regular}$$

EJEMPLO 5.2.

Comportamiento en el límite de la matriz regular.

EJEMPLO 5.3. Las familias de una comunidad autónoma se clasifican según datos bancarios como conservadores, de riesgo medio o de riesgo alto. Los estudios estiman que en promedio, en el curso de un año, el 50 % de los ahorradores cambian a riesgo medio y el otro 50 % cambia a riesgo alto. 25 % de las personas de riesgo medio cambian a conservadores y el 50 % a riesgo alto; y finalmente el 25 % de los de alto riego pasan a conservadores y un 50 % cambian a riesgo medio

- La matriz M= $\mathrm{M} = \begin{pmatrix} 0 & 1/4 & 1/4 \\ 1/2 & 1/4 & 1/2 \\ 1/2 & 1/2 & 1/4 \end{pmatrix}$ es una matriz regular ya que M^2

$$\begin{pmatrix} 1/4 & 1/4 & 1/8 \\ 1/2 & 1/2 & 3/8 \\ 1/2 & 1/4 & 1/2 \end{pmatrix}$$

Por tanto su autovalor $\lambda = 1$ verifica que es de multiplicidad 1 y es dominante

$$\mathrm{M} = \begin{pmatrix} 0 & 1/4 & 1/4 \\ 1/2 & 1/4 & 1/2 \\ 1/2 & 1/2 & 1/4 \end{pmatrix} \Longrightarrow |\mathrm{M} - \lambda \mathrm{I}| = -\lambda^3 + \frac{\lambda^2}{2} + \frac{7\lambda}{+}\frac{1}{16} = 0 \Longrightarrow \lambda = 1,\ \frac{-1}{4},$$

El autovalor $\lambda = 1$ es dominante y su autoespacio es

$$E(1) = \left\{v \in \mathbb{R}^3 / \mathrm{M}v = v\right\} = \left\{v \in \mathbb{R}^3 / (\mathrm{M} - \mathrm{I}v = 0\right\} =$$

$$\left\{ v \in \mathbb{R}^3 / \begin{pmatrix} -1 & 1/4 & 1/4 \\ 1/2 & -3/4 & 1/2 \\ 1/2 & 1/2 & -3/4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\} = L\{(1, 2, 2)\}$$

Si suponemos que la distribución inicial ($t = 0$) viene dada por el vector $\vec{x}(0) = (x_1(0), x_2(0), x_3(0))$ (donde $x_i(0)$ es la cantidad de la población total N que hay en el estado i)y teniendo en cuenta que los autovectores asociados a los autovalores $\lambda = 1, 1/4$ son

$$E(1) = L\{(1,2,2) = L\{u\} \quad E(1/4) = L\{(-1,1,0),(-1,0,1)\}$$

obtenemos

$$\lim_{k\to\infty} \vec{x}(k) = \lim_{k\to\infty} \mathrm{M^k}\cdot\vec{x}(0) = \lim_{k\to\infty} \mathrm{P}\cdot\begin{pmatrix}1&0&0\\0&1/4&0\\0&0&0\end{pmatrix}^k \cdot \mathrm{P}\cdot\vec{x}(0) = \underbrace{\begin{pmatrix}1&-1&-1\\2&1&0\\2&0&1\end{pmatrix}}_{\mathrm{P}}\cdot\begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix}\cdot$$

$$\underbrace{\begin{pmatrix}1/5&1/5&1/5\\-2/5&3/5&-2/5\\-2/5&-2/5&3/5\end{pmatrix}}_{\mathrm{P^{-1}}}\cdot\begin{pmatrix}x_1(0)\\x_2(0)\\x_3(0)\end{pmatrix} = \underbrace{\begin{pmatrix}1&-1&-1\\2&1&0\\2&0&1\end{pmatrix}}_{\mathrm{P}}\cdot\begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix}\cdot\underbrace{\begin{pmatrix}1/5(x_1(0)+x_2(0)+x_3(0))\\-2/5x_1(0)+3/5x_2(0)-2/5x_3(0)\\-2/5x_1(0)-2/5x_2(0)+3/5x_3(0)\end{pmatrix}}_{\mathrm{P^{-1}\cdot x(0)}}$$

$$= \underbrace{\begin{pmatrix}1&-1&-1\\2&1&0\\2&0&1\end{pmatrix}}_{\mathrm{P}}\cdot\underbrace{\begin{pmatrix}1/5(x_1(0)+x_2(0)+x_3(0))\\0\\0\end{pmatrix}}_{\mathrm{D^k\cdot P^{-1}\cdot x(0)}} = \begin{pmatrix}1/5(x_1(0)+x_2(0)+x_3(0))\\2/5(x_1(0)+x_2(0)+x_3(0))\\2/5(x_1(0)+x_2(0)+x_3(0))\end{pmatrix} = \begin{pmatrix}1/5\\2/5\\2/5\end{pmatrix}\cdot N = 1/5N\begin{pmatrix}1\\2\\2\end{pmatrix}$$

Observar también que

$$\begin{pmatrix}1&-1&-1\\2&1&0\\2&0&1\end{pmatrix}\begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix}\begin{pmatrix}1/5&1/5&1/5\\2/5&2/5&2/5\\2/5&2/5&2/5\end{pmatrix} = \begin{pmatrix}1/5&1/5&1/5\\2/5&2/5&2/5\\2/5&2/5&2/5\end{pmatrix} = \frac{1}{5}\begin{pmatrix}1&1&1\\2&2&2\\2&2&2\end{pmatrix}$$

-¿Qué ocurrirá, a la larga, con el tamaño de la población ? ¿Y con la proporción de individuos en cada clase?

A la larga la población se estabiliza distribuyéndose el 20 % de la población en la clase de ahorradores, 40 % en riesgo alto y medio independientemente del la distribución inicial o de partida. En general, para conocer la distribución a la larga no hace falta hacer el cálculo de $P\cdot M^k\cdot P^{-1}$, basta con calcular el autoespacio asociado al autovalor $\lambda = 1$ r escoger un autovector cuyas coordenadas sumen 1.

TEOREMA A. *Si M es una matriz de Markov de orden n regular, entonces la sucesión de matrices M^n tiende a una matriz con todas sus columnas iguales M= $(\vec{c},\vec{c},\cdots,\vec{c})$ con $\vec{c} = (c_1, c_2, \cdots, c_)$ tal que*

- $\vec{c}$ es un autovector asociado al autovalor $\lambda_0 = 1$.

- $\sum_{i=1}^{n} c_i = 1$

Puesto que M es una matriz de markov regular, $\lambda_1 = 1$ es un autovalor dominante de la matriz de la matriz. Es decir,

$$\mathrm{M} = \mathrm{P} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} \cdot \mathrm{P}^{-1}$$

Por tanto

$$\vec{x}(k) = \mathrm{M^k} \cdot \vec{x}(0) = \mathrm{P} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \lambda_2^k & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_n^k \end{pmatrix} \cdot \mathrm{P}^{-1} \cdot \vec{x}(0)$$

Si denotamos a la primera fila de la matriz P^{-1} por $c_1, c_2, \cdots, c_n = \vec{c}$ tenemos

$$\lim_{k\to\infty} \vec{x}(k) = \lim_{k\to\infty} \mathrm{M^k} \cdot \vec{x}(0) = \mathrm{P} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \cdot \mathrm{P}^{-1} \cdot \vec{x}(0) = \mathrm{P} \cdot \begin{pmatrix} c_1 & c_2 & \cdots & c_n \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \cdot \vec{x}(0) =$$

$$= \mathrm{P} \cdot \begin{pmatrix} c \cdot \vec{x}(0) \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \vec{v_1} \underbrace{c \cdot \vec{x}(0)}_{C} = C\vec{v_1}$$

Es decir para k suficientemente grandes podemos suponer que

$$\vec{x}(k) \approx C\vec{v_1}$$

Puesto que $\vec{x}(k-1) \approx C\vec{v_1}$ se sigue que a la larga $(k \to \infty)$

$$\vec{x}(k) \approx C\vec{v_1} \approx \vec{x}(k-1)$$

Observamos que el vector de distribución a la larga se estabiliza y es igual al autovector asociado a $\lambda = 1$ cuyas componentes suman 1. Además es independiente del vector inicial $\vec{x}(0)$.

3. Poblaciones estructuradas

En un bosque se han clasificado los árboles, por su altura en n clases (población estructurada). Se ha elegido la altura como variable para la clasificación porque en este modelo es la que marca el precio de mercado de los árboles. Organizamos la información sobre estos aspectos correspondiente al estado inicial (momento en que observamos por primera vez por primera vez la población) en la siguiente tabla:

Clase	**1**	**2**	**3**	$\cdots$	**n-1**	**n**
Altura	$[0, h_1)$	$[h_1, h_2)$	$[h_2, h_3)$		$[h_{n-2}, h_{n-1})$	$[h_{n-1}, \infty)$
Número inicial de árboles	x_1	x_2	x_3		x_{n-1}	x_n
Valor	0	p_2	p_3		p_{n-1}	p_n

Observamos de nuevo la población de árboles al cabo de un periodo de tiempo T. Los árboles habrán crecido y parte de ellos habrán pasado a la clase de altura siguiente. Suponemos que el periodo de tiempo T trascurrido permite sólo un crecimiento tal que un árbol pasa como mucho a la clase de altura siguiente. Adicionalmente, asumiremos que ningún árbol muere en ninguna de las clases.

Analizada la evolución de la altura de la población se han estimado, para cada una de las clases de altura la proporción de individuos que pasan a la clase siguiente. Denotaremos por g_i a la proporción de individuos de la clase i que pasa a la clase $i+1$. Por lo tanto la proporción de individuos que permanece en su clase es $1 - g_i$.

Además, asumiremos que esos parámetros son estables en el tiempo, es decir, se mantienen para periodos de tiempo iguales a T.

Supongamos que la población al cabo de k periodos de tiempo es

$$\vec{x}(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{pmatrix}$$

En el periodo siguiente, la distribución de árboles por clases de altura será

$$\begin{aligned} x_1(k+1) &= (1-g_1)x_1(k) \\ x_2(k+1) &= (1-g_2)x_2(k) + g_1 x_1(k) \\ x_3(k+1) &= (1-g_3)x_3(k) + g_2 x_2(k) \\ &\cdots \\ x_{n-1}(k+1) &= (1-g_{n-1})x_{n-1}(k) + g_{n-2}x_{n-2}(k) \\ x_n(k+1) &= x_n(k) + g_n x_{n-1}(k) \end{aligned}$$

Matricialmente podemos escribir esto como

$$\vec{x}(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{pmatrix} = \begin{pmatrix} 1-g_1 & 0 & 0 & \cdots & 0 & 0 \\ g_1 & 1-g_2 & 0 & \cdots & 0 & 0 \\ 0 & g_2 & 1-g_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-g_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & g_{n-1} & 1 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{pmatrix}$$

Llamando $G = \begin{pmatrix} 1-g_1 & 0 & 0 & \cdots & 0 & 0 \\ g_1 & 1-g_2 & 0 & \cdots & 0 & 0 \\ 0 & g_2 & 1-g_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-g_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & g_{n-1} & 1 \end{pmatrix}$

tenemos que

$$\vec{x}(k+1) = G\vec{x}(k) = G^2\vec{x}(k-1) = \cdots = G^k\vec{x}(0).$$

¿Qué ocurrirá con este bosque a largo plazo?

4. Modelo de gestión forestal sostenible

Supongamos un bosque en el que hay árboles de distintas alturas. El bosque se deja crecer por un tiempo y después los árboles son talados para ser vendidos.

Supuestos para el modelo:

- Por cada árbol cortado se planta uno nuevo en el mismo lugar.
- Todos los árboles plantados sobreviven hasta ser cortados
- Árboles de diferentes alturas tienen diferentes precios.
- Los árboles más pequeños (los de la primera clase) no tienen valor.
- Se conoce la matriz G de crecimiento, la que permite expresar la evolución de la población de árboles en altura según las clases de edad.

Posibles preguntas:
- ¿Por qué interesa cortar árboles de diferentes clases de edad?
- ¿No sería mejor cortar todos los de la mayor clase de edad?
- ¿Qué nos ofrece más ganancia de inmediato?
- ¿Qué nos ofrece más ganancia a largo plazo?

Supondremos que, al final de cada periodo de crecimiento, se talan y_j árboles de la clase j-ésima ($j = 2, \ldots, n$). No se talan árboles de la primera clase de altura puesto que su rentabilidad es nula.

Como en total se talan $y_2 + \cdots + y_n$ árboles, son esos los que hay que repoblar en la primera clase.

Así, matricialmente, resulta que

$$\vec{x}(k+1) = G \cdot \vec{x}(k) - \begin{pmatrix} 0 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} + \begin{pmatrix} y_2 + \cdots + y_n \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

EJEMPLO 5.4. Supongamos un bosque con cuatro clases de altura y matriz de crecimiento

$$G = \begin{pmatrix} 0{,}4 & 0 & 0 & 0 \\ 0{,}6 & 0{,}3 & 0 & 0 \\ 0 & 0{,}7 & 0{,}2 & 0 \\ 0 & 0 & 0{,}8 & 1 \end{pmatrix}$$

Supongamos que, inicialmente, la situación inicial es $\vec{x}_0 = \begin{pmatrix} 1000 \\ 1000 \\ 1000 \\ 1000 \end{pmatrix}$

El valor de cada árbol es de 100 €, 150 € o 300 €, según pertenezca a la clase segunda, tercera o cuarta.

Estudia qué ocurre en las tres primeras etapas y calcula su rendimiento (beneficio) con cada una de las tres gestiones siguientes:

a) El bosque no es intervenido y se deja evolucionar sin talas

b) Al final de cada etapa de crecimiento se talan 300 árboles de la tercera clase y 900 de la cuarta (que se vuelven a plantar)

c) Al final de cada etapa de crecimiento se talan todos los árboles de la última clase (que se vuelven a plantar)

De ser posible mantener cada gestión indefinidamente ¿se estabilizaría la composición del bosque con el tiempo?

DEFINICIÓN 5.1. Diremos que una gestión es *sostenible* si existen unas condiciones iniciales $\vec{x}_0$, de manera que partiendo de esas condiciones iniciales, la composición del bosque permanezca igual a las condiciones de partida al final de cada etapa.

Vamos a interpretar qué significa que una gestión sea sostenible y lo escribiremos en términos de matrices para encontrar condiciones que aseguren una gestión con estas características.

$$\begin{pmatrix} 1-g_1 & 0 & 0 & \cdots & 0 & 0 \\ g_1 & 1-g_2 & 0 & \cdots & 0 & 0 \\ 0 & g_2 & 1-g_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-g_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & g_{n-1} & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} - \begin{pmatrix} 0 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} + \begin{pmatrix} y_2 + \cdots + y_n \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}$$

Efectuando operaciones, esa condición se traduce en

$$\begin{aligned} g_1 x_1 &= y_1 + \cdots + y_n \\ g_1 x_1 - g_2 x_2 &= y_2 \\ g_2 x_2 - g_3 x_3 &= y_3 \\ &\vdots \\ g_{n-2} x_{n-2} - g_{n-1} x_{n-1} &= y_{n-1} \\ g_{n-1} x_{n-1} &= y_n \end{aligned}$$

La primera ecuación es la suma de todas las demás y, por tanto, puede eliminarse puesto que no aporta información nueva.

La única condición que hay sobre los y_i es que todos ellos sean números no negativos. Por tanto, será posible una gestión sostenible si

$$g_1x_1 \geq g_2x_2 \geq \cdots \geq g_{n-1}x_{n-1} \geq 0$$

El *rendimiento* es el valor que se obtiene por la venta de los árboles talados. Viene dado por

$$\begin{aligned} R &= p_2y_2 + p_3y_3 + \cdots + p_ny_n \\ &= p_2(g_1x_1 - g_2x_2) + p_3(g_2x_2 - g_3x_3) + \cdots + p_{n-1}(g_{n-2}x_{n-2} - g_{n-1}x_{n-1}) + p_ng_{n-1}x_{n-1} \\ &= p_2g_1x_1 + (p_3 - p_2)g_2x_2 + \cdots + (p_n - p_{n-1})g_{n-1}x_{n-1} \end{aligned}$$

EJEMPLO 5.5. Con la matriz de crecimiento $G = \begin{pmatrix} 0{,}4 & 0 & 0 & 0 \\ 0{,}6 & 0{,}3 & 0 & 0 \\ 0 & 0{,}7 & 0{,}2 & 0 \\ 0 & 0 & 0{,}8 & 1 \end{pmatrix}$ indica si con las siguientes composiciones es posible una gestión sostenible inmediata:

a) $\vec{x}_0 = (500, 1000, 2000, 500)$

b) $\vec{x}_0 = (1500, 1000, 500, 500)$

5. Modelo de Leslie

Este modelo es usado para el estudio de predicción de la evolución de poblaciones de hembras en poblaciones humanas o animales.

Supuestos para el modelo:

- La población (de hembras) está dividida en n clases de edad de igual amplitud y ordenadas. Por ejemplo dividiendo entre n la máxima edad que puede alcanzar una hembra en esa población. O considerando periodos de tiempo de los de uso habitual (años, meses, etc.)
- En cada etapa (intervalo de tiempo igual a la amplitud de las clases de edad) el número de hembras en cada clase de edad cambia por uno de tres posibles eventos: nacimiento, muerte y aumento en la edad. La probabilidad de que una hembra permanezca en una misma clase al cambiar de etapa es cero.

- En cada etapa y para cada clase i (i $\in 1, 2n, , ..., n-1$) se tiene que:

  - **Tasa de fertilidad** a_i: nacen, en media, a_i hembras por cada una de las que hay en esa clase. (Las que nacen formarán parte de la clase 1). Se sigue que $a_1 \geq 0$.

  - **Probabilidad de mortalidad** m_i. tasa de mortalidad para esa clase i;

  - **Probabilidad de supervivencia** $b_i = 1 - m_i$. La proporción de hembras de la clase i que pasan a la clase i+1 (por aumento de edad); Observar que $0 <\leq b_i \leq 1$. ya que si $b_i = 0$ implicaría que todas las hembras en la clase i mueren.
- Todos los individuos de la clase n mueren (un individuo vive como mucho n etapas)
- Se supone conocida la composición de cada clase de edad al comienzo del estudio instante $t = 0$ comienzo del estudio, es decir, conocemos el vector $\vec{x}(0)$

$$\vec{x}(0) = \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ \vdots \\ x_{n-1}(0) \\ x_n(0) \end{pmatrix}$$

Vamos a expresar estas hipótesis mediante ecuaciones. Supongamos que la población al cabo de k periodos de tiempo es

$$\vec{x}(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{pmatrix}$$

En el periodo siguiente, la distribución de la población por clases de altura será

$$\begin{aligned} x_1(k+1) &= a_1x_1(k) + a_2x_2(k) + a_3x_3(k) + \cdots + a_nx_n(k) \\ x_2(k+1) &= b_1x_1(k) \\ x_3(k+1) &= b_2x_2(k) \\ &\cdots \\ x_{n-1}(k+1) &= b_{n-2})x_{n-2}(k) \\ x_n(k+1) &= b_{n-1}x_{n-1}(k) \end{aligned}$$

Matricialmente podemos escribir esto como

$$\vec{x}(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n \\ b_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & b_2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & b_n & 0 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{pmatrix} = L\vec{x}(k)$$

De manera general, a la matriz con la estructura de la matriz de transición L se le conoce con el nombre de matriz de **Leslie.**

Tenemos que

$$\vec{x}(k+1) = L\vec{x}(k) = L^2\vec{x}(k-1) = \cdots = L^{k+1}\vec{x}(0).$$

EJEMPLO 5.6. Supongamos que las hembras de una población animal viven una media de 6 meses y que esta población se divide en tres clases de edades iguales con intervalos de 2 meses. Supongamos también que solo se reproducen en la última clase de manera que por cada hembra en esta clase nacen de media 6 crías. Además en cada periodo el 50 % de la primera clase pasa a la segunda y el 30 % de la segunda a la tercera. Si inicialmente hay 100 hembras de la primera clase, 60 de la segunda y 20 de la tercera, estudia la evolución de la población después de 2 y cuatro meses.

Este ejemplo se describe matricialmente como

$$\vec{x}(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 6 \\ 1/2 & 0 & 0 \\ 0 & 3/10 & 0 \end{pmatrix}}_{\text{matriz de Leslie, } L} \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{pmatrix} = L \cdot \vec{x}(k)$$

Como se quiere estudiar la evolución de la población después de 2 meses(1 periodo de tiempo, $\vec{x}(1)$) y 4 meses (2 periodos $\vec{x}(2)$) se tiene que

$$\vec{x}(1) = L \cdot \vec{x}(0) = \begin{pmatrix} 0 & 0 & 6 \\ 1/2 & 0 & 0 \\ 0 & 3/10 & 0 \end{pmatrix} \begin{pmatrix} 100 \\ 60 \\ 20 \end{pmatrix} = \begin{pmatrix} 120 \\ 50 \\ 18 \end{pmatrix}$$

$$\vec{x}(2) = L \cdot \vec{x}(1) = L^2 \cdot \vec{x}(0) = \begin{pmatrix} 0 & 1{,}8 & 0 \\ 0 & 0 & 3 \\ 0{,}15 & 0 & 0 \end{pmatrix} \begin{pmatrix} 100 \\ 60 \\ 20 \end{pmatrix} = \begin{pmatrix} 108 \\ 60 \\ 15 \end{pmatrix}$$

Propiedades de la matriz de Leslie

Supongamos que L es una matriz de Leslie de tamaño n.

- El polinomio característico de L es

  $$|\mathrm{L} - \lambda \mathrm{I}| = (-1)^{\mathrm{n}}(\lambda^{\mathrm{n}} - \mathrm{a}_1\lambda^{\mathrm{n}-1} - \mathrm{a}_2\mathrm{b}_1\lambda^{\mathrm{n}-2} - \mathrm{a}_3\mathrm{b}_1\mathrm{b}_2\lambda^{\mathrm{n}-3} - \cdots - \mathrm{a}_{\mathrm{n}}\mathrm{b}_1\mathrm{b}_2\cdots\mathrm{b}_{\mathrm{n}-1})$$

- Se llama **Tasa media de Reproducción** de una población a la expresión

  $$R = a_1 + a_2b_1 + a_3b_1b_2 + \cdots + a_nb_1b_2\cdots b_{n-1}$$

  que representa el promedio de crías que tiene una hembra durante su esperanza de vida.

- Una matriz de Leslie L, tiene un único autovalor positivo λ_1; el resto son cero, negativos o complejos. λ_1 es un autovalor simple (multiplicidad algebraica= 1 y por tanto la geométrica también). Su autovector asociado tiene todas sus coordenadas positivas y su autoespacio es

  $$E(\lambda_1) = L\{(1, \mathrm{b}_1/\lambda_1, (\mathrm{b}_1\mathrm{b}_2)/\lambda_1^2, (\mathrm{b}_1\mathrm{b}_2\mathrm{b}_3)/\lambda_1^3, \cdots, (\mathrm{b}_1\mathrm{b}_2\cdots\mathrm{b}_{\mathrm{n}-1})/\lambda_1^{\mathrm{n}})\}$$

- Cualquier otro autovalor λ_{r} verifica que

  $$|\lambda_{\mathrm{r}}| \leq \lambda_1$$

- si dos entradas consecutivas de la primera fila de la matriz son diferentes de cero el autovalor λ_1 es dominante. Es decir,

  $$|\lambda_{\mathrm{r}}| < \lambda_1$$

EJEMPLO 5.7. Una población de escarabajos está divida en tres clases de edades de un año de duración, a las que llamaremos crías, jóvenes y adultas. Se sabe que cada escarabajo joven aporta una media de 4 escarabajos crías y cada adulta un promedio de tres. Además se sabe que la mitad de la crías sobreviven para llegar a jóvenes y el 25 % de las jóvenes se hacen adultas.

- Dar la ecuación característica y el autovalor dominante si existe.

$$L = \begin{pmatrix} 0 & 4 & 3 \\ 1/2 & 0 & 0 \\ 0 & 1/4 & 0 \end{pmatrix} \Longrightarrow |L - \lambda I| = -\lambda^3 + 2\lambda + 3/8 \Longrightarrow \lambda = \frac{3}{2},\ \frac{-3+\sqrt{5}}{4},\ \frac{-3+\sqrt{5}}{4}$$

- Hallar los autovectores asociados al autovalor dominante.

Puesto que

$$\left|\frac{-3+\sqrt{5}}{4}\right| < \frac{3}{2} \qquad \left|\frac{-3-\sqrt{5}}{4}\right| < \frac{3}{2}$$

El autovalor $\frac{3}{2}$ es dominante y su autoespacio es

$$E\left(\frac{3}{2}\right) = \left\{v \in \mathbb{R}^3 / \, Lv = \frac{3}{2}v\right\} = \left\{v \in \mathbb{R}^3 / \, (L - \frac{3}{2}I)v = 0\right\} =$$

$$\left\{v \in \mathbb{R}^3 / \begin{pmatrix} -3/2 & 4 & 3 \\ 1/2 & -3/2 & 0 \\ 0 & 1/4 & -3/2 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\right\} = L\{(18, 6, 1)\}$$

Comportamiento en el límite de la matriz de Leslie con autovalor dominante.

Supongamos que λ_1 es un autovalor dominante de la matriz de Leslie y la matriz de Leslie es diagonalizable. Es decir,

$$\mathrm{L} = \mathrm{P} \cdot \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} \cdot \mathrm{P}^{-1}$$

Por tanto

$$\vec{x}(k) = \mathrm{L}^{\mathrm{k}} \cdot \vec{x}(0) = \mathrm{P} \cdot \begin{pmatrix} \lambda_1^k & 0 & \cdots & 0 \\ 0 & \lambda_2^k & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_n^k \end{pmatrix} \cdot \mathrm{P}^{-1} \cdot \vec{x}(0)$$

$$\frac{\vec{x}(k)}{\lambda_1^k} = \mathrm{L}^{\mathrm{k}} \cdot \vec{x}(0) = \mathrm{P} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{\lambda_2^k}{\lambda_1^k} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\lambda_n^k}{\lambda_1^k} \end{pmatrix} \cdot \mathrm{P}^{-1} \cdot \vec{x}(0)$$

Si denotamos a la primera fila de la matriz P^{-1} por $c_1, c_2, \cdots, c_n = \vec{c}$ tenemos

$$\lim_{k\to\infty}\frac{\vec{x}(k)}{\lambda_1^k} = \mathrm{L}^k\cdot\vec{x}(0) = \mathrm{P}\cdot\begin{pmatrix}1 & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0\\ \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & 0\end{pmatrix}\cdot\mathrm{P}^{-1}\cdot\vec{x}(0) = \mathrm{P}\cdot\begin{pmatrix}c_1 & c_2 & \cdots & c_n\\ \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & 0\end{pmatrix}\cdot\vec{x}(0) =$$

$$= \mathrm{P}\cdot\begin{pmatrix}c\cdot\vec{x}(0)\\ 0\\ \vdots\\ 0\end{pmatrix} = \vec{v_1}\underbrace{c\cdot\vec{x}(0)}_{C} = C\vec{v_1}$$

Es decir para k suficientemente grandes podemos suponer que

$$\vec{x}(k)\approx\lambda_1^k C\vec{v_1}$$

Puesto que $\vec{x}(k-1)\approx\lambda_1^{k-1}C\vec{v_1}$ se sigue

$$\vec{x}(k)\approx\lambda_1^k C\vec{v_1} = \lambda_1\lambda_1^{k-1}C\vec{v_1}\approx\lambda_1\vec{x}(k-1)$$

Observamos que cada vector de distribución es un múltiplo del vector de distribución anterior, siendo ese múltiplo el autovalor dominante λ_1. Además la proporción de población en cada clase se mantiene constante.

Por tanto, si $\lambda_1 > 1$ la población crece, si $\lambda_1 < 1$ la población decrece y si $\lambda_1 = 1$ la población tiende a estabilizarse.

Si v_1 es el autovector asociado al autovalor dominante λ_1, el porcentaje población en la clase i-ésima será:

$$\frac{v_i}{\sum_{j=1}^n v_j}100$$

EJEMPLO 5.8. Una población de escarabajos está divida en tres clases de edades de un año de duración, a las que llamaremos crías, jóvenes y adultas. Se sabe que cada escarabajo joven aporta una media de 4 escarabajos crías y cada adulta un promedio de tres. Además se sabe que la mitad de la crías sobreviven para llegar a jóvenes y el 25 % de las jóvenes se hacen adultas.

-¿Qué ocurrirá, a la larga, con el tamaño de la población ? ¿Y con la proporción de individuos en cada clase?

Cómo $\frac{3}{2} > 1$ es el autovalor dominante tenemos que

$$\vec{x}(k) \approx \left(\frac{3}{2}\right)^k C\vec{v_1} = \left(\frac{3}{2}\right)\vec{x}(k-1) = \left(\frac{3}{2}\right)^k C\begin{pmatrix}18\\6\\1\end{pmatrix}$$

Por tanto, la población crecerá sin límite un 50 % en cada etapa.

Puesto que el autovector asociado a $\left(\frac{3}{2}\right)$ es $v = (18, 6, 1)$ el porcentaje de población en cada clase es

$$\begin{cases} \frac{v_1}{\sum_{j=1}^{3} v_j}100 = \frac{18}{25}100 = 72\,\% \\ \frac{v_2}{\sum_{j=1}^{3} v_j}100 = \frac{6}{25}100 = 24\,\% \\ \frac{v_3}{\sum_{j=1}^{3} v_j}100 = \frac{1}{25}100 = 4\,\% \end{cases}$$

Obsérvese que si $\lambda_1 = 1$ es autovalor entonces la tasa neta de reproducción toma el valor 1

$$R = a_1 + a_2b_1 + a_3b_1b_2 + \cdots + a_nb_1b_2\cdots b_{n-1} = 1$$

6. Ejercicios variados

Ejercicio 5.4

Para estudiar una población de petirrojos hembra cuya edad máxima es de 3 años se divide en tres clases de edad, correspondiente a un año de duración cada una. De la observación se deduce que sólo una cuarta parte de las de la primera clase (0-1 años) sobreviven hasta el siguiente periodo de tiempo y que sólo la mitad de las de la segunda clase (1-2 años) sobreviven hasta que alcanzan la última clase (2-3 años). Asimismo se observa que, en promedio, cada hembra petirrojo de la primera clase procrea una nueva hembra, mientras que las de la segundo clase procrean 8 nuevas hembras.

a) Escribe el modelo de Leslie para estos datos en forma matricial y decide, justificadamente si la dicha matriz tiene un autovalor dominante.

b) ¿Qué ocurrirá, a largo plazo, con el tamaño de la población? ¿Y con la proporción de individuos en cada clase?.

Ejercicio 5.5

En un pinar se han considerado 6 clases arbóreas en función del área basimétrica de cada uno de los pinos. La clase 1 se corresponde con los de menor tamaño y la clase 6 con los más grandes. Se ha estudiado la evolución del pinar y se ha visto que, cada 5 años,

- el 28 % de los árboles de clase 1 pasa a clase 2
- el 31 % de los árboles de clase 2 pasa a clase 3
- el 25 % de los árboles de clase 3 pasa a clase 4
- el 23 % de los árboles de clase 4 pasa a clase 5
- el 37 % de los árboles de clase 5 pasa a clase 6

Además, en cada periodo de 5 años, se talan todos los árboles que había en la clase 6 al inicio del periodo.

Por otra parte, el bosque se regenera de modo que

- por cada 100 árboles de clase 4 nacen 7 árboles
- por cada 100 árboles de clase 5 nacen 15 nuevos árboles
- por cada 100 árboles de clase 6 nacen 20 nuevos árboles.

Es decir, la matriz que proporciona la gestión del bosque en esas condiciones es

$$\begin{pmatrix} 0{,}72 & 0 & 0 & 0{,}07 & 0{,}15 & 0{,}2 \\ 0{,}28 & 0{,}69 & 0 & 0 & 0 & 0 \\ 0 & 0{,}31 & 0{,}75 & 0 & 0 & 0 \\ 0 & 0 & 0{,}25 & 0{,}77 & 0 & 0 \\ 0 & 0 & 0 & 0{,}23 & 0{,}63 & 0 \\ 0 & 0 & 0 & 0 & 0{,}37 & 0 \end{pmatrix}$$

Se supone que inicialmente hay 1000 árboles en cada clase.

a) Indica el número de árboles que habría en cada clase al final del décimo periodo.

b) Si cada árbol talado produce un beneficio de 500 €, estima la ganancia obtenida al cabo de 50 años de gestión (10 periodos).

c) Analiza la composición del bosque al cabo de 100 periodos. ¿Qué observas?

d) Escribe la nueva matriz que representa la gestión del bosque. ¿En qué periodo el bosque supera los 8000 pinos?

e) No nos interesa que el bosque supere los 8000 pinos. Cuando eso ocurre, adi-

Ejercicio 5.6

En un determinado ecosistema se pueden considerar dos zonas diferenciadas: el interior del bosque y la periferia. Al observar una población de garduñas *Martes foina* vemos que, cada año, 1/3 de las que están en el interior del bosque se mueven hacia la periferia, más cerca de las zonas residenciales, mientras que 1/5 de las que habitan en la periferia se mueven hacia el interior del bosque.

Supondremos además que el número total de individuos permanece constante.

Supondremos que en enero de 2023 hay 200 individuos en el interior y 100 en la periferia.

a) Determina la población de garduñas en cada zona a primeros de 2024 y de 2025. (No es necesario efectuar las operaciones: se pueden dejar indicadas).

b) ¿Se aproxima la distribución en zonas a un equilibrio? Indica razonadamente si es así.

Ejercicio 5.7

Papá Noel tiene que diseñar un plan para mantener estable su granja de renos. Comienza en enero de 2023 con 80 renos en una zona suficientemente grande con unas condiciones ambientales ideales y que todas las crías nacen en el mismo momento del año y que, en cualquier grupo de edad, hay la mitad de individuos de cada sexo.

Supondremos la población estructurada en 7 clases de edad (0 a 6 años o más). En la siguiente tabla proporcionamos el número de hembras que hay inicialmente en cada una de las clases de edad, la tasa de tasa de fertilidad de la correspondiente clase y la probabilidad de supervivencia en ese grupo de edad:

Edad	Número de hembras	T. Fertilidad	P. Supervivencia
0	10	0.1	0.5
1	2	0.64	0.8
2	8	0.9	0.9
3	5	0.9	0.9
4	14	0.81	0.9
5	0	0.27	0.9
6	1	0	0

a) (0.5 puntos) Haz una predicción de la estructura de la población de renos en enero de 2024 y enero de 2025, utilizando el modelo de Leslie.

b) (1 punto) Haz una predicción sobre la estructura de la población al cabo de 10 años. ¿Se alcanza una distribución estable de la población?

c) (0.5 puntos) ¿Cuál sería la distribución al cabo de 10 años si Papá Noel hubiera comenzado con 10 hembras en cada una de las clases de 2, 3, 4 y 5 años y ninguna en las clases 0, 1 y 6 ?

Ejercicio 5.8

En cierto bosque se han clasificado los árboles en 5 clases diamétricas. Se ha modelizado la evolución del bosque, su crecimiento, mediante la matriz siguiente:

$$G = \begin{pmatrix} 0{,}2 & 0 & 0 & 0 & 0 \\ 0{,}7 & 0{,}3 & 0 & 0 & 0 \\ 0 & 0{,}5 & 0{,}2 & 0 & 0 \\ 0 & 0 & 0{,}7 & 0{,}1 & 0 \\ 0 & 0 & 0 & 0{,}8 & 0{,}9 \end{pmatrix}$$

Si dejásemos evolucionar el sistema sin intervención, el bosque tendería a desaparecer. Nos planteamos, además de repoblar para mantener constante el número de árboles, talar árboles de la clase superior para obtener beneficio económico.

El coste de la repoblación es de 5 euros por árbol mientras que por cada árbol talado de la clase superior obtenemos 200 euros. Talaremos, siempre que sea posible, 100 árboles de la clase superior. En caso de que haya menos árboles en esa clase talaremos todos los que haya.

Cada etapa consta de tres partes: el crecimiento según la matriz G, la tala de árboles de la 5ª clase (100 o menos si no los hubiera) y la repoblación de tantos árboles como sea necesario para que la cantidad total del bosque permanezca igual a 10000 unidades.

Por ejemplo, si tras una etapa hubiera 2000 árboles en cada clase, se tendría:

Primera parte:

$$\begin{pmatrix} 0{,}2 & 0 & 0 & 0 & 0 \\ 0{,}7 & 0{,}3 & 0 & 0 & 0 \\ 0 & 0{,}5 & 0{,}2 & 0 & 0 \\ 0 & 0 & 0{,}7 & 0{,}1 & 0 \\ 0 & 0 & 0 & 0{,}8 & 0{,}9 \end{pmatrix} \cdot \begin{pmatrix} 2000 \\ 2000 \\ 2000 \\ 2000 \\ 2000 \end{pmatrix} = \begin{pmatrix} 400 \\ 2000 \\ 1400 \\ 1600 \\ 3400 \end{pmatrix}$$

Tras la fase de crecimiento hay en total hay 8800 árboles.

En la segunda parte se extraerían 100 de la 5ª clase.

En la tercera se repoblarían 1300 árboles en la 1ª clase.

Al finalizar la etapa (crecimiento, tala y repoblación) se tendrían 1700 en la 1ª clase, 2000 en la 2ª, 1400 en la 3ª, 1600 en la 4ª y 3300 en la 5ª. En total 10000 árboles. Y se habrá obtenido un beneficio de 13500 euros (200x100 - 5x1300).

Si partimos de nuevo de una situación inicial de 10000 árboles, todos en la 1ª clase o inferior,

1 Escribe una función que calcule la situación de la población tras *n* etapas y el beneficio obtenido.

2 ¿Parecen estabilizarse la población y el beneficio? En caso afirmativo, ¿cuántos árboles se repoblarán en cada etapa?, ¿cuál será el beneficio obtenido por etapa?

Ejercicio 5.9

Una población de ardillas está dividida en tres clases de edades de 2 años de duración. Su evolución está determinada por un modelo de Leslie siendo su matriz

$$\begin{pmatrix} 0 & \alpha & 3 \\ 1/2 & 0 & 0 \\ 0 & 1/4 & 0 \end{pmatrix}$$

1 Interpreta biológicamente cada elemento de la matriz de Leslie dada.

2 Determina el valor de α que hace que el autovalor dominante sea 3/2. Interpreta en el modelo el papel que juega el autovalor dominante.

3 Para el valor de α calculado en el apartado anterior, si a largo plazo el número de hembras es de 800, ¿cuántas de ellas corresponderán a la primera clase de edad?

7. Soluciones a los ejercicios

Solución Ejercicio 5.4

a)

$$L = \begin{pmatrix} 1 & 8 & 0 \\ 1/4 & 0 & 0 \\ 0 & 1/2 & 0 \end{pmatrix}$$

puesto que la matriz tiene dos tasas de fertilidad seguidas distintas de cero, sabes que va a tener un autovalor dominante. También se pueden calcular los autovalores, $|L - \lambda I| = 0 \Longrightarrow \lambda = 0, -1, 2$, y comprobarlo ($\exists \lambda_r$ tal que $|\lambda_r| > |\lambda|$)

b) Calculamos los autovalores de la matriz

$$L = \begin{pmatrix} 1 & 8 & 0 \\ 1/4 & 0 & 0 \\ 0 & 1/2 & 0 \end{pmatrix} \Longrightarrow |L - \lambda I| = -\lambda^3 + \lambda + 2\lambda = 0 \Longleftrightarrow \lambda = 0, -1, 2$$

Puesto que $\lambda = 2 > 1$ es el autovalor dominante, concluimos que el tamaño de la población crecerá $x(k) = d2^k x(0)$ y la proporción de individuos en cada clase sera

$$Lv = 2v \Longrightarrow v = (16/19, 2/19, 1/19)$$

84,21 % en la primera clase, 10,52 % en la segunda y 5,26 % en la última clase

Solución Ejercicio 5.5

En un pinar se han considerado 6 clases arbóreas en función del área basimétrica de cada uno de los pinos. La clase 1 se corresponde con los de menor tamaño y la clase 6 con los más grandes. Se ha estudiado la evolución del pinar y se ha visto que, cada 5 años,

- el 28 % de los árboles de clase 1 pasa a clase 2
- el 31 % de los árboles de clase 2 pasa a clase 3
- el 25 % de los árboles de clase 3 pasa a clase 4
- el 23 % de los árboles de clase 4 pasa a clase 5
- el 37 % de los árboles de clase 5 pasa a clase 6

Además, en cada periodo de 5 años, se talan todos los árboles que había en la clase 6 al inicio del periodo.

Por otra parte, el bosque se regenera de modo que

- por cada 100 árboles de clase 4 nacen 7 árboles
- por cada 100 árboles de clase 5 nacen 15 nuevos árboles
- por cada 100 árboles de clase 6 nacen 20 nuevos árboles.

Es decir, la matriz que proporciona la gestión del bosque en esas condiciones es

$$\begin{pmatrix} 0{,}72 & 0 & 0 & 0{,}07 & 0{,}15 & 0{,}2 \\ 0{,}28 & 0{,}69 & 0 & 0 & 0 & 0 \\ 0 & 0{,}31 & 0{,}75 & 0 & 0 & 0 \\ 0 & 0 & 0{,}25 & 0{,}77 & 0 & 0 \\ 0 & 0 & 0 & 0{,}23 & 0{,}63 & 0 \\ 0 & 0 & 0 & 0 & 0{,}37 & 0 \end{pmatrix}$$

Se supone que inicialmente hay 1000 árboles en cada clase.

a) Indica el número de árboles que habría en cada clase al final del décimo periodo.

Solución.

```
  a=c(0.72,0.28,0,0,0,0,0,0.69,0.31,0,0,0,0,0,0.75,
  0.25,0,0,0.07,0,0,0.77,0.23,0,0.15,
  0,0,0,0.63,0.37,0.2,0,0,0,0,0)
A=matrix(a,6)
A

# el número de árboles en cada clase al final del décimo periodo es elresult
```

```
# multiplicar la matriz A 10 veces por la cantidad inicial

X=matrix(1000,6,1)
X

for (i in 1:10){
   X=A%*%X
}
print(sum(X))
```

Al final del décimo periodo hay 5084 árboles.

b) Si cada árbol talado produce un beneficio de 500 €, estima la ganancia obtenida al cabo de 50 años de gestión (10 periodos).

Solución.

Hay que ver el número de árboles talados y multiplicar esa cantidad por 500. En el enunciado se dice que los árboles que se talan son los que hay en la clase 6 al inicio del periodo. Esto es, en el primer periodo talamos 1000 árboles, en el segundo 370 (que son los que hay en clase 6 al final del primer periodo y, por tanto, al inicio del segundo, y así sucesivamente)

```
    #los árboles que se talan son los de 6ª clase. El total de árboles
talados de la 6ª clase es

X=matrix(1000,6,1)
arboles=1000 #estos se talan al finalizar el 1er periodo y hay que sumar
los que talamos en los 9 restantes
for (i in 1:9){
   X=A%*%X
   arboles=arboles+X[6,1]
}
print(arboles)
beneficio=arboles*500
print(beneficio)
```

El beneficio es de 1781890 euros.

c) Analiza la composición del bosque al cabo de 100 periodos. ¿Qué observas?

```
  X=matrix(1000,6,1)
```

```
X

for (i in 1:100){
   X=A%*%X
}
print(X)
sum(X)

#Al cabo de 100 periodos hay 3114 árboles y habíamos empezado con 6000
#por lo que parece que disminuye el total
#no se pide pero vamos a ver qué pasa tras 200 periodos (1000 años)

X=matrix(1000,6,1)
X

for (i in 1:200){
   X=A%*%X
}
print(X)
sum(X)

#quedan 1802 árboles. Todo apunta a la extinción
```

A partir de ahora supondremos que, en lugar de talar todos los árboles de la sexta clase, se corta solo la mitad.

d) Escribe la nueva matriz que representa la gestión del bosque. ¿En qué periodo el bosque supera los 8000 pinos?

```
    #Como ahora solo se tala la mitad de los árboles quedará la mitad de los
de esta clase
#la matriz nueva será

A[6,6]=0.5
print(A)
```

```
#Vamos a ver en qué periodo se superarían los 8000 pinos

X=matrix(1000,6,1)
periodo=0
while(sum(X)<8000){
   X=A%*%X
   periodo=periodo+1
}
print(periodo)
```

El periodo a partir del cual se superan los 8000 árboles es el periodo 60.

e) No nos interesa que el bosque supere los 8000 pinos. Cuando eso ocurre, adicionalmente a la mitad de los árboles de la clase 6, se propone talar tantos árboles de la clase 6 como sea necesario para mantener el bosque con 8000 pinos. ¿Es factible esta gestión? ¿Con esta nueva gestión se estabilizaría la distribución en clases?

La operación que hacemos es hacer que cuando haya más de 8000 árboles talamos "los que sobran" en la clase 6. Eso se refleja en el interior del bucle siguiente. (El bucle se ejecuta 200 veces para ver qué ocurre en 200 periodos)

```
X=matrix(1000,6,1)
for (i in 1:200){
  tala=0
  X=A%*%X
  if(sum(X)>8000){
    tala=(sum(X)-8000)
  }
  X[6,1]=X[6,1]-tala
  print(X)
  print(sum(X))
}
```

Como nunca aparecen números negativos esa gestión se puede realizar. Si aparecieran números negativos supondría la necesidad de talar más de los árboles que tenemos.

Por otra parte, la distribución de árboles en cada clase se mantiene estable, en torno a estos números:

¿matrix2latex(X)

$$\begin{bmatrix} 1480{,}462 \\ 1337{,}191 \\ 1658{,}117 \\ 1802{,}301 \\ 1120{,}349 \\ 601{,}5789 \end{bmatrix}$$

Solución Ejercicio 5.6

En un determinado ecosistema se pueden considerar dos zonas diferenciadas: el interior del bosque y la periferia. Al observar una población de garduñas *Martes foina* vemos que, cada año, $1/3$ de las que están en el interior del bosque se mueven hacia la periferia, más cerca de las zonas residenciales, mientras que $1/5$ de las que habitan en la periferia se mueven hacia el interior del bosque.

Supondremos además que el número total de individuos permanece constante.

Supondremos que en enero de 2023 hay 200 individuos en el interior y 100 en la periferia.

a) Determina la población de garduñas en cada zona a primeros de 2024 y de 2025. (No es necesario efectuar las operaciones: se pueden dejar indicadas).

Solución.

$x(t)$ =”número de garduñas en el interior en el tiempo t”, y =”número de garduñas en la periferia en el tiempo t”

$$\begin{aligned} x(t+1) &= 2/3\, x(t) + 1/5\, y(t) \\ y(t+1) &= 1/3\, x(t) + 4/5\, y(t) \end{aligned} \iff \begin{pmatrix} x(t+1) \\ y(t+1) \end{pmatrix} = \begin{pmatrix} 2/3 & 1/5 \\ 1/3 & 4/5 \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

Tomando $t = 0$ como 2023, en 2024 se obtiene

$$\begin{pmatrix} x(1) \\ y(1) \end{pmatrix} = \begin{pmatrix} 2/3 & 1/5 \\ 1/3 & 4/5 \end{pmatrix} \begin{pmatrix} x(0) \\ y(0) \end{pmatrix} = \begin{pmatrix} 2/3 & 1/5 \\ 1/3 & 4/5 \end{pmatrix} \begin{pmatrix} 200 \\ 100 \end{pmatrix}$$

Para 2025

$$\begin{pmatrix} x(2) \\ y(2) \end{pmatrix} = \begin{pmatrix} 2/3 & 1/5 \\ 1/3 & 4/5 \end{pmatrix} \begin{pmatrix} x(1) \\ y(1) \end{pmatrix} = \begin{pmatrix} 2/3 & 1/5 \\ 1/3 & 4/5 \end{pmatrix}^2 \begin{pmatrix} 200 \\ 100 \end{pmatrix}$$

b) ¿Se aproxima la distribución en zonas a un equilibrio? Indica razonadamente si es así.

Solución.

Puesto que la matriz $A = \begin{pmatrix} 2/3 & 1/5 \\ 1/3 & 4/5 \end{pmatrix}$ es una matriz de Markov (la suma de los elementos de sus columnas es uno) regular, M tiene todos sus elementos o entradas positivos, entonces la sucesión de matrices M^n tiende a una matriz con todas sus columnas iguales M= $(\vec{c}, \vec{c}, \cdots, \vec{c})$ con $\vec{c} = (c_1, c_2, \cdots, c_)$ tal que $\vec{c}$ es un autovector asociado al autovalor $\lambda = 1$ y $\sum_{i=1}^{n} c_i = 1$

Por tanto, nuestra población a la larga se estabilizará, independientemente de cuál haya sido su distribución inicial, según el vector $\vec{c}$.

Calculamos el vector $\vec{c}$.

$$(A - I) = \begin{pmatrix} -1/3 & 1/5 \\ 1/3 & -1/5 \end{pmatrix} \Longrightarrow E(\lambda = 1) = L\{(3, 5)\} \Longrightarrow \vec{c} = (3/8, 5/8)$$

Concluimos que a la larga 37,5 % de la población de garduñas estará en el interior del bosque y el 62,5 % estará en la periferia.

Solución Ejercicio 5.7

Papá Noel tiene que diseñar un plan para mantener estable su granja de renos. Comienza en enero de 2023 con 80 renos en una zona suficientemente grande con unas condiciones ambientales ideales y que todas las crías nacen en el mismo momento del año y que, en cualquier grupo de edad, hay la mitad de individuos de cada sexo.

Supondremos la población estructurada en 7 clases de edad (0 a 6 años o más). En la siguiente tabla proporcionamos el número de hembras que hay inicialmente en cada una de las clases de edad, la tasa de tasa de fertilidad de la correspondiente clase y la probabilidad de supervivencia en ese grupo de edad:

Edad	Número de hembras	T. Fertilidad	P. Supervivencia
0	10	0.1	0.5
1	2	0.64	0.8
2	8	0.9	0.9
3	5	0.9	0.9
4	14	0.81	0.9
5	0	0.27	0.9
6	1	0	0

a) (0.5 puntos) Haz una predicción de la estructura de la población de renos en enero de 2024 y enero de 2025, utilizando el modelo de Leslie.

b) (1 punto) Haz una predicción sobre la estructura de la población al cabo de 10 años. ¿Se alcanza una distribución estable de la población?

c) (0.5 puntos) ¿Cuál sería la distribución al cabo de 10 años si Papá Noel hubiera comenzado con 10 hembras en cada una de las clases de 2, 3, 4 y 5 años y ninguna en las clases 0, 1 y 6 ?

Solución. La matriz de Leslie asociada al modelo es

$$\begin{bmatrix} 0{,}1 & 0{,}64 & 0{,}9 & 0{,}9 & 0{,}81 & 0{,}27 & 0 \\ 0{,}5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0{,}9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0{,}9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0{,}9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0{,}9 & 0 \end{bmatrix}$$

y se puede generar con la instrucción

```
L=matrix(c(0.1,0.5,0,0,0,0,0,0.64,0,0.8,0,0,0,0,0.9,0,0,0.9,0,0,0,0.9,0,0,0
0,0,0.81,0,0,0,0,0.9,0,0.27,0,0,0,0,0,0.9,0,0,0,0,0,0,0),7)
```

El número incial de hembras viene dado por `X0=matrix(c(10,2,8,5,14,0,1),7,1)` (lo expresamos como matriz para no tener problemas al multiplicar matrices posteriormente)

La población en enero de 2024 viene dada por $X1 = L \cdot X0$. En R la orden es

```
X1=L%*%X0
```

y al ejecutarlo y pedir que lo imprima da como resultado $\begin{bmatrix} 25{,}32 \\ 5 \\ 1{,}6 \\ 7{,}2 \\ 4{,}5 \\ 12{,}6 \\ 0 \end{bmatrix}$ La población en enero de 2025 viene dada por $X2 = L \cdot X1$. En R la orden es

```
X2=L%*%X1
```

y al ejecutarlo y pedir que lo imprima da como resultado $\begin{bmatrix} 20{,}699 \\ 12{,}66 \\ 4 \\ 1{,}44 \\ 6{,}48 \\ 4{,}05 \\ 11{,}34 \end{bmatrix}$

Al cabo de 10 años la población vendrá dada por la expresión $X_{10} = L^{10} \cdot X0$. En R podemos hacerlo con un bucle

```
for (i in 1:10){
  X1=L%*%X0
  X0=X1
  }
print(X1)
```

La población al cabo de 10 años será $\begin{bmatrix} 61{,}15839 \\ 27{,}4082 \\ 19{,}59789 \\ 15{,}70374 \\ 12{,}56258 \\ 10{,}21649 \\ 8{,}457609 \end{bmatrix}$

Para ver si se estabiliza vemos lo que pasa, por ejemplo a los 20 años $\begin{bmatrix} 573{,}1287 \\ 256{,}2357 \\ 183{,}2936 \\ 147{,}5052 \\ 118{,}7045 \\ 95{,}52715 \\ 76{,}87532 \end{bmatrix}$ vemos que la población crece desmesuradamente pero para ver si la proporción entre clases de edad es estable dividimos en cada caso por el total de individuos.

En el caso de 20 años resulta $\begin{bmatrix} 0{,}3949153 \\ 0{,}1765596 \\ 0{,}1262987 \\ 0{,}1016387 \\ 0{,}08179352 \\ 0{,}06582314 \\ 0{,}05297106 \end{bmatrix}$ mientras que en el caso de 10 años resulta $\begin{bmatrix} 0{,}3949152 \\ 0{,}1765596 \\ 0{,}1262987 \\ 0{,}1016386 \\ 0{,}08179352 \\ 0{,}06582318 \\ 0{,}05297109 \end{bmatrix}$ con lo que podemos concluir que se estabiliza la distribución de población

c) Si hubiera sido `X0=matrix(c(0,0,10,10,10,10,0))` Al cabo de 10 años, ejecutando el mismo código anterior, habríamos obtenido esta población:

$$\begin{bmatrix} 45{,}61815 \\ 20{,}22476 \\ 14{,}36129 \\ 11{,}84485 \\ 9{,}951576 \\ 7{,}374695 \\ 5{,}501713 \end{bmatrix}$$

Para ver si hay una distribución estable hacemos lo mismo de antes la distribución al cabo de 10 años es $\begin{bmatrix} 0{,}3971042 \\ 0{,}1760557 \\ 0{,}1250145 \\ 0{,}1031089 \\ 0{,}08662807 \\ 0{,}06419643 \\ 0{,}0478922 \end{bmatrix}$ y la distribución al cabo de 20 años es $\begin{bmatrix} 0{,}3949152 \\ 0{,}1765596 \\ 0{,}1262986 \\ 0{,}1016384 \\ 0{,}08179344 \\ 0{,}06582341 \\ 0{,}05297128 \end{bmatrix}$ por lo que podemos aceptar que la distribución entre grupos de edad se estabiliza.