

Capítulo 2

Teoría de matrices con R

2.1. Introducción

El entorno de programación R es una implementación libre del lenguaje S. Además de ser utilizado para analizar problemas estadísticos, también se puede utilizar para hacer álgebra lineal.

El álgebra lineal es un sistema notacional que permite pensar en conjunto de datos y que se utiliza en multitud de problemas en ciencias e ingeniería. Por un lado se tiene el dato simple y en un nivel superior el conjunto de datos (matrices).

Las matrices fueron empleadas por los Babilonios y Chinos para realizar cálculos básicos y resolver ecuaciones simultáneas. Sin embargo, no fueron introducidas en las matemáticas occidentales hasta principios del siglo XIX. Las matrices pueden ser consideradas como *hojas de cálculo* con álgebra propia.

En las siguientes secciones se verá una pequeña introducción a las matrices, así como algunas aplicaciones de las mismas. También se estudiará, con cierto detalle, los conceptos de valor y vector propio asociados a una matriz, por la gran cantidad de aplicaciones y problemas que se basan en su cálculo.

2.2. Matrices en R

Una matriz es una disposición de objetos dispuestos por filas y columnas y constituye uno de los objetos matemáticos más simples, aunque más potentes, por la cantidad de aplicaciones y problemas de carácter científico donde aparecen.

Aquí tenemos algunos ejemplos de matrices.

$$\begin{pmatrix} 1 & 4 & 3 \\ 2 & 5 & 4 \\ 1 & -3 & -2 \end{pmatrix}, \quad \begin{pmatrix} 1 & 4 & x & 17 \\ 2 & x+y & 7 & -19 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \end{pmatrix}, \quad (x \quad y \quad z \quad w)$$

Cuando se habla de matrices $m \times n$ significa que tiene m filas y n columnas. Así, el primer ejemplo es una matriz cuadrada 3×3 y el segundo una matriz 2×4 (2 filas y 4 columnas). Los dos ejemplos finales tienen una sola columna y

una sola fila y, cuando ocurre esto, se les suele llamar vectores (vector columna y vector fila respectivamente).

A menudo puede interesar representar una matriz $m \times n$ con variables en cada localización; esto se hace como sigue:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{pmatrix}$$

El elemento situado en la fila i , columna j se representa por a_{ij} , donde $1 \leq i \leq m$ y $1 \leq j \leq n$. Algunas veces, cuando no hay dudas sobre las dimensiones de la matriz, esta puede representarse como:

$$(a_{ij}).$$

Como ya se ha comentado en la introducción, una matriz en R es un objeto de dos dimensiones, filas y columnas. La diferencia con un array es que éste puede tener más de dos dimensiones.

Una forma de crear una matriz es mediante la función

`matrix(data=NA, nrow=1, ncol=1, byrow=FALSE, dimnames=NULL)`

Esta función llena, por defecto, la matriz por columnas, primero la primera columna, luego la segunda y así sucesivamente. Para llenar una matriz por filas hay que utilizar el argumento `byrow=TRUE`.

R 1 (Definiendo una matriz mediante `matrix()`).

```
> A = matrix(c(2,3,4,5,6,7), 3, 2)
      [,1] [,2]
[1,]    2    5
[2,]    3    6
[3,]    4    7
}
```

Obviamente, `[3,]` indica que es la tercera fila y `[,2]` que es la segunda columna.

R 2 (Definiendo una matriz mediante `matrix()`).

```
> x <- 0:10
> matrix(x, nrow=2, ncol=5)
      [,1] [,2] [,3] [,4] [,5]
[1,]    0    2    4    6    8
[2,]    1    3    5    7    9
Warning message:
In matrix(x, nrow = 2, ncol = 5) :
data length [11] is not a sub-multiple or multiple of the number of rows
}
```

En el caso anterior, primero definimos el vector \mathbf{x} , que tiene 11 componentes. Después, mediante la función `matrix()` definimos una matriz con esos elementos dispuestos en dos filas y cinco columnas. Si se define una matriz $m \times n$, hay que asegurarse de que el número de entradas en el vector \mathbf{x} sea igual m veces

el número de columnas. Si no lo son, o bien R corta o añade argumentos hasta alcanzar el tamaño deseado. Además, aparece un mensaje que advierte que la longitud de los datos no es un múltiplo del número de filas.

Los argumentos principales de la función `matrix()` son:

data Un objeto de datos opcional (incluyendo una lista o un vector).

nrow Número deseado de filas de la matriz.

ncol Número deseado de columnas.

byrow Expresión lógica. Si es FALSE (por defecto), la matriz se llena por columnas; en otro caso, por filas.

dimnames Si no es nulo (por defecto) es una lista de longitud dos que nos proporciona los nombres de las filas y las columnas, respectivamente. Si no escribimos nada, se toma como NULL.

La función `is.matrix()` devuelve TRUE si lo que se le pasa como argumento tiene un atributo *dimensión* de longitud 2 y FALSE en caso contrario. Además, es importante tener en cuenta que `data.frame` no es una matriz.

A continuación mostramos un ejemplo donde proponemos nombres a las filas y columnas de una matriz.

R 3 (Una matriz con nombres).

```
> x <- 1:6
> mdat <- matrix(x, nrow = 2, ncol = 3, byrow = TRUE,
dimnames = list(c("fila1", "fila2"), c("col1", "col2", "col3")))
```

La función `scan()` permite introducir datos de forma directa e interactiva por el usuario en la ventana de comandos. A continuación se muestra un ejemplo.

R 4 (Introduciendo una matriz con `scan()`).

```
> A <- matrix(scan(), nrow=3)
1: 1 2
3: 3 4
5: 5 6
7:
Read 6 items
> A
[,1] [,2]
[1,] 1 4
[2,] 2 5
[3,] 3 6
```

En este ejemplo se crea un objeto (matriz) con 3 filas y dos columnas introduciendo directamente los elementos de la matriz mediante consola. Se puede comprobar que se trata de un objeto matriz, simplemente tecleando la instrucción `data.class(A)`.

Como ya se ha visto, también se pueden crear matrices utilizando una de las funciones más importantes en R como es `c()`. Esta función nos define el vector con los elementos de la matriz.

R 5 (Introduciendo una matriz con `c()`).

```

> A <- matrix(c(1:12), nrow=3)
> A
[,1] [,2] [,3] [,4]
[1,]  1  4  7 10
[2,]  2  5  8 11
[3,]  3  6  9 12
> B <- matrix(c(1:12), nrow=3, byrow=T)
> B
[,1] [,2] [,3] [,4]
[1,]  1  2  3  4
[2,]  5  6  7  8
[3,]  9 10 11 12
> C <- matrix(c("a", "b", "c", "d", "e", "f", "g", "h"), nrow=2, byrow=T)
> C
[,1] [,2] [,3] [,4]
[1,] "a" "b" "c" "d"
[2,] "e" "f" "g" "h"
> dim(A)
[1] 3 4
> mode(C)
[1] "character"
> length(B)
[1] 12

```

En estos ejemplos se ha visto diversas formas de utilización de la función `c()` para construir matrices. Además, se han utilizado las funciones `dim()`, `length()` y `mode()` que informan sobre la dimensión, longitud o tipo de matriz (numérica, caracteres, ...).

Otra forma de construir matrices es utilizar dos funciones que permiten combinar o encadenar vectores en columnas o filas: `cbind` y `rbind`. La primera (`cbind`) encadena vectores como columnas mientras que la segunda (`rbind`) lo hace como filas.

R 6 (Introduciendo una matriz con `cbind`, `rbind`).

```

> c1 <- c(1,1,1)
> c2 <- c(2,2,2)
> c3 <- c(3,3,3)
> A <- cbind(c1,c2,c3)
> A
c1 c2 c3
[1,] 1 2 3
[2,] 1 2 3
[3,] 1 2 3
> B <- rbind(c1,c2,c3)
> B
[,1] [,2] [,3]
c1  1  1  1
c2  2  2  2
c3  3  3  3

```

Resumimos algunas de las funciones sobre matrices más importantes.

| | |
|-----------|--|
| dim | dimensiones de una matriz. |
| dimnames | devuelve las dimensiones de una matriz. |
| colnames | devuelve el nombre de las columnas. |
| rownames | devuelve el nombre de las filas. |
| mode | devuelve el tipo de datos de los elementos de la matriz. |
| length | devuelve el número total de elementos de la matriz. |
| is.matrix | devuelve T si el objeto es una matriz, F si no lo es. |
| [,] | accede a elementos de la matriz. |
| apply | aplica una función sobre las filas o columnas. |
| cbind | añade una columna a una matriz dada. |
| rbind | añade una fila a una matriz dada. |

R 7.

Funciones para matrices]

```

> A <- matrix(1:10,5,2)
> A
      [,1] [,2]
[1,]    1    6
[2,]    2    7
[3,]    3    8
[4,]    4    9
[5,]    5   10
> dim(A)
[1] 5 2
> length(A)
[1] 10
> nrow(A)
[1] 5
> ncol(A)
[1] 2
> dimnames(A)
NULL

```

2.3. Extrayendo elementos de un vector o una matriz.

Los siguientes comandos o funciones se pueden usar para extraer elementos de un vector

- $x[n]$ Extrae el elemento n -ésimo del vector x .
- $x[-n]$ Extrae todos los elementos excepto el n -ésimo.
- $x[1 : n]$ Extrae del 1 al n -ésimo elemento.
- $x[-(1 : n)]$ Extrae los elementos que no estén entre el 1 y el n -ésimo del.
- $x[c(1, 5, 9)]$ Extrae los elementos 1, 5 y 9 del vector x .
- $x[x > 3]$ Extrae los elementos mayores que 3.
- $x[x > 2 \& x < 5]$ Extrae los elementos mayores que 2 y menores que 5.

R 8 (Extrayendo elementos de un vector).

```

> x <- seq(2,20,by=2)
> x
[1] 2 4 6 8 10 12 14 16 18 20
> x[5]
[1] 10
> x[-5]
[1] 2 4 6 8 12 14 16 18 20
> x[1:5]
[1] 2 4 6 8 10
> x[-(1:5)]
[1] 12 14 16 18 20
> x[c(1,5,9)]
[1] 2 10 18
> x[x>4]
[1] 6 8 10 12 14 16 18 20
> x[x>2 & x<5]
[1] 4

```

Los siguientes comandos o funciones se pueden usar para extraer elementos de una matriz

- $A[i, j]$ Extrae el elemento de la fila i columna j .
- $A[i,]$ Extrae todos los elementos de la fila i .
- $A[, j]$ Extrae todos los elementos de la columna j .
- $A[5 : n,]$ Extrae todos los elementos de la fila 5 a la n .
- $A[, c(1, 4, 5)]$ Extrae los elementos de las columnas 1, 4 y 5 de la matriz A .

R 9 (Extrayendo elementos de una matriz).

```

> A <- matrix(1:20,4,5)
> A
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> A[3,5]
[1] 19
> A[2,]
[1] 2 6 10 14 18
> A[,5]
[1] 17 18 19 20
> A[3:4,]
      [,1] [,2] [,3] [,4] [,5]
[1,]    3    7   11   15   19
[2,]    4    8   12   16   20
> A[,c(1,4,5)]
      [,1] [,2] [,3]
[1,]    1   13   17
[2,]    2   14   18
[3,]    3   15   19
[4,]    4   16   20

```

2.4. Operaciones con matrices en \mathbb{R}

La suma y resta de matrices se hace elemento a elemento. Por ejemplo:

$$\begin{pmatrix} 1 & 3 & 7 \\ 2 & 6 & -4 \\ 2 & 15 & \pi \end{pmatrix} + \begin{pmatrix} 3 & 2 & 1 \\ 5.5 & 3 & -e \\ 2 & 5 & \sqrt{2} \end{pmatrix} = \begin{pmatrix} 4 & 5 & 8 \\ 7.5 & 9 & -4 - e \\ 4 & 20 & \pi + \sqrt{2} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 7 \\ 2 & 6 & -4 \\ 2 & 15 & \pi \end{pmatrix} - \begin{pmatrix} 3 & 2 & 1 \\ 5.5 & 3 & -e \\ 2 & 5 & \sqrt{2} \end{pmatrix} = \begin{pmatrix} -2 & 1 & 6 \\ -3.5 & 3 & e - 4 \\ 0 & 10 & \pi - \sqrt{2} \end{pmatrix}$$

Para darle sentido a la suma o resta de matrices, ambas deben tener el mismo número de filas y de columnas.

Para la multiplicación de matrices, solo se requiere que el número de columnas de la primera matriz sea igual al número de filas de la segunda matriz. En otras palabras, puedes multiplicar una matriz $m \times k$ por otra de tamaño $k \times n$, con la matriz $m \times k$ a la izquierda y la matriz $k \times n$ a la derecha. En los ejemplos, la primera multiplicación se puede realizar, pero la segunda no.

$$\begin{pmatrix} 1 & 3 & 5 \\ 4 & 7 & 2 \\ 9 & 1 & 6 \\ 0 & 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 7 \\ 3 & 7 \\ 7 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 2 & 1 & 3 \end{pmatrix}$$

Al multiplicar una matriz $m \times k$ por otra $k \times n$, el elemento (i, j) resultante, se obtiene multiplicando término a término los elementos de la fila i de la matriz de la izquierda por la columna j de la matriz de la derecha y sumando todos estos productos.

Por ejemplo:

$$\begin{pmatrix} 1 & 3 & 2 \\ 5 & 0 & 7 \\ 6 & 9 & 8 \end{pmatrix} \begin{pmatrix} 4 & 11 \\ 6 & 10 \\ 5 & 9 \end{pmatrix} = \begin{pmatrix} 1 \times 4 + 3 \times 6 + 2 \times 5 & 1 \times 11 + 3 \times 10 + 2 \times 9 \\ 5 \times 4 + 0 \times 6 + 7 \times 5 & 5 \times 11 + 0 \times 10 + 7 \times 9 \\ 6 \times 4 + 9 \times 6 + 8 \times 5 & 6 \times 11 + 9 \times 10 + 8 \times 9 \end{pmatrix} = \begin{pmatrix} 32 & 59 \\ 55 & 118 \\ 118 & 228 \end{pmatrix}$$

En general, si se multiplica una matriz $m \times k$ por otra $k \times n$ se obtiene una matriz de tamaño $m \times n$:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix}$$

Se puede escribir c_{ij} como:

$$c_{ij} = \sum_{p=1}^k a_{ip} b_{pj}.$$

Una propiedad importante es que el producto de matrices, en general, NO cumple la propiedad conmutativa. De hecho, las matrices casi nunca conmutan bajo la multiplicación. Es fácil encontrar ejemplos de matrices A and B donde $MN \neq NM$.

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}; \quad \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

Así pues, en esta operación, el orden es muy importante y es por eso que se habla de multiplicación por la derecha o multiplicación por la izquierda.

En R las principales operaciones con matrices se pueden realizar de la forma siguiente:

R 10 (Generación de matrices).

```
> x <- 1:4
> y <- 5:8
> A <- matrix(x, nrow=2, ncol=2)
> A
      [,1] [,2]
[1,]    1    3
[2,]    2    4

> B <- matrix(y, nrow=2, ncol=2)
> B
      [,1] [,2]
[1,]    5    7
[2,]    6    8
```

Algunas operaciones básicas con matrices son:

- Suma y resta +, −

R 11 (Suma, resta).

```
> C<- A + B
> C
      [,1] [,2]
[1,]    6   10
[2,]    8   12

> D<- B-A
> D
      [,1] [,2]
[1,]    4    4
[2,]    4    4
```

- Producto matricial %*%

R 12 (Producto matricial).

```
> E<- A%*%B
> E
      [,1] [,2]
[1,]   23   31
[2,]   34   46
```


- Transpuesta de una matriz `t()`

R 13 (Transpuesta de una matriz).

```
> t(B)
      [,1] [,2]
[1,]    5    6
[2,]    7    8
```

- Producto cruzado de matrices `crossprod(A,B) = t(A) %*% B`

R 14 (Producto cruzado de matrices).

```
> crossprod(A,B)
      [,1] [,2]
[1,]   17   23
[2,]   39   53
```

- Producto elemento a elemento `*`

R 15 (Producto elemento a elemento).

```
> A * B
      [,1] [,2]
[1,]    5   21
[2,]   12   32
```

- Dimensión de una matriz `dim()`

R 16 (Dimensión de una matriz).

```
> dim(B)
[1] 2 2
```

- Determinante de una matriz `det()`

R 17 (Determinante de una matriz).

```
> det(B)
[1] -2
```

- Inversa de una matriz `solve()`

R 18 (Inversa de una matriz).

```
> solve(B)
      [,1] [,2]
[1,]   -4  3.5
[2,]    3 -2.5

> b<-c(2,3)
> solve(A,b)
[1] 0.5 0.5
```

- Diagonal de la matriz en forma de vector `diag()`

R 19 (Diagonal de la matriz en forma de vector).

```
> diag(B)
[1] 5 8
> x<-c(3,7)
> diag(x)
      [,1] [,2]
[1,]    3    0
[2,]    0    7

> diag(diag(A))
      [,1] [,2]
[1,]    1    0
[2,]    0    4
```

- Suma de los elementos de una matriz `sum()`

R 20 (Suma de los elementos de una matriz).

```
> sum(A)
[1] 10
```

- Suma de las columnas de la matriz `colSums()`

R 21 (Suma de las columnas de la matriz).

```
> colSums(B)
[1] 11 15
```

- Suma de las filas de la matriz `rowSums()`

R 22 (Suma de las filas de la matriz).

```
> rowSums(B)
[1] 12 14
```

- Suma acumulativa de una matriz `cumsum()`

R 23 (Suma acumulativa de una matriz).

```
> cumsum(A)
[1] 1 3 6 10
```

- Producto acumulativa de una matriz `cumsum()`

R 24 (Producto acumulativa de una matriz).

```
> cumprod(A)
[1] 1 2 6 24
```

- Mínimo y máximo valor de una matriz `min()`

R 25 (Producto acumulativa de una matriz).

```
> min(A)
[1] 1
> max(A)
[1] 4
```

- Valores y vectores propios de la matriz `eigen()`

R 26 (Valores y vectores propios de la matriz).

```
> eigen(B)
$values
[1] 13.1520673 -0.1520673

$vectors
      [,1]      [,2]
[1,] -0.6514625 -0.8053759
[2,] -0.7586809  0.5927644
```

2.5. Decisiones de compra basadas en matrices

Existen en la vida grandes decisiones, como la compra de una vivienda, de un automóvil, de una motocicleta, etc, que pueden realizarse de una forma más técnica de lo que se piensa. En otras palabras, se puede *matematizar* decisiones de compra simplemente utilizando las propiedades básicas que ofrecen las matrices y vectores. Veamos un caso concreto donde se pone de manifiesto lo que hemos comentado.

Se parte del planteamiento inicial de un problema: la compra de una motocicleta. Realmente no parece que esto suponga un problema para aquellas personas decididas y poco dubitativas; sin embargo, existe un buen número de personas que, ante el hecho de tener que decidir por un determinado artículo cuando tienes la posibilidad de elegir entre una gama más o menos extensa del mismo, sufren una serie de indecisiones y cambios repentinos de opinión que hacen de su decisión un suplicio.

Ante esta elección, la primera idea que se nos ocurre es la de comprar revistas, ver vídeos publicitarios, anuncios, preguntar en las tiendas, ver fotos, etc. Pero todo esto resulta un tanto desordenado. Parece claro que lo ideal quizás sería tratar el asunto desde el punto de vista numérico. Se puede tener en cuenta aquellos factores o características que se consideren importantes en la elección de una moto y tratar de asignarles valores cuantitativos, de manera que ayude a establecer una clasificación. En dicha clasificación, el modelo que ocupara la posición con una mayor puntuación sería la que en su conjunto presentara unas mejores características de acuerdo con los gustos particulares.

Se va a dividir el problema de la compra de una motocicleta, en tres fases:

Fase I Condiciones, limitaciones y planteamiento del problema.

Fase II Recogida de datos y cuantificación del problema.

Fase III Resolución numérica.

Fase I

El problema es la elección de una moto de entre una serie de modelos a la venta en el mercado. El tipo de moto elegida es la scouter y su cilindrada será al menos de 90 centímetros cúbicos. El objetivo es el de establecer una clasificación numérica de los modelos que más se acercan a nuestras necesidades y nuestros gustos. Antes de plantear un posible método de resolución siguiendo este esquema, surgen ciertas dudas sobre la posibilidad de establecer unos criterios comunes determinados para todas las personas que se enfrenten ante el mismo problema. Por ejemplo,

- ¿Qué factores o características de una moto se consideran más importantes a la hora de la elección?
- ¿Se consideran todas las características igualmente importantes?
- ¿Qué ocurre si los factores que uno considera importantes no coinciden con los del resto de la gente?
- ¿Cómo se puede cuantificar una característica cualitativa para establecer un orden numérico?

Estas y algunas otras cuestiones surgen al reflexionar sobre este problema. Aunque parece ciertamente complejo, se puede desarrollar un método de resolución.

La idea sobre la que se basa el método es considerar una matriz de datos con los que vamos a trabajar. Por un lado se tendrá un determinado número de modelos de motocicletas y por otro lado un conjunto de características que se deben cuantificar. El problema de la cuantificación será tratado posteriormente con mayor detenimiento.

Una vez se disponga de esa matriz de datos, se establecerán los factores de corrección de cada característica. Con todo esto, mediante el producto de matrices se obtendrá la solución final. Los factores de corrección tienen la función de ponderar las distintas características asociados a los diferentes modelos analizados.

Fase II Toma de datos _____

Se tiene que elegir entre una de las motos de la siguiente lista:

| | | |
|------------------------|----------------------|-----------------------|
| 1 Aprilia Leonardo 125 | 2 Honda 125 Spacy | 3 Honda Yupy 90 |
| 4 Honda 100 Bali | 5 Honda SH100 | 6 Honda Spazio 10 |
| 7 Kymco Heroism 125 | 8 Kymco Movie 125 | 9 Italjet Formula 125 |
| 10 PGO Galaxy 90 | 11 Peugeot 125 | 12 Piaggio Typhoon |
| 13 Piaggio Sfera 125 | 14 Piaggio Vespa 125 | 15 Suzuki Vecstar |

De estas 15 motocicletas hay que establecer aquéllas características que se consideren más importantes y que influirán en la toma de la decisión. En este caso se consideraron y analizaron las siguientes características:

C_1 = Cilindrada; C_2 =Potencia; C_3 =Peso; C_4 =Depósito;
 C_5 =Frenada C_6 =Estética C_7 =Precio.

Una vez seleccionadas, se da una puntuación a cada una de los modelos respecto de la característica que se está estudiando. Es decir, si se está estudiando, por ejemplo, la característica **frenada**, se tiene que establecer una puntuación a cada motocicleta, donde la máxima puntuación será para la moto que mejor frenada posea, y así en orden decreciente.

Con esto, se establece una matriz de datos, donde cada fila representa un modelo de motocicleta y cada columna representa cada una de las características que se analizan.

Todas estas características no son igualmente importantes en la toma de la decisión. Así, por ejemplo, el factor estético es muy importante y pasa por encima de otros muchos. Esto se hace mediante un vector de ponderación, que establece la importancia de cada factor o característica.

El vector de ponderación o factor de importancia que se asignó a cada una de las características anteriores fue:

Cilindrada = 1.25; Potencia = 1.35; Peso = 1.1; Depósito = 1;
 Frenada = 1.5; Estética = 1.65; Precio = 2.

La tabla 2.1 muestra las puntuaciones alcanzadas por cada motocicleta respecto a las características estudiadas. Hay que hacer notar que las puntuaciones son sobre 10.

| modelo | C_1 | C_2 | C_3 | C_4 | C_5 | C_6 | C_7 |
|-----------------------|-------|-------|-------|-------|-------|-------|-------|
| Aprilia Leonardo | 4 | 6 | 6 | 5 | 7 | 4 | 3 |
| Honda CH 125 | 4 | 3 | 5 | 3 | 1 | 2 | 4 |
| Honda Yupy NH 90 | 1 | 1 | 2 | 2 | 4 | 1 | 6 |
| Honda SJ 100 Bali | 2 | 2 | 2 | 2 | 4 | 2 | 6 |
| Honda Scoopy | 3 | 2 | 2 | 1 | 5 | 1 | 6 |
| Honda CN 250 Spazio | 5 | 7 | 7 | 6 | 4 | 4 | 1 |
| Kymco Heroism 125 | 4 | 3 | 4 | 1 | 3 | 4 | 5 |
| Kymco Movie 125 | 4 | 3 | 4 | 2 | 3 | 3 | 5 |
| Italjet Formula 125 | 4 | 5 | 4 | 5 | 6 | 5 | 2 |
| PGO Galaxy 90 | 1 | 1 | 1 | 1 | 2 | 2 | 7 |
| Peugeot SV 125 L | 4 | 3 | 2 | 4 | 4 | 2 | 6 |
| Piaggio Typhoom 125 | 4 | 4 | 2 | 3 | 1 | 3 | 6 |
| Piaggio Sfera 125 | 4 | 4 | 3 | 3 | 5 | 2 | 5 |
| Piaggio Vespa 125 | 4 | 4 | 4 | 4 | 5 | 1 | 4 |
| Suzuki AN 125 Vecstar | 4 | 4 | 3 | 3 | 2 | 4 | 4 |

Tabla 2.1: Puntuaciones de cada motocicleta por característica evaluada.

Fase III Resolución numérica

En consecuencia la matriz de datos es

$$M_d = \begin{bmatrix} 4 & 6 & 6 & 5 & 7 & 4 & 3 \\ 4 & 3 & 5 & 3 & 1 & 2 & 4 \\ 1 & 1 & 2 & 2 & 4 & 1 & 6 \\ 2 & 2 & 2 & 2 & 4 & 2 & 6 \\ 3 & 2 & 2 & 1 & 5 & 1 & 6 \\ 5 & 7 & 7 & 6 & 4 & 4 & 1 \\ 4 & 3 & 4 & 1 & 3 & 4 & 5 \\ 4 & 3 & 4 & 2 & 3 & 3 & 5 \\ 4 & 5 & 4 & 5 & 6 & 5 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 7 \\ 4 & 3 & 2 & 4 & 4 & 2 & 6 \\ 4 & 4 & 2 & 3 & 1 & 3 & 6 \\ 4 & 4 & 3 & 3 & 5 & 2 & 5 \\ 4 & 4 & 4 & 4 & 5 & 1 & 4 \\ 4 & 4 & 3 & 3 & 2 & 4 & 4 \end{bmatrix}$$

El vector con el factor multiplicativo es \mathbf{v} ,

$$[1.25 \quad 1.35 \quad 1.10 \quad 1.00 \quad 1.50 \quad 1.65 \quad 2.00]^T$$

Por lo que si multiplicamos $M_d \cdot v$ tendremos

$$\begin{bmatrix} 47.80 \\ 30.35 \\ 26.45 \\ 30.70 \\ 30.80 \\ 44.00 \\ 35.55 \\ 34.90 \\ 42.40 \\ 25.00 \\ 36.55 \\ 34.05 \\ 37.50 \\ 35.95 \\ 34.30 \end{bmatrix}$$

Este vector representa las puntuaciones obtenidas por cada modelo. Se observa que queda como ganador el modelo primero con 47.80 puntos, lo que significa que esa sería la mejor elección.

Ejercicio 1 (Decisión de compra).

Imagina que debes comprar un automóvil para trasladarte a la universidad. Se trata de una decisión de compra a la que podemos aplicar el modelo matricial estudiado en este capítulo.

Supongamos que tu presupuesto está limitado a 10.000 euros. Realiza los siguientes pasos para cuantificar la decisión de compra.

1. Obtén una lista con modelos del mercado que se ajustan a tu presupuesto.
2. Determina un listado de características de los diferentes modelos que vas a tener en cuenta a la hora de tomar la decisión. Determina la importancia que le asignas a cada característica.
3. Toma de datos, basándote en las informaciones que consideres oportunas: revistas especializadas, páginas web, etc.
4. Calcula numéricamente la resolución del problema, utilizando R.

2.6. Resolución de sistemas con R

El problema de la resolución de sistemas con R es muy sencillo, ya que R dispone de una función `solve()`, que permite resolver la ecuación $A\Delta x = b$, para x , donde b puede ser un vector o una matriz.

Así, si se tiene en una matriz A los coeficientes del sistema de ecuaciones lineales y en b el conjunto de los términos independientes del sistema, la instrucción

`solve(A,b)`

devuelve un vector solución del sistemas de ecuaciones.

Por ejemplo:

R 27 (Resolución de un sistema).

```
> A <- matrix(c(1:9), nrow=3)
> b <- matrix(c(1,1,1),nrow=3)
> solve(A,b)
Error in solve.default(A, b) :
Lapack routine dgesv: system is exactly singular: U[3,3] = 0
```

En el ejemplo mostrado anteriormente, no se obtiene ningún vector como resultado, sino un mensaje de error que nos indica que la matriz de coeficientes es singular, por lo que no existe solución.

Se quiere resolver el sistema de ecuaciones:

$$\begin{aligned}x + 2y - z &= 1 \\ -2x + y + 2z &= 1 \\ x + y + z &= 1\end{aligned}$$

En R se procede de la siguiente forma:

R 28 (Resolución de un sistema).

```
> A <- matrix(c(1,-2,1,2,1,1,-1,2,1), nrow=3)
> b <- matrix(c(1,1,1),nrow=3)
> solve(A,b)
[,1]
[1,] 0.1
[2,] 0.6
[3,] 0.3
```

Véase un problema de ejemplo que se puede resolver fácilmente utilizando los comandos de R adecuados.

Ejercicio 2.

Una florista vende tres tipos de ramos de flores formados por margaritas, claveles y rosas. Los ramos pequeños están formados por una margarita, 3 claveles y 3 rosas. Los ramos medianos contienen 2 margaritas, 4 claveles y 6 rosas. Finalmente, los ramos grandes constan de 4 margaritas, 8 claveles y 6 rosas. Si la florista dispone de 35 margaritas, 75 claveles y 69 rosas, ¿cuántos ramos de cada tipo podrá hacer?

2.7. El problema de los valores y vectores propios

En esta sección se trata brevemente, sin entrar en demasiados detalles, una cuestión fundamental del álgebra lineal como es el cálculo de los valores y vectores propios de una matriz.

Dado el vector $\mathbf{u} = (6, 2)$ y el escalar $\alpha = 7$ si se hace el producto

$$7 \cdot \mathbf{u} = 7 \cdot \begin{bmatrix} 6 \\ 2 \end{bmatrix} = \begin{bmatrix} 42 \\ 14 \end{bmatrix}$$

lo que se obtiene es otro vector de la misma dirección, que constituye una transformación lineal del vector \mathbf{u} .

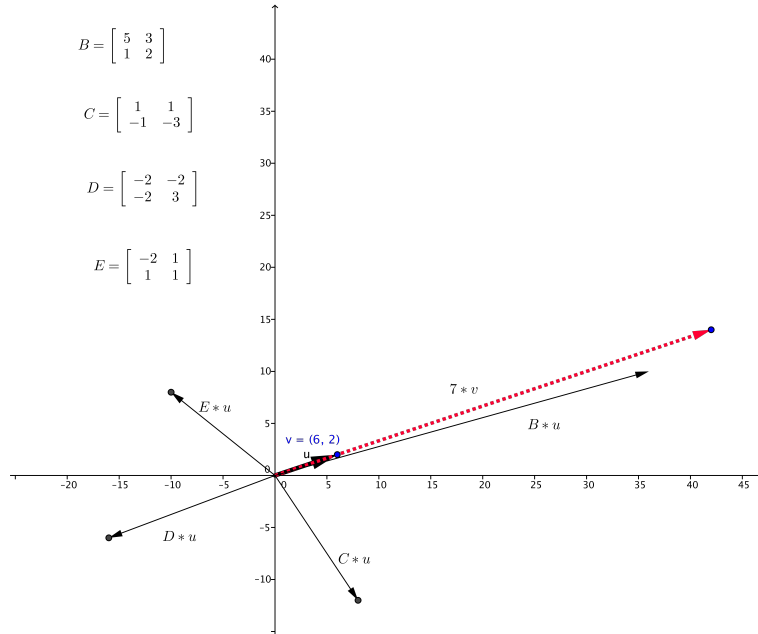


Figura 2.1: El producto de una matriz por un vector.

Dadas las matrices

$$B = \begin{bmatrix} 5 & 3 \\ 1 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \\ -1 & -3 \end{bmatrix}, \quad D = \begin{bmatrix} -2 & -2 \\ -2 & 3 \end{bmatrix}, \quad E = \begin{bmatrix} -2 & 1 \\ 1 & 1 \end{bmatrix}.$$

En la figura 2.1 se observa el resultado geométrico de multiplicar cada una de las matrices anteriores por el vector $\mathbf{u} = (6, 2)$. Al multiplicar una matriz por un vector obtenemos un nuevo vector que cambia de dirección. En los casos anteriores, el producto matriz-vector no era una transformación lineal del vector puesto que cambiaba la dirección del mismo.

Dada otra matriz

$$M = \begin{bmatrix} 5 & 6 \\ 1 & 4 \end{bmatrix},$$

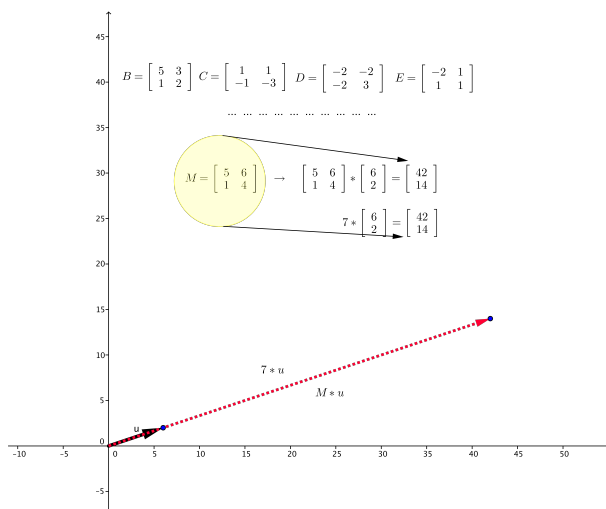


Figura 2.2: Representación geométrica del concepto de valor y vector propio.

resulta que

$$M \cdot \mathbf{u} = \begin{bmatrix} 5 & 6 \\ 1 & 4 \end{bmatrix} \cdot \begin{bmatrix} 6 \\ 2 \end{bmatrix} = 7 \cdot \begin{bmatrix} 6 \\ 2 \end{bmatrix},$$

como se aprecia en la figura 2.2.

Este ejemplo muestra que se ha encontrado un vector, $\mathbf{u} = (6, 2)$, un escalar 7 y una matriz cuadrada M tales que

$$M \cdot \mathbf{u} = 7 \cdot \mathbf{u}.$$

Entonces se dice que 7 es un valor propio asociado a M , mientras que el vector $\mathbf{u} = (6, 2)$ es un vector propio asociado a M .

Por poner un ejemplo real donde se aplica el concepto de valor propio, pensemos que a medida que la Tierra rota, los vectores en el eje de rotación permanecen invariantes. Si se considera la transformación lineal que sufre la Tierra tras una hora de rotación, una flecha que partiera del centro de la Tierra al polo Sur geográfico sería un vector propio de esta transformación, pero una flecha que partiera del centro a un punto del ecuador no sería un vector propio. Dado que la flecha que apunta al polo no cambia de longitud por la rotación, su valor propio es 1.

La idea geométrica de valor propio introduce el concepto formal de valor y vector propio.

Definición 1.

Valores y vectores propios. Dada una matriz A cuadrada de dimensión n , existen como máximo n escalares λ y vectores \mathbf{x} tales que

$$A\mathbf{x} = \lambda\mathbf{x}. \quad (2.1)$$

A los escalares que verifican esta ecuación se les llama **valores propios** de la matriz A , mientras que los vectores no nulos \mathbf{x} , que satisfacen la ecuación reciben el nombre de **vectores propios** asociados al valor propio λ .

De la propia definición (2.1) se tiene que

$$A\mathbf{x} = \lambda\mathbf{x} \quad \Leftrightarrow \quad (A - \lambda I)\mathbf{x} = 0,$$

lo que significa que la matriz $A - \lambda I$ es singular. Esto significa que los valores propios λ son los valores que hacen nulo el determinante de la matriz $A - \lambda I$. En otras palabras, los valores propios de A son las raíces de la ecuación

$$\det(A - \lambda I) = 0.$$

Definición 2.

Polinomio característico de A . El polinomio

$$\det(A - \lambda I) = b_0 + b_1\lambda + b_2\lambda^2 + \dots + b_n\lambda^n$$

se llama **polinomio característico** de A y se denota por $p_A(\lambda)$, mientras que la ecuación $\det(A - \lambda I) = 0$ se conoce como **ecuación característica**.

Cálculo de los valores propios de A .

Se sabe que λ es un valor propio de A si $A\mathbf{x} = \lambda\mathbf{x}$, por lo que para calcular los valores propios de una matriz A basta resolver el determinante

$$\det(A - \lambda I) = 0.$$

Esto significa que, aunque A tiene n valores propios puesto que son las raíces del polinomio característico, algunas pueden ser complejas (aunque las entradas de A sean números reales) y algunos pueden estar repetidos, como ocurre con las raíces de cualquier polinomio. Si A es de números reales, entonces sus valores propios complejos se dan por pares conjugados.

Las transformaciones lineales del espacio, como la rotación, la reflexión, la dilatación o cualquier combinación de las anteriores, pueden interpretarse mediante el efecto que producen en los vectores. Los vectores pueden visualizarse como flechas de una cierta longitud apuntando en una dirección y sentido determinados.

Los vectores propios de las transformaciones lineales son vectores que, o no son afectados por la transformación o sólo resultan multiplicados por un escalar; y, por tanto, no varían su dirección.

- El valor propio de un vector propio es el factor de escala por el que ha sido multiplicado.
- Un espacio propio es un espacio formado por todos los vectores propios del mismo valor propio, además del vector nulo, que no es un vector propio.
- La multiplicidad geométrica de un valor propio es la dimensión del espacio propio asociado.
- El espectro de una transformación en espacios vectoriales finitos es el conjunto de todos sus valores propios.
- Por ejemplo, un vector propio de una rotación en tres dimensiones es un vector situado en el eje de rotación sobre el cual se realiza la rotación. El valor propio correspondiente es 1 y el espacio propio es el eje de giro. Como es un espacio de una dimensión, su multiplicidad geométrica es uno. Es el único valor propio del espectro (de esta rotación) que es un número real.

Valores y vectores propios en R

La función que proporciona los valores y vectores propios de una matriz A cuadrada es `eigen()`, de la forma

```
eigen(x, symmetric, only.values = FALSE, EISPACK = FALSE)
```

cuyos principales argumentos son los siguientes:

x Una matriz numérica o compleja.

symmetric Si es TRUE, se supone una matriz simétrica.

only.values Si es TRUE, solo los valores propios son calculados.

La función `eigen()` devuelve una lista con dos componentes: *values* y *vectors*. *values* es un vector conteniendo los k valores propios de x , en orden decreciente de acuerdo con el módulo. *Vectors* es una matriz de tamaño $k \times k$ cuyas columnas contienen los vectores propios de x , normalizados respecto de la unidad.

Ejemplo desarrollado con R.

R 29 (Cálculo de valores y vectores propios).

```

> A <- matrix(c(1:9), nrow=3)
> A
[,1] [,2] [,3]
[1,]  1   4   7
[2,]  2   5   8
[3,]  3   6   9
> eigen(A)
$values
[1] 1.611684e+01 -1.116844e+00 -5.700691e-16

$vectors
[,1] [,2] [,3]
[1,] -0.4645473 -0.8829060 0.4082483
[2,] -0.5707955 -0.2395204 -0.8164966
[3,] -0.6770438 0.4038651 0.4082483

```

La función que se utiliza en R para el cálculo de los valores y vectores propios está basada en ciertas rutinas de la librería LAPACK.

Ejercicio 3.

Calcula los valores y vectores propios de la matriz

$$A = \frac{1}{9} = \begin{bmatrix} 7 & -2 & 0 \\ -2 & 6 & 2 \\ 0 & 2 & 5 \end{bmatrix}$$

Encuentra la solución general de la ecuación

$$\mathbf{x}_{k+1} = A\mathbf{x}, \quad \text{si } \mathbf{x}_0 = \begin{bmatrix} 1 \\ 11 \\ -2 \end{bmatrix}$$

¿Se observa algún comportamiento en la secuencia de soluciones cuando k aumenta?

2.8. La factorización y algoritmo QR

La factorización QR de una matriz es una descomposición de la misma como producto de una matriz ortogonal por una triangular superior. La descomposición QR es la base del algoritmo QR utilizado para el cálculo de los vectores y valores propios de una matriz.

Se va a calcular la factorización QR mediante el método de ortogonalización de Gram-Schmidt. El método de ortogonalización de Gram-Schmidt construye una base ortonormal de vectores en un espacio vectorial V de dimensión n a

partir de una base cualquiera de vectores. Pierre-Simon Laplace ya estudió un algoritmo de este tipo en el siglo XVIII, aunque el algoritmo que se utiliza en la actualidad se debe a los matemáticos del siglo XIX Jorgen Gram y Erhard Schmidt. Dicho algoritmo fue uno de los primeros en el campo del álgebra lineal computacional.

Inicialmente se escribe la matriz A por columnas, $A = (\mathbf{a}_1 | \mathbf{a}_2 | \cdots | \mathbf{a}_n)$. Entonces, si aplicamos a los vectores columna de A el algoritmo de Gram-Schmidt.

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{a}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\ \mathbf{u}_2 &= \mathbf{a}_2 - \text{proj}_{\mathbf{e}_1} \mathbf{a}_2, & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\ \mathbf{u}_3 &= \mathbf{a}_3 - \text{proj}_{\mathbf{e}_1} \mathbf{a}_3 - \text{proj}_{\mathbf{e}_2} \mathbf{a}_3, & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\ &\vdots & & \\ \mathbf{u}_k &= \mathbf{a}_k - \sum_{j=1}^{k-1} \text{proj}_{\mathbf{e}_j} \mathbf{a}_k, & \mathbf{e}_k &= \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|} \end{aligned}$$

Despejando \mathbf{a}_j se tiene

$$\begin{aligned} \mathbf{a}_1 &= \mathbf{e}_1 \|\mathbf{u}_1\| \\ \mathbf{a}_2 &= \text{proj}_{\mathbf{e}_1} \mathbf{a}_2 + \mathbf{e}_2 \|\mathbf{u}_2\| \\ \mathbf{a}_3 &= \text{proj}_{\mathbf{e}_1} \mathbf{a}_3 - \text{proj}_{\mathbf{e}_2} \mathbf{a}_3 + \mathbf{e}_3 \|\mathbf{u}_3\| \\ &\vdots \\ \mathbf{a}_k &= \sum_{j=1}^{k-1} \text{proj}_{\mathbf{e}_j} \mathbf{a}_k + \mathbf{e}_k \|\mathbf{u}_k\| \end{aligned}$$

Se puede escribir matricialmente estas ecuaciones lineales.

$$(\mathbf{e}_1 | \mathbf{e}_2 | \cdots | \mathbf{e}_n) \begin{bmatrix} \|\mathbf{u}_1\| & \mathbf{e}_1 \cdot \mathbf{a}_2 & \mathbf{e}_1 \cdot \mathbf{a}_3 & \cdots \\ 0 & \|\mathbf{u}_2\| & \mathbf{e}_2 \cdot \mathbf{a}_3 & \cdots \\ 0 & 0 & \|\mathbf{u}_2\| & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Este producto de matrices es igual a la matriz A con la que se inicia la factorización. De esta forma, se puede escribir

$$A = QR,$$

donde $Q = (\mathbf{e}_1 | \mathbf{e}_2 | \cdots | \mathbf{e}_n)$ es una matriz ortogonal, ya que está formada por los vectores \mathbf{e}_j , para $j = 1, \dots, n$ obtenidos mediante el algoritmo de Gram-Schmidt. La matriz R

$$R = \begin{bmatrix} \|\mathbf{u}_1\| & \mathbf{e}_1 \cdot \mathbf{a}_2 & \mathbf{e}_1 \cdot \mathbf{a}_3 & \cdots \\ 0 & \|\mathbf{u}_2\| & \mathbf{e}_2 \cdot \mathbf{a}_3 & \cdots \\ 0 & 0 & \|\mathbf{u}_2\| & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

es triangular superior.

Nótese que se tiene otra forma de calcular la matriz R como

$$R = Q^T A.$$

⊙ Factorización QR para matrices simétricas con \mathbf{R}

Dada una matriz simétrica A , es posible utilizar la factorización QR para determinar sus valores y vectores propios.

Los pasos son los siguientes:

- Se hace $A_1 = A$.
- Para $i = 1$, se calcula la descomposición QR de A_1 , es decir, $A_1 = Q_1 R_1$ y hacemos $A_2 = R_1 Q_1$.
- Para $i = 2$, se calcula la descomposición QR de A_2 , es decir, $A_2 = Q_2 R_2$ y hacemos $A_3 = R_2 Q_2$.
- Para $i = 3$, se calcula la descomposición QR de A_3 , es decir, $A_3 = Q_3 R_3$ y hacemos $A_4 = R_3 Q_3$.
- Para $i = 4, 5, \dots$ se sigue repitiendo los mismos cálculos que en los pasos anteriores, hasta que se produzca la convergencia.

Mediante estos pasos, se obtienen dos secuencias de matrices

$$A_n = (A_1, A_2, A_3, \dots, A_n, A_{n+1}, \dots)$$

$$Q_n = (Q_1, Q_2, Q_3, \dots, Q_n, Q_{n+1}, \dots).$$

La secuencia A_n converge a una matriz diagonal D donde aparecen los valores propios de A en la diagonal. En cuanto a los vectores propios, éstos son las columnas de la matriz

$$S = \prod_i Q_i.$$

A continuación se muestra un código muy simple en R que muestra los pasos anteriores.

R 30 (Código simple de una factorización QR).

```
# se crea una matriz simétrica
A <- matrix( sample(1:30,16), ncol=4)
A <- A + t(A);

# inicialicion
X <- A;
pQ <- diag(1, dim(A)[1]);

# iteraciones
for(i in 1:30)
{
  d <- qr(X);
  Q <- qr.Q(d);
  pQ <- pQ %*% Q;
  X <- qr.R(d) %*% Q;
}
```

Después de generar este código, se visualizan las matrices X y pQ para determinar los valores propios (diagonal de X) y los vectores propios (columnas de pQ).

Nótese que se ha generado la matriz mediante la instrucción:

```
sample(x, size, replace = FALSE, prob = NULL)
```

Normalmente x constituye un conjunto de números entre los que escoger y $size$ es el tamaño de la muestra escogida. En el ejemplo, se generan 16 elementos de entre 1 y 30. A su vez se disponen en una matriz de 4 columnas.

Una vez construida A , se puede determinar una matriz simétrica de una manera sencilla mediante la operación $A + A^t$, a la que se vuelve a llamar A . En el siguiente bloque se inicializa X y pQ , como una matriz diagonal de unos en la misma.

El último bloque de la función aplica el algoritmo QR estudiado anteriormente. Para ello, utiliza la función **qr** y las diversas variantes que tiene implementado R. Por defecto, para matrices reales utiliza la librería LINPACK, aunque se puede hacer que use LAPACK (incluyendo en los argumentos LAPACK=TRUE). LINPACK está restringido a matrices con menos de 2^{31} elementos. Las diversas funciones relacionadas con la factorización que se utilizan en el código son:

- **qr** calcula la descomposición QR de una matriz. El argumento principal debe ser una matriz numérica.
- **qr.Q** devuelve la matriz original Q desde la cual se construyó el objeto o los componentes de la descomposición.
- **qr.R** devuelve la matriz original R desde la cual se construyó el objeto o los componentes de la descomposición.

A partir de estas funciones se puede comprender mejor la simpleza del código.

R 31 (Ejemplo de una factorización QR).

```
> A <- matrix(sample(0:1,16,replace=TRUE), ncol=4)
> A<- A+t(A)
> A
      [,1] [,2] [,3] [,4]
[1,]    2    1    2    1
[2,]    1    0    1    2
[3,]    2    1    2    2
[4,]    1    2    2    2
> X<- A
> pQ <- diag(1, dim(A)[1])
> for(i in 1:60)
+ {
+   d <- qr(X);
+   Q <- qr.Q(d);
+   pQ <- pQ %*% Q;
+   X <- qr.R(d) %*% Q;
+ }
> X
      [,1] [,2] [,3] [,4]
[1,] 6.219682e+00 -2.380724e-16 -1.579337e-15 2.979779e-16
[2,] 3.876280e-40 -1.345213e+00 -7.278536e-03 2.472861e-17
[3,] -1.919798e-42 -7.278536e-03 1.223237e+00 2.029689e-16
[4,] -9.831037e-108 -5.194473e-68 9.370655e-66 -9.770487e-02
> pQ
      [,1] [,2] [,3] [,4]
[1,] 0.4875893 -0.1996191 0.6979472 -0.4850552
[2,] 0.3494405 0.7960344 -0.2930128 -0.3979498
[3,] 0.5767476 0.2007975 0.1889389 0.7689894
[4,] 0.5545360 -0.5349409 -0.6255520 -0.1225263
```

Parece interesante comparar con los valores y vectores propios que obtenemos cuando utilizamos el comando habitual *eigen*.

R 32 (Ejemplo de una factorización QR).

```
> eigen(A)
$values
[1] 6.21968174 1.22325719 -0.09770487 -1.34523406

$vectors
      [,1]      [,2]      [,3]      [,4]
[1,] -0.4875893 -0.6985101  0.4850552  0.1976405
[2,] -0.3494405  0.2952674  0.3979498 -0.7952009
[3,] -0.5767476 -0.1883691 -0.7689894 -0.2013321
[4,] -0.5545360  0.6240336  0.1225263  0.5367115
```

Se observa la similitud en las soluciones obtenidas.

2.9. El método de Lanczos para el cálculo de valores y vectores propios

El algoritmo de Lanczos para matrices A simétricas calcula, mediante un proceso iterativo, los valores y vectores propios de matrices grandes y dispersas. Constituye el algoritmo más eficiente, desde el punto de vista computacional, cuando se trabaja con matrices grandes, simétricas y con pocos elementos distintos de cero.

El n -ésimo paso del algoritmo transforma la matriz A inicial, de dimensión n en una matriz T tridiagonal tal que esta matriz es semejante a A . Por definición de matrices semejantes, se puede afirmar que existe una matriz V ortonormal tal que se verifica que

$$T = V^T A V, \quad \longrightarrow \quad V T = A V.$$

Sea dicha matriz T de la forma

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ & & & & \beta_{n-1} & \alpha_n \end{bmatrix}, \quad V = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n).$$

La ecuación $T = V^T A V$ puede escribirse como $AV = VT$, que matricialmente es:

$$A(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n) = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n) \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ & & & & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

Desarrollando esta expresión,

$$\begin{aligned} A\mathbf{v}_1 &= \alpha_1\mathbf{v}_1 + \beta_1\mathbf{v}_2 \\ A\mathbf{v}_2 &= \beta_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \beta_2\mathbf{v}_3 \\ A\mathbf{v}_3 &= \beta_2\mathbf{v}_2 + \alpha_3\mathbf{v}_3 + \beta_3\mathbf{v}_4 \\ &\vdots \end{aligned}$$

es decir,

$$A\mathbf{v}_j = \beta_{j-1}\mathbf{v}_{j-1} + \alpha_j\mathbf{v}_j + \beta_j\mathbf{v}_{j+1}, \quad (2.2)$$

para $j = 1, 2, \dots, n-1$.

Suponiendo que $\beta_0 = \mathbf{v}_0 = 0$, entonces

$$\begin{aligned} \mathbf{v}_j^T A\mathbf{v}_j &= \beta_{j-1}\mathbf{v}_j^T \mathbf{v}_{j-1} + \alpha_j\mathbf{v}_j^T \mathbf{v}_j + \beta_j\mathbf{v}_j^T \mathbf{v}_{j+1} \\ &= 0 + \alpha_j\|\mathbf{v}_j\| + 0 \\ &= \alpha_j, \end{aligned}$$

luego

$$\alpha_j = \mathbf{v}_j^T A\mathbf{v}_j, \quad j = 1, 2, \dots, n. \quad (2.3)$$

A partir de las expresiones (2.2) y (2.3) se puede construir el algoritmo de Lanczos.

Algorithm 1. Algoritmo de Lanczos.

Entrada: Matriz A simétrica y \mathbf{v}_1 unitario.

Salida: T tridiagonal y V ortogonal.

Step 1 Inicializar \mathbf{v}_1 y $\mathbf{v}_0 \leftarrow 0$, $\beta_0 \leftarrow 0$.

Step 2 Se realiza el bucle:

```
for  $i = 1, 2, \dots, n$ 
   $\mathbf{w}_j \leftarrow A\mathbf{v}_j$ 
   $\alpha_j \leftarrow \mathbf{w}_j^T \mathbf{v}_j$ 
   $\mathbf{w}_j \leftarrow \mathbf{w}_j - \alpha_j \mathbf{v}_j - \beta_j \mathbf{v}_{j-1}$ 
   $\beta_j \leftarrow \|\mathbf{w}_j\|$ 
   $\mathbf{v}_{j+1} \leftarrow \mathbf{w}_j / \beta_j$ 
endfor
```

Step 3 Se hace $\mathbf{w}_n \leftarrow A\mathbf{v}_n$.

Step 4 Se hace $\alpha_n \leftarrow \mathbf{w}_n^T \mathbf{v}_n$.

Un ejemplo es:

Ejemplo 1 (Ejemplo del algoritmo de Lanczos).

Se considera

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}.$$

Aplicando el algoritmo de Lanczos para determinar los valores y vectores propios de esta matriz.

Se inicializa $\mathbf{v}_1 = (1, 0, 0)^T$ y $\mathbf{v}_0 \leftarrow 0$, $\beta_0 \leftarrow 0$.

$j = 1$ Se ejecutan los pasos del algoritmo para $j = 1$.

$$\mathbf{w}_1 = A\mathbf{v}_1 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

$$\alpha_1 = \mathbf{w}_1^T \mathbf{v}_1 = (1, 2, 3) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 1.$$

$$\mathbf{w}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} - 1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - 1\Delta 0 = \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix}.$$

$$\beta_1 = \|\mathbf{w}_1\| = \sqrt{13} = 3.6056.$$

$$\mathbf{v}_2 = \mathbf{w}_1 / \beta_1 = \begin{bmatrix} 0 \\ 0.5547 \\ 0.8321 \end{bmatrix}.$$

$j = 2$ Se ejecutan los pasos del algoritmo para $j = 2$.

$$\mathbf{w}_2 = A\mathbf{v}_2 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5547 \\ 0.8321 \end{bmatrix} = \begin{bmatrix} 3.6056 \\ 4.9923 \\ 6.3791 \end{bmatrix}.$$

$$\alpha_2 = \mathbf{w}_2^T \mathbf{v}_2 = 8.0769.$$

$$\mathbf{w}_2 = \mathbf{w}_2 - \alpha_2 \mathbf{v}_2 - \beta_2 \mathbf{v}_1 = \begin{bmatrix} 0 \\ 0.5120 \\ -0.3414 \end{bmatrix}.$$

$$\beta_2 = \|\mathbf{w}_2\| = 0.6154.$$

$$\mathbf{v}_3 = \mathbf{w}_2 / \beta_2 = \begin{bmatrix} 0 \\ 0.8321 \\ -0.5547 \end{bmatrix}.$$

Se calcula finalmente \mathbf{w}_3 y α_3 .

$$\mathbf{w}_3 A \Delta \mathbf{v}_3 = \begin{bmatrix} 0 \\ 0.2774 \\ 0.5547 \end{bmatrix} \longrightarrow \alpha_3 = \mathbf{w}_3^T \mathbf{v}_3 = -0.0769.$$

$$T_3 = \begin{bmatrix} \alpha_1 & \beta_1 & 0 \\ \beta_1 & \alpha_2 & \beta_2 \\ 0 & \beta_2 & \alpha_3 \end{bmatrix} = \begin{bmatrix} 1 & 3.6056 & 0 \\ 3.6056 & 8.0769 & 0.6134 \\ 0 & 0.6154 & -0.0769 \end{bmatrix}$$

$$V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5547 & 0.8321 \\ 0 & 0.8321 & -0.5547 \end{bmatrix}.$$

Como se puede comprobar en el ejemplo estudiado, se cumple que T es una

matriz semejante a A , por lo que se verifica que

$$AV = VT.$$

Lo más importante de este algoritmo es que se obtiene una matriz tridiagonal T cuyos vectores propios constituyen una buena aproximación de los vectores propios de la matriz A .

El problema ahora es obtener los valores y vectores propios de una matriz tridiagonal T de la forma

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \\ & & & \ddots & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

Pero este problema queda resuelto aplicando el algoritmo QR basado en la factorización QR estudiada en el apartado anterior.

2.9.1. La función de Lanczos en R

R dispone de una función ya implementada que calcula los valores y vectores propios de una matriz simétrica mediante métodos iterativos, utilizando el algoritmo de Lanczos. La función requiere la instalación previa de un paquete denominado **mgcv**. Así, se debe introducir las siguientes instrucciones antes de ejecutar la función:

R 33 (Instalación del paquete *mgcv*).

```
> install.package("mgcv")
> library(mgcv)
```

Esta función es

`slanczos(A, k, k1, tol, nt)`

cuyos argumentos principales son los siguientes:

A Matriz simétrica.

k, k1 El parámetro k debe ser 0 o no negativo. En cuanto al parámetro $k1$, si es negativo, muestra los k valores propios más grandes en valor absoluto, así como los vectores propios. Si, por el contrario, $k1$ es positivo muestra los k valores y vectores propios más grandes (en valor absoluto), así como los $k1$ valores y vectores propios más pequeños.

tol Tolerancia para el cálculo de los valores propios. El error cometido será menor que la magnitud del mayor valor propio multiplicada por *tol* (o la precisión de la máquina).

nt Número de threads a utilizar para el producto de A por el vector.

A continuación se verá algún ejemplo de utilización de esta función.

R 34 (Ejemplo slanczos).

```
slanczos(A, 10, -1, tol=.Machine$double.eps^0.5, nt=1)
```

Esta instrucción produce los 10 valores y vectores propios más grandes de la matriz A .

R 35 (Ejemplo slanczos).

```
slanczos(A, 5, 5)
```

Esta instrucción produce los 5 valores y vectores propios más grandes y más pequeños de la matriz A .

Ejercicio 4.

Dada la matriz

$$A = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix}$$

calcula el valor propio más grande y el más cercano a cero.

2.10. Cadenas de Markov

2.10.1. Introducción

Las **cadenas de Markov** representan una herramienta para analizar el comportamiento y el gobierno de determinados tipos de procedimientos estocásticos, esto es, procesos que evolucionan de forma no determinista a lo largo del tiempo a partir de un conjunto de estados.

Una cadena de Markov representa, por tanto, un sistema que varía su estado a través del tiempo, representando cada cambio una transición del sistema. Dichos cambios no están predeterminados, aunque sí lo está la probabilidad del próximo estado en función de los estados anteriores, probabilidad que es constante a lo largo del tiempo. Eventualmente, en una transición, el nuevo estado puede ser el mismo que el anterior, y es posible que exista la posibilidad de influir en las probabilidades de transición actuando adecuadamente sobre el sistema.

Vamos a centra la atención en las llamadas cadenas de Markov finitas, en las que el número de estados de transición es finito.

Definición 3.

Formalmente, se puede definir una cadena de Markov a partir de los siguientes elementos:

- Un conjunto de estados del sistema.
- La definición de transición.
- La ley de probabilidad condicional, que defina la probabilidad del nuevo estado en función de los estados anteriores.

Los estados son una caracterización de la situación en que se halla el sistema en un instante dado, pudiendo ser dicha caracterización tanto cuantitativa como cualitativa. Desde un punto de vista eminentemente práctico, la pregunta que se puede formular para entender lo que significa un estado concreto en un sistema es ¿cómo están las cosas?

Más formalmente, el estado de un sistema en un instante t es una variable cuyos valores sólo pueden pertenecer al conjunto de estados del sistema. El sistema modelizado por la cadena, por lo tanto, es una variable que cambia de valor con el tiempo, cambio al que se denomina **transición**.

Dicho de otro modo, se trata de un conjunto de variables indexadas en el tiempo E_t donde t denota intervalos significativos de tiempo para el fenómeno estudiado. Los valores de E_t se toman de un conjunto de categorías mutuamente excluyentes denominados estados del sistema. Por tratarse de un proceso estocástico, se conocerá únicamente la probabilidad asociada a cada uno de los estados.

Se puede considerar un ejemplo de cadena de Markov con dos estados A y B ; en general, se tendrá un número finito n de estados. En cada etapa el proceso puede estar en cualquiera de los dos estados; en la etapa siguiente, el proceso puede permanecer en su estado actual o cambiar al otro estado. El estado al que el proceso se mueve en la etapa siguiente y la probabilidad de hacerlo así, dependen solamente del estado actual y no de los estados anteriores del proceso. Estas probabilidades decimos que son probabilidades de transición y suponemos que son constantes (es decir, la probabilidad de moverse del estado i al estado j es siempre la misma). Véase, a continuación, ejemplo en la sección siguiente.

2.10.2. Ejemplo inicial

Supongase que en una zona geográfica, que comprende varias ciudades, se venden dos marcas diferentes de leche, que identificamos como A y B . Se sabe que debido al precio, a la publicidad, etc., el 80 % de las personas que durante un mes compraron la marca A , continuaron comprándola al mes siguiente, mientras que el 20 % cambió a la marca B ; análogamente, el 70 % de las personas que durante un mes compraron la marca B , continuaron comprándola al mes siguiente, mientras que el 30 % cambió a la marca A .

Se elige una muestra de 1000 personas de las que 640 compran la marca A mientras que las 360 restantes compran la marca B . Al cabo de un mes, el

número de personas que compran la marca A será

$$0.80 \cdot 640 + 0.30 \cdot 360 = 620,$$

es decir, el 80 % de las que inicialmente compraban la marca A más el 30 % de las que compraban la marca B que cambiaron a la marca A . Análogamente, al cabo de un mes, el número de personas que compran la marca B será

$$0.20 \cdot 650 + 0.70 \cdot 350 = 380,$$

es decir, el 20 % de las que compraban la marca A que cambiaron a la marca B más el 70 % de las que inicialmente compraban la marca B . Se pueden escribir las expresiones anteriores en forma matricial como

$$\begin{bmatrix} 0.80 & 0.30 \\ 0.20 & 0.70 \end{bmatrix} \begin{bmatrix} 640 \\ 360 \end{bmatrix} = \begin{bmatrix} 620 \\ 380 \end{bmatrix}$$

o, de forma análoga,

$$P\mathbf{x}_0 = \mathbf{x}_1,$$

siendo

$$\mathbf{x}_0 = \begin{bmatrix} 640 \\ 360 \end{bmatrix}, \quad \mathbf{x}_1 = \begin{bmatrix} 620 \\ 380 \end{bmatrix},$$

los vectores correspondientes al número de personas que inicialmente y al cabo de un mes, respectivamente, compraban las marcas A y B (en dicho orden).

La matriz

$$P = \begin{bmatrix} 0.80 & 0.30 \\ 0.20 & 0.70 \end{bmatrix}$$

se denomina **matriz de transición** de la cadena de Markov. Se puede pensar en las columnas de P como los estados actuales y en las filas de P como los estados siguientes.

Para determinar el número de personas que compran cada marca al cabo de dos meses, simplemente se aplica el mismo razonamiento, comenzando con \mathbf{x}_1 en lugar de \mathbf{x}_0 . Así,

$$\mathbf{x}_2 = P \cdot \mathbf{x}_1 = \begin{bmatrix} 0.80 & 0.30 \\ 0.20 & 0.70 \end{bmatrix} \begin{bmatrix} 620 \\ 380 \end{bmatrix} = \begin{bmatrix} 610 \\ 390 \end{bmatrix}.$$

Nótese que P es una **matriz estocástica** (la suma de sus columnas es igual a 1).

En general, si se tiene una cadena de Markov con n estados, entonces la matriz de transición P es una matriz estocástica de tamaño $n \times n$ y los vectores \mathbf{x}_k se dice que son los vectores de estados. Así, para una cadena de Markov se tiene la relación

$$\mathbf{x}_{k+1} = P \cdot \mathbf{x}_k, \quad k = 0, 1, 2, \dots$$

que permite calcular un vector de estados arbitrario de forma iterativa una vez que se conoce \mathbf{x}_0 y P . Es decir, una cadena de Markov está completamente determinada por su matriz de transición y su estado inicial. Supóngase que en el ejemplo no se quiere conocer el número exacto de personas que compra cada una de las marcas de leche sino que, en su lugar, interesan las proporciones de personas que compran cada marca.

Se puede convertir los datos en proporciones dividiendo entre 1000. De este modo, se comenzaría con el vector

$$\begin{bmatrix} \frac{620}{1000} \\ \frac{380}{1000} \end{bmatrix} = \begin{bmatrix} 0.62 \\ 0.38 \end{bmatrix}$$

que proporciona la información de que el 62% de la población compra inicialmente la marca A mientras que el 38% se decanta por la marca B . Así,

$$\mathbf{x}_1 = P \cdot \mathbf{x}_0 = \begin{bmatrix} 0.62 \\ 0.38 \end{bmatrix}.$$

Los vectores de este tipo, cuyas componentes no son negativas y que suman 1 se denominan **vectores de probabilidad**. Por otro lado se sabe que

$$\mathbf{x}_k = P^k \cdot \mathbf{x}_0, \quad k = 1, 2, \dots$$

lo cual sugiere examinar las potencias de P .

Examinando P^2 ,

$$P^2 = \begin{bmatrix} 0.80 & 0.30 \\ 0.20 & 0.70 \end{bmatrix} \begin{bmatrix} 0.80 & 0.30 \\ 0.20 & 0.70 \end{bmatrix} = \begin{bmatrix} 0.70 & 0.45 \\ 0.30 & 0.55 \end{bmatrix}$$

se ve que vuelve a ser una matriz estocástica.

Considérese, por ejemplo, el elemento $p_{12}^2 = 0.45$ del ejemplo anterior. Se puede construir un diagrama de árbol que describa el origen de este elemento. Si inicialmente el proceso está en el estado B , hay cuatro cambios de estado posibles que pueden ocurrir a lo largo de dos meses, los cuales corresponden a las cuatro trayectorias de longitud 2 del diagrama de árbol. Una persona que inicialmente compraba la marca B puede terminar, dos meses después, comprando la marca A , de dos formas diferentes:

- puede continuar comprando B después de un mes y posteriormente comprar A , con probabilidad $0.30 \cdot 0.80 = 0.24$.
- puede cambiar a A después de un mes y entonces quedarse con A , con probabilidad $0.70 \cdot 0.30 = 0.21$.

La suma de ambas probabilidades da una probabilidad total de 0.45. Ahora bien, estos cálculos son precisamente los que hacemos para calcular p_{12}^2 . Así pues, $p_{12}^2 = 0.45$ es la probabilidad de moverse del estado 2 (comprar la marca B) al estado 1 (comprar la marca A) en dos transiciones.

Finalmente, el orden de los subíndices es el contrario del que se podía esperar. Se puede generalizar el argumento del ejemplo anterior para obtener el siguiente resultado

Teorema 1.

Si P es la matriz de transición de una cadena de Markov y P^k representa el producto de la matriz $P = [p_{ij}]$ por ella misma k veces, entonces el elemento p_{ij}^k representa la probabilidad de moverse del estado j al estado i en k transiciones.

Volviendo al ejemplo anterior,

$$\begin{aligned} \mathbf{x}_0 &= \begin{bmatrix} 0.64 \\ 0.36 \end{bmatrix} & \mathbf{x}_1 &= \begin{bmatrix} 0.62 \\ 0.38 \end{bmatrix} & \mathbf{x}_2 &= \begin{bmatrix} 0.61 \\ 0.39 \end{bmatrix} \\ \mathbf{x}_3 &= \begin{bmatrix} 0.605 \\ 0.395 \end{bmatrix} & \mathbf{x}_4 &= \begin{bmatrix} 0.6025 \\ 0.3975 \end{bmatrix} & \mathbf{x}_5 &= \begin{bmatrix} 0.6012 \\ 0.3987 \end{bmatrix} \\ \mathbf{x}_6 &= \begin{bmatrix} 0.6006 \\ 0.3994 \end{bmatrix} & \mathbf{x}_7 &= \begin{bmatrix} 0.6003 \\ 0.3997 \end{bmatrix} & \mathbf{x}_8 &= \begin{bmatrix} 0.6002 \\ 0.3998 \end{bmatrix}. \end{aligned}$$

Lo que se puede deducir de esta sucesión es que, en el futuro, el 60 % de las personas comprarán la marca *A* mientras que el 40 % elegirán la marca *B*. Es fácil comprobar que

$$\mathbf{x}_k = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix},$$

para $k \geq 10$.

Se cumple una propiedad destacada: $P \cdot \mathbf{x} = \mathbf{x}$. Un vector verificando esta propiedad se llama **vector de estado estacionario**. En términos matemáticos se dice que 1 es un valor propio de P y que \mathbf{x} es un vector propio asociado al valor propio 1.

2.10.3. Cadenas de Markov con R

Las cadenas de Markov discretas, como las que se han estudiado en los apartados anteriores, permiten modelar la transición de probabilidades entre estados discretos por medio de matrices.

Existen diversos packages para trabajar con cadenas de Markov. Uno de los más conocidos e interesantes para el trabajo con cadenas de Markov discretas es *markovchain* <https://cran.r-project.org/web/packages/markovchain/>.

Se puede definir la matriz de transición en R de la siguiente forma:

R 36.

```
> marcas <- c("A","B")
> matrizTran <- matrix(c(0.80,0.20,0.30,0.70), nrow=2,
dimnames = list(marcas,marcas))
> matrizTran
  A  B
A 0.8 0.3
B 0.2 0.7
>
```

Se sabe, a partir de los datos del ejemplo anterior, que 640 personas toman la marca *A*, mientras que 360 toman la marca *B*, luego se conoce el vector inicial, que es

R 37.

```
> vectorInicial <- matrix(c(0.64,0.36), nrow=2, dimnames =
list(marcas,""))
> vectorInicial
A 0.64
B 0.36
```

Se puede ir calculando las diversas iteraciones

$$(P\mathbf{x}_0, P^2\mathbf{x}_0, P^3\mathbf{x}_0, P^4\mathbf{x}_0, \dots, P^k\mathbf{x}_0, \dots).$$

R 38.

```
> x1 <- matrizTran %*% vectorInicial
> x2 <- matrizTran %*% x1
> x3 <- matrizTran %*% x2
> x4 <- matrizTran %*% x3
> x5 <- matrizTran %*% x4
> x6 <- matrizTran %*% x5
> x7 <- matrizTran %*% x6
> x7
A 0.6003125
B 0.3996875
```

Al aumentar las iteraciones, el vector \mathbf{x}_k tiende a estabilizarse, de manera que el 60 % de las personas comprarán la marca A mientras que el 40 % elegirán la marca B , a largo plazo.

Antes de finalizar con este ejemplo, se debe recalcar la idea que se ha comentado antes, relacionando el estado estacionario que alcanza la matriz P con el concepto de valores y vectores propios de P . Evidentemente, si llegamos a que

$$P\mathbf{x} = \mathbf{x},$$

la teoría espectral de matrices dice que el vector \mathbf{x} es un vector propio de P asociado al valor propio $\lambda = 1$, que es el mayor en valor absoluto. Como consecuencia de estas propiedades, si se calcula el vector propio de la matriz del ejemplo P asociado con el valor propio 1 se comprueba que se obtiene el mismo resultado. Se puede utilizar R para el cálculo rápido de los valores y vectores propios.

R 39.

```
> P
[,1] [,2]
[1,] 0.8 0.3
[2,] 0.2 0.7
> P <- matrix(c(0.8,0.2,0.3,0.7),nrow=2)
> K <- eigen(P)
$values
[1] 1.0 0.5

$vectors
[,1] [,2]
[1,] 0.8320503 -0.7071068
[2,] 0.5547002 0.7071068

> x1 <- K$vectors[,1]
> k1 <- norm(as.matrix(x1))
> x1/k1
[1] 0.6 0.4
>
```

Varios detalles importante en el cálculo realizado con R son:

- Cuando se calculan los valores y vectores propios mediante la instrucción *eigen*, si se observa el vector propio asociado al valor propio dominante 1, no se obtiene el mismo resultado que cuando se calculan las potencias de P . Esto es debido a que falta normalizar el resultado para estudiar los porcentajes correspondientes.
- Para trabajar con el vector propio asociado a $\lambda = 1.0$, conviene extraerlo de la matriz que tiene todos los vectores propios. Basta extraer la primera columna, puesto que están ya ordenados. Le podemos llamar a ese vector x_1 .
- se normaliza el vector siguiendo el procedimiento habitual, es decir, dividiendo por la norma del vector. Una vez finalizados los cálculos, se comprueba el resultado obtenido, 0.6, 0.4, que se corresponde exactamente con los resultados obtenidos mediante el proceso iterativo.

Ejercicio 5 (Matrices de Markov).

Se realiza un estudio de los movimientos de población entre una ciudad y los suburbios. La migración poblacional entre estas dos partes de la región metropolitana es tal que cada año el 5 % de la población de la ciudad se mueve a los suburbios, mientras que el 3 % de la población de los suburbios se mueve a la ciudad. Supongamos que en 2014 la población de la región era de 600 mil personas en la ciudad y 400 mil personas en los suburbios.

1. ¿Cómo será la distribución de población en 2015?
2. ¿Y en 2018? ¿Y en 2030?
3. ¿Puedes extraer alguna conclusión?

Ejercicio 6 (Matrices de Markov).

Un animal en un laboratorio debe comer una entre tres comidas cada día. El laboratorio recoge datos y estima que si el animal elige una comida un día en una elección, entonces elegirá la misma comida al día siguiente con una probabilidad de un 60 %, mientras que elegirá las otras dos comidas con la misma probabilidad.

1. ¿Cuál es la matriz estocástica o de transición de esta situación?
2. ¿Si el animal elige la comida C1 en su elección inicial, cuál es la probabilidad de que elija la comida C2 en la elección siguiente?
3. ¿Existe una situación estable a largo plazo?

2.10.4. La formación de huracanes como cadenas de Markov.

Los huracanes son un tipo de ciclones tropicales caracterizados por vientos intensos y lluvias abundantes, que se originan sobre los océanos tropicales en ambos lados del ecuador.

Los huracanes pueden llegar a tener un radio de entre 300 y 500 km., mientras que el centro del mismo, llamado ojo del huracán, tiene un diámetro del orden de 20 – 30 km, de forma circular o elíptica.

Diversos factores son imprescindibles para la formación de un huracán, como son:

- La principal fuente de energía para la formación de un huracán es el calor que extrae del océano; el agua del océano deberá estar por encima de los 26.6°C para que se evapore el agua necesaria. Cuando se desplaza por temperaturas menores, la pérdida de energía es grande y se disipa rápidamente.

| Estado térmico | Rango temperaturas | Formación huracán |
|-----------------|--------------------|-------------------|
| Frío (F) | (24.30, 26.02) | NO |
| Moderado 1 (M1) | (26.03, 26.59) | NO |
| Moderado 2 (M2) | (26.60, 27.74) | SI |
| Caliente (C) | (27.75, 29.47) | SI |

Tabla 2.2: Rango de los distintos estados térmicos.

- Los vientos deben encontrarse cerca de la superficie.
- El efecto Coriolis.
- Condiciones inestables en la atmósfera.
- El aire por encima de los 5486 km. necesitará estar húmedo al unirse a la tormenta.

El movimiento de un huracán es como el de un trompo. En sus primeras fases de formación, cuando se le llama tormenta tropical, se desplaza muy lentamente (de 15 a 20 km/h). Conforme el huracán madura aumenta su velocidad (de 30 a 45 km/h). Puede pasar un momento en que se detenga para después acelerarse por vientos del oeste y alcanzar velocidades de 80 a 100 km/h.

A continuación se verá la relación existente entre la formación de huracanes y el álgebra lineal. Se sabe que la variabilidad de la temperatura es un factor bastante importante en la generación de los mismos. La temperatura tanto de la atmósfera como del océano no puede ser determinada con certeza; sin embargo, la probabilidad de que se encuentre a una temperatura en algún momento y lugar determinado puede ser especificada si se han realizado observaciones previas. Este proceso constituye un ejemplo de cadena de Markov.

En muestreos oceánicos por imágenes vía satélite, se podrían registrar en tiempos diferentes (cada día, mes, etc.) tres estados térmicos en el agua de la superficie del océano:

- CÁLIDO (C).
- MODERADO (M).
- FRÍO (F).

Si a cada estado térmico se le asigna una probabilidad asociada a cambiar de un estado a otro o permanecer en el mismo, entonces podremos generar una matriz de transición, P . Con estos datos se puede conocer la probabilidad de que se tengan temperaturas ideales para la formación de huracanes.

Ejemplo concreto.

Con los datos existentes de la temperatura en una determinada región (10° N, $98, 5^\circ$ W), desde el año 1968 se procede a generar una matriz de transición.

Se necesita inicialmente establecer los rangos de los distintos estados térmicos, que pueden ser los que aparecen en la tabla 2.2.

El estado del tiempo moderado se dividió en dos intervalos para conocer las temperaturas de ese rango que ayudan a la formación de huracanes y las que no.

La matriz de transición generada a partir de los datos recogidos es la siguiente

| | F | M1 | M2 | C |
|----|------|------|------|------|
| F | 0.77 | 0.60 | 0.18 | 0.00 |
| M1 | 0.15 | 0.20 | 0.27 | 0.00 |
| M2 | 0.08 | 0.20 | 0.55 | 0.25 |
| C | 0.00 | 0.00 | 0.00 | 0.75 |

Las columnas representan el estado actual, mientras que las filas representan el estado siguiente. Así, por ejemplo, la probabilidad que actualmente la temperatura sea moderada 2 y pase a fría en el instante siguiente es de 0.18.

Si se multiplica esta matriz de transición por el vector de probabilidades para el tiempo actual se obtiene la probabilidad de que la temperatura superficial del océano para el siguiente mes sea de

$$\begin{array}{c|cccc} & F & M1 & M2 & C \\ \hline F & 0.77 & 0.60 & 0.18 & 0.00 \\ M1 & 0.15 & 0.20 & 0.27 & 0.00 \\ M2 & 0.08 & 0.20 & 0.55 & 0.25 \\ C & 0.00 & 0.00 & 0.00 & 0.75 \end{array} \begin{bmatrix} 0.18 \\ 0.32 \\ 0.35 \\ 0.15 \end{bmatrix} = \begin{bmatrix} 0.3936 \\ 0.1855 \\ 0.3084 \\ 0.1125 \end{bmatrix}.$$

Interpretando estos datos se llega a la conclusión de que existe una mayor probabilidad, 0.3936, de que la temperatura superficial caiga dentro del rango de temperatura fría y, por consiguiente, no se formen huracanes. Se advierte que existe una probabilidad de 0.3084 de que la temperatura caiga en el rango de moderado 1. Los otros dos estados son mucho menos probables (del 18 % y del 11 % aproximadamente).

Estos datos se realizan teniendo en cuenta una simplificación del problema, ya que solamente se cuenta como factor para la formación de huracanes la temperatura superficial del océano. Además, sólo se está midiendo los datos en un punto geográfico determinado.

Utiliza R para contestar las siguientes preguntas.

Ejercicio 7.

1. Intenta predecir para los siguientes meses las probabilidades de cada uno de los estados térmicos.
2. ¿Qué probabilidades podemos calcular de aquí a tres años? ¿Y a cinco?
3. Haz un estudio similar variando ligeramente los valores del vector de probabilidad actual. ¿Qué conclusiones obtienes?

2.10.5. Problema de la Ruina del Jugador como cadena de Markov.

Se consideran los siguientes problemas:

1. Una persona tiene 20 euros, necesita 50 y la única posibilidad de obtenerlos es jugar en un casino. Solo se se puede apostar 10 euros y el juego es el siguiente: se tira una moneda y si sale cara ganas 10 euros (tus 10 euros originales más otros 10) y si sale cruz pierdes los 10 euros. Por supuesto la moneda no está trucada y tiene exactamente las mismas posibilidades de salir cara que cruz (50%). La pregunta es ¿Cuál es la probabilidad de obtener los 50 euros que se necesita? ¿Y si se hubiera empezado con 10 euros? ¿Y si se hubiera empezado con 30 euros?
2. El mismo problema anterior, excepto que esta vez se parte de 40 euros y necesita 100. En esta ocasión se puede apostar 10 o 20 euros en cada tirada. ¿Cuál es la mejor estrategia de apuesta?
3. En este caso, el problema es parecido al anterior pero que se apuesta 10 euros al rojo o negro en una ruleta. Esta ruleta tiene 18 números rojos, 18 negros y 2 verdes. Ganar rojo o negro dobla la apuesta. Rojo o negro tienen una probabilidad de salir de $18/38 = 9/19$. Responder a todas las preguntas de los problemas anteriores si tus apuestas se realizan en la ruleta.

Todos estos tipos de problemas son conocidos como *Problemas de la ruina del jugador*. Un jugador sigue apostando hasta que va a la quiebra, o hasta que alcanza un determinado objetivo, y no existen otras posibilidades. El problema tiene muchas soluciones elegantes que se pasarán por alto. Se tiene un ordenador disponible para simular el problema.

En cualquier etapa en el proceso, hay seis estados posibles, dependiendo de la fortuna: 0 euros (se ha perdido) o 10 o 20 o 30 o 40 (todavía se está jugando), o 50 euros (se ha ganado). Cuando se empieza a jugar se está en un cierto estado (con 20 euros en el caso del primer problema). Después de haber jugado un rato, un montón de cosas diferentes pueden haber sucedido, por lo que dependiendo de cuánto tiempo ha pasando, se tiene varias probabilidades de estar en los diversos estados. Llamando a los estados 0 para el Estado 0, 1 para Estado 1 y así hasta 5 para Estado 5 (que representa estar con 50 euros).

En un momento determinado, sea p_0 la probabilidad de tener 0 euros, p_1 la probabilidad de tener 10 euros, ..., y p_5 la probabilidad de ganar (tener 50 euros).

Podemos dar una descripción probabilística de su estado como un vector columna de la siguiente forma:

$$\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix}.$$

Si se está en el Estado 0, o en el Estado 5, se permanecerá allí seguro ya que se habrá llegado al final. Si se está en cualquier otro estado, hay una probabilidad del 50% de ascender en un estado y un 50% de posibilidades de disminuir un estado. Si se expresa en forma de producto de matrices:

$$\begin{pmatrix} 1 & .5 & 0 & 0 & 0 & 0 \\ 0 & 0 & .5 & 0 & 0 & 0 \\ 0 & .5 & 0 & .5 & 0 & 0 \\ 0 & 0 & .5 & 0 & .5 & 0 \\ 0 & 0 & 0 & .5 & 0 & 0 \\ 0 & 0 & 0 & 0 & .5 & 1 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} p_0 + .5p_1 \\ .5p_2 \\ .5p_1 + .5p_3 \\ .5p_2 + .5p_4 \\ .5p_3 \\ .5p_4 + p_5 \end{pmatrix}. \quad (2.4)$$

Claramente, el producto anterior representa el cambio en las probabilidades de estar en los diversos estados, dada una distribución probabilística inicial. La posibilidad de estar en el estado 0 después de un lanzamiento de moneda es la probabilidad de estar en ese estado más la mitad de la posibilidad de que estuviera en el estado 1.

Si a la matriz de la izquierda de la ecuación (2.4) le llamamos P , cada vez que multiplicamos el vector correspondiente a la situación inicial por P , se encontrarán las probabilidades de estar en los distintos estados. Así, por ejemplo después de tirar la moneda 1000 veces, el estado se representará por $P^{1000}v$, donde v es el vector de probabilidades inicial ($p_0 = p_1 = p_3 = p_4 = p_5 = 0$ y $p_2 = 1$).

Pero hacer P^{1000} es un poco pesado, incluso con un ordenador, se puede realizar P^2 , y después $(P^2)^2 = P^4$, $(P^4)^2 = P^8$, y así sucesivamente. Con solo 10 multiplicaciones se puede obtener P^{1024} .

En el caso estudiado, y utilizando 10 dígitos de precisión se calcula P^{1024} :

$$\begin{pmatrix} 1 & .8 & .6 & .4 & .2 & 0 \\ 0 & 1.549 \times 10^{-95} & 0 & 2.506 \times 10^{-95} & 0 & 0 \\ 0 & 0 & 4.05 \times 10^{-95} & 0 & 2.506 \times 10^{-95} & 0 \\ 0 & 2.506 \times 10^{-95} & 0 & 4.05 \times 10^{-95} & 0 & 0 \\ 0 & 0 & 2.506 \times 10^{-95} & 0 & 1.549 \times 10^{-95} & 0 \\ 0 & .2 & .4 & .6 & .8 & 1 \end{pmatrix}.$$

Para propósitos prácticos, se puede aproximar:

$$P^{1024} = \begin{pmatrix} 1 & .8 & .6 & .4 & .2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .2 & .4 & .6 & .8 & 1 \end{pmatrix}.$$

Básicamente, si se comienza con 0, 10, ..., 50 euros, se tienen probabilidades 0, 0.2, 0.4, 0.6, 0.8 y 1.0 de ganar (alcanzar la meta de 50 euros), y las probabilidades complementarias de perder.

Para resolver el segundo problema, se tiene que hacer el mismo cálculo, ya sea con 6 estados (una matriz de 6×6) y encontrar una potencia grande de la matriz.

Para el problema final se puede usar exactamente la misma técnica, pero en este caso la matriz original de probabilidades P tendrá entradas de $9/19$ y $10/19$. En general, si se tiene una probabilidad p de ganar una apuesta y una

probabilidad $q = 1 - p$ de perder, la matriz de transición será:

$$\begin{pmatrix} 1 & q & 0 & 0 & 0 & 0 \\ 0 & 0 & q & 0 & 0 & 0 \\ 0 & p & 0 & q & 0 & 0 \\ 0 & 0 & p & 0 & q & 0 \\ 0 & 0 & 0 & p & 0 & 0 \\ 0 & 0 & 0 & 0 & p & 1 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} p_0 + qp_1 \\ qp_2 \\ pp_1 + qp_3 \\ pp_2 + qp_4 \\ pp_3 \\ pp_4 + p_5 \end{pmatrix}.$$

2.10.6. Diseño de un juego de mesa con cadenas de Markov.

Imagínese que se está diseñando un juego de mesa para niños pequeños y se desea asegurar que el juego no se hace demasiado. ¿Cuántos movimientos se tarda, de promedio, para completar los siguientes juegos?:

1. El juego cuenta consta de 100 casillas, y se inicia en la casilla 1. Se lanza un solo dado de 6 caras (cada uno con una probabilidad $1/6$). Se avanza el marcador tantas casillas como indique el dado. Al llegar a la casilla 100, el juego se termina. No hay que llegar exactamente a 100, por ejemplo, si se está en la casilla 98 y se obtiene un 3, el juego se acaba.
2. Igual que el anterior, pero ahora se tiene que alcanzar exactamente la casilla 100. ¿Hay algún límite para la duración de este juego?

En cierto sentido, es exactamente el mismo problema que el de la Ruina del Jugador, pero no es tan uniforme. Para el primer problema, hay 100 estados, que representan la casilla en la que se encuentra actualmente. Por lo que un vector columna de 100 elementos puede representar la probabilidad de estar en cualquiera de los 100 estados, y una matriz de transición de 100×100 puede representar las probabilidades de transición. Se puede simplificar el juego, sin perder generalidad, con la finalidad de obtener matrices mas pequeñas y manejables, por ejemplo para un juego de 8 casillas en total. Se empieza en la casilla uno y gana si se llega a la casilla ocho. Sea p_i la probabilidad de estar en casilla i , etc. En este caso la matriz de transición sera:

$$\begin{pmatrix} 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 0 \\ 0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 2/6 \\ 0 & 0 & 0 & 0 & 1/6 & 1/6 & 1/6 & 3/6 \\ 0 & 0 & 0 & 0 & 0 & 1/6 & 1/6 & 4/6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/6 & 5/6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{pmatrix}.$$

Si se obliga a terminar exactamente en la casilla 8, la matriz de transición cambia:

$$\begin{pmatrix} 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 0 \\ 0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 2/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 0 & 3/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 0 & 0 & 4/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5/6 & 1/6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{pmatrix}.$$

2.11. Gráficos por Computador

Algunos dibujos pueden ser trazados con una larga serie de líneas mediante instrucciones. Por ejemplo, la letra F puede ser dibujada en su orientación natural en el origen de coordenadas con las siguientes instrucciones.:

1. Dibujar una línea de $(0, 0)$ a $(0, 5)$.
2. Dibujar una línea de $(0, 5)$ a $(3, 5)$.
3. Dibujar una línea de $(0, 3)$ a $(2, 3)$.

Se quiere ahora dibujar formas más complicadas que la letra F con cientos de instrucciones similares a las anteriores teniendo en cuenta que para ello se van a utilizar matrices. Se van a tratar los siguientes temas:

1. ¿Cómo se pueden convertir las coordenadas para que el dibujo sea el doble de grande? ¿Cómo se haría par hacerlo dos veces mayor (y -dirección) y tres veces mayor a lo ancho (x -dirección)?
2. ¿Se podría dibujar la imagen especular a través del eje y ?
3. ¿Cómo se puede mover el dibujo 4 unidades hacia la derecha y 5 unidades hacia abajo?
4. ¿Se puede rotar el dibujo 90° en sentido antihorario alrededor del origen? ¿Se puede rotar un ángulo θ en sentido antihorario alrededor del origen?
5. ¿Se puede rotar el dibujo un angle θ alrededor del punto $(7, -3)$?
6. ¿Qué ocurriría si los gráficos fueran en tres dimensiones? ¿Se podrían resolver los problemas similares a los mencionados anteriormente en coordenadas de 3 dimensiones?

Resulta que las respuestas a todos los problemas anteriores está en hacer multiplicaciones por una matriz, por supuesto una matriz diferente para cada uno de ellos. Aquí están las soluciones:

Solución 1:

Para escalar en la dirección del eje x por un factor de 2, se necesita multiplicar todas las coordenadas x por 2. Igualmente, para escalar en la dirección del eje y por un factor de 2, se necesita multiplicar todas las coordenadas y por 2. En general para escalar cualquier gráfico por un factor s_x en la dirección x y un

factor s_y en la dirección del eje y se tiene que multiplicar todos los vectores de entrada por una matriz de escalado:

$$\begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \end{pmatrix}.$$

Solución 2:

Una imagen especular es escalar por -1 . Así pues, obtener la imagen especular a través del eje y -axis significa que cada coordenada x se sustituirá por su negativa. Es decir, se utiliza una matriz para realizar la multiplicación:

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -x \\ y \end{pmatrix}.$$

Solución 3:

Para trasladar un punto 4 unidades a la derecha y 5 unidades hacia abajo, se necesita añadir 4 a cada x y restar 5 a cada coordenada y . Si se intenta hacer esto con productos de matrices 2×2 se ve que es imposible. Esto viene del hecho de que la traslación no es una aplicación lineal. Para solucionar este problema, se añade a cada vector de dos componentes, una tercera igual a 1. Así, el punto $(3, 6)$ se convertirá en $(3, 6, 1)$, el origen de coordenadas será $(0, 0, 1)$, etc. Los vectores columna ahora tendrán tres filas por lo que la matriz de transformación será 3×3 . Para trasladar t_x en el eje x y t_y en la dirección del eje y , se multiplica el vector por una matriz:

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}.$$

El vector resultante es justo lo que se quiere y también tiene el 1 añadido como tercera componente. Para obtener una solución particular al problema propuesto al principio, simplemente se hace $t_x = 4$ y $t_y = -5$.

La pregunta que se puede hacer es: ¿qué ocurre si tenemos que mezclar traslaciones con escalados? ¿Habría que utilizar las nuevas coordenadas para hacer la traslación, realizar el producto matricial, volver a dos coordenadas y realizar el otro producto matricial? Como resulta un poco engorroso lo que se hace es usar "siempre" las coordenadas extendidas, tanto para traslaciones como escalados o rotaciones. Solo hay que cambiar la matriz de escalado por:

$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ 1 \end{pmatrix}.$$

A partir de ahora siempre se añadirá la tercera componente al final.

Solución 4:

Básicamente, rotar un punto en sentido antihorario 90° alrededor del origen de coordenadas es convertir las coordenadas x originales en las coordenadas y y viceversa. Pero, hay que tener en cuenta un detalle importante, cualquier punto que tenga una coordenada positiva y se transformará, después de la rotación, en una coordenada x negativa y viceversa. En definitiva, la nueva coordenada y es la vieja coordenada x , y la nueva x es la negativa de la vieja coordenada y .

Es decir la matriz de rotación de 90° será:

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} -y \\ x \\ 1 \end{pmatrix}.$$

En general para una rotación antihoraria de un ángulo θ se realiza utilizando la siguiente matriz de multiplicación:

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x \cos \theta - y \sin \theta \\ x \sin \theta + y \cos \theta \\ 1 \end{pmatrix}.$$

Solución 5:

En este problema es donde el la utilidad de la matrices empieza a vislumbrarse. Si lo que se quiere es rotar un punto alrededor de otro y no del origen de coordenadas, lo primero que se debe hacer es una traslación del punto $(7, -3)$ al origen, realizar la rotación alrededor del origen, para finalmente volver a trasladar el resultado al punto $(7, -3)$. Cada una de estas tres operaciones se puede realizar mediante productos matriciales. En este caso la solución sería:

$$\begin{pmatrix} 1 & 0 & 7 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -7 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Hay que tener especial cuidado con el orden de las matrices. La matriz mas cercana al vector columna $(x, y, 1)$ es la primera operación que se realiza (en nuestro caso trasladar el punto $(7, -3)$ al origen). La siguiente operación a realizar es la rotación de un ángulo θ , para finalmente, volver a trasladar el resultado al punto $(7, -3)$ (matriz de la izquierda).

Solución 6:

El caso de tres dimensiones es exactamente igual al de dos dimensiones, solo que utilizando matrices 4×4 .

A la izquierda esta la matriz para escalado de s_x , s_y , y s_z en las direcciones x , y y z respectivamente. A la derecha se tiene la matriz de traslación en las tres direcciones para t_x , t_y y t_z .

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Finalmente, para rotar un ángulo θ en sentido antihorario alrededor de los tres ejes se multiplica por la matriz apropiada (izquierda alrededor del eje x , en medio alrededor del eje y y derecha alrededor del eje z):

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Imaginemos que se quiere trasladar y rotar una imagen que representa un cuadrado. Los vértices se trasladan $(1, 2)$ y después se rota el cuadrado un ángulo de 45°

R 40 (Ejemplo de una traslación y rotación).

```
> plot.new()
> plot.window(c(0,3),c(0,6))
> axis(1)
> axis(2)
> x<-c(0,2,2,0,0)
> y<-c(0,0,2,2,0)
> lines(x,y)
```

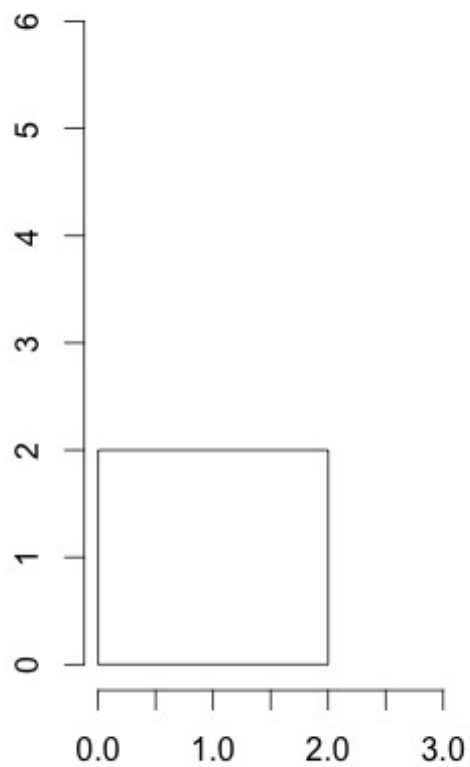


Figura 2.3: Cuadrado original.

El siguiente paso es trasladar el cuadrado, rotarlo y volverlos a trasladar.

R 41 (Ejemplo de una traslación y rotación).

```

> Mtras2D <- matrix(c(1,0,0, 0,1,0, 1,2,1), nrow=3,ncol=3)
> Mtras2D1 <- matrix(c(1,0,0, 0,1,0, -1,-2,1), nrow=3,ncol=3)
> Mrot_45 <- matrix(c(cos(45),sin(45),0, -sin(45), cos(45),0, 0,0,1),nrow=3,ncol=3)
> vert1 <- matrix(c(0, 0, 1), nrow=3,ncol=1)
> vert2 <- matrix(c(0, 2, 1), nrow=3,ncol=1)
> vert3 <- matrix(c(2, 2, 1), nrow=3,ncol=1)
> vert4 <- matrix(c(2, 0, 1), nrow=3,ncol=1)
> newver1 <- Mtras2D %*% Mrot_45 %*% Mtras2D1 %*% vert1
> newver2 <- Mtras2D %*% Mrot_45 %*% Mtras2D1 %*% vert2
> newver3 <- Mtras2D %*% Mrot_45 %*% Mtras2D1 %*% vert3
> newver4 <- Mtras2D %*% Mrot_45 %*% Mtras2D1 %*% vert4
> nuevover1 <- newver1[1:2]
> nuevover2 <- newver2[1:2]
> nuevover3 <- newver3[1:2]
> nuevover4 <- newver4[1:2]
> plot.new()
> plot.window(c(0,3),c(0,6))
> axis(1)
> axis(2)
> x<-c(nuevover1[1],nuevover2[1],nuevover3[1],nuevover4[1],nuevover1[1])
> y<-c(nuevover1[2],nuevover2[2],nuevover3[2],nuevover4[2],nuevover1[2])
> lines(x,y)

```

2.12. Gráfico de Rutas

Imagínese una situación en la que hay 7 lugares posibles: 1, 2, ..., 7 conectados por calles. Estas son de un solo sentido y conectan siempre un par de lugares. Por ejemplo, si se puede ir a partir de la ubicación 3 a la 7, habrá una calle etiquetada (3, 7). La lista de calles es:

(1, 2), (1, 7), (2, 3), (2, 4), (2, 5), (3, 6), (4, 5), (4, 6), (5, 6), (5, 7), (6, 7), (7, 1)

Si se comienza en la posición 1, y se dan 16 pasos, ¿cuántas rutas diferentes hay que poner en cada una de las ubicaciones? (La discusión que sigue será mucho más fácil de seguir si se dibuja una imagen. Hacer un círculo de 7 puntos, numerados del 1 al 7, y para cada uno de los pares anteriores, dibujar una flecha desde el primer punto del par al segundo).

Se puede representar el número de caminos a cada localización mediante un vector columna:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \end{pmatrix},$$

donde p_i es el número de caminos a la localización i . El vector inicial será $p_1 = 1$ y $p_2 = p_3 = p_4 = p_5 = p_6 = p_7 = 0$. En otras palabras, después de 0 etapas hay exactamente un camino a la localización 1 y no hay caminos a las otras localizaciones. Se puede ver que si se multiplica ese vector inicial por una matriz de unos se muestra exactamente un camino a 2 y un camino a 7.

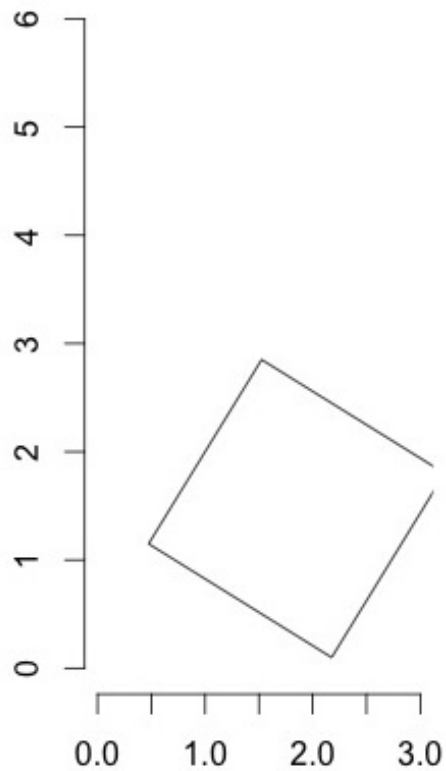


Figura 2.4: Cuadrado trasladado y rotado.

La matriz de transición será

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \end{pmatrix}, \quad (2.5)$$

la cual cuenta el número de caminos en $n + 1$ etapas si la entrada es el número de caminos en la etapa n .

Si la matriz de la izquierda de la ecuación (2.5) se denomina P , entonces

después de 16 etapas, el contador de rutas vendrá dado por:

$$P^{16} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \end{pmatrix}.$$

Con un ordenador se puede obtener P^{16} :

$$\begin{pmatrix} 481 & 440 & 104 & 280 & 292 & 188 & 288 \\ 288 & 293 & 72 & 192 & 176 & 104 & 188 \\ 188 & 184 & 48 & 125 & 120 & 72 & 104 \\ 188 & 184 & 48 & 125 & 120 & 72 & 104 \\ 292 & 300 & 77 & 203 & 197 & 120 & 176 \\ 384 & 404 & 107 & 281 & 280 & 173 & 264 \\ 728 & 676 & 188 & 480 & 476 & 288 & 481 \end{pmatrix}.$$

Y puesto que inicialmente $p_1 = 1$ y $p_2 = p_3 = p_4 = p_5 = p_6 = p_7 = 0$, se tiene:

$$P^{16} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \end{pmatrix} = \begin{pmatrix} 481 & 440 & 104 & 280 & 292 & 188 & 288 \\ 288 & 293 & 72 & 192 & 176 & 104 & 188 \\ 188 & 184 & 48 & 125 & 120 & 72 & 104 \\ 188 & 184 & 48 & 125 & 120 & 72 & 104 \\ 292 & 300 & 77 & 203 & 197 & 120 & 176 \\ 384 & 404 & 107 & 281 & 280 & 173 & 264 \\ 728 & 676 & 188 & 480 & 476 & 288 & 481 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 481 \\ 288 \\ 188 \\ 188 \\ 292 \\ 384 \\ 728 \end{pmatrix},$$

Como se puede ver, hay 481 rutas a la localización 1, 288 rutas a la location 2, etc.

Esto es muy difícil de comprobar para 16 etapas, pero para tres es más sencillo. La ecuación correspondiente para solo 3 etapas es:

$$P^3 \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 3 \\ 2 \end{pmatrix}.$$

Así, se ve que no hay ninguna ruta para llegar a la localización 1 en tres etapas, una ruta para llegar a la localización 2: $(1 \rightarrow 7 \rightarrow 1 \rightarrow 2)$, no hay ninguna ruta para llegar a las localizaciones 3 y 4, una ruta para llegar a la localización 5: $(1 \rightarrow 2 \rightarrow 4 \rightarrow 5)$, tres rutas para llegar a la localización 6: $(1 \rightarrow 2 \rightarrow 3 \rightarrow 6)$, $(1 \rightarrow 2 \rightarrow 4 \rightarrow 6)$, y $(1 \rightarrow 2 \rightarrow 5 \rightarrow 6)$, y finalmente dos rutas para llegar a la localización 7: $(1 \rightarrow 2 \rightarrow 5 \rightarrow 7)$ y $(1 \rightarrow 7 \rightarrow 1 \rightarrow 7)$.

2.13. Ejercicios

Ejercicio 8.

En una heladería, por una copa de la casa, dos horchatas y cuatro batidos te cobran 34€. Unos días después, por cuatro copas de la casa y cuatro horchatas te cobran 44€. Un tercer día, te piden 26€ por una horchata y cuatro batidos. ¿Alguno de los días se han equivocado en la cuenta?

Ejercicio 9.

Un comerciante de café vende tres mezclas de café. Una bolsa de la mezcla de la casa contiene 300 gramos de granos colombianos y 200 gramos de café tostado francés. Una bolsa de la mezcla especial contiene 200 gramos de granos colombianos, 200 gramos de judías de Kenia, y 100 gramos de café tostado francés. Una bolsa de la mezcla gourmet contiene 100 gramos de granos colombianos, 200 gramos de judías de Kenia, y 200 gramos de café tostado francés. El comerciante tiene a su disposición 30 kilogramos de granos colombianos, 15 kilogramos de granos de Kenia, y 25 kilogramos de granos tostados franceses. Si desea utilizar de los granos, ¿cuántas bolsas de cada tipo de mezcla se puede hacer?

Ejercicio 10.

Una empresa de automóviles dispone de dos plantas de fabricación localizadas una en España y otra en Gran Bretaña. En ellas se fabrica dos modelos de coches M_1 y M_2 , en tres colores x, y, z . Su capacidad de producción diaria en cada planta está dada por las siguientes matrices:

$$A = \begin{bmatrix} 300 & 95 \\ 250 & 100 \\ 200 & 100 \end{bmatrix} \quad B = \begin{bmatrix} 190 & 90 \\ 200 & 100 \\ 150 & 80 \end{bmatrix}.$$

1. Determina la representación matricial de la producción total por día.
2. Si se eleva la producción en España un 20 % y se disminuye en Gran Bretaña un 10 % ¿qué matriz representa la nueva producción total?

Ejercicio 11.

Consideremos el alfabeto siguiente:

```

A --> 1  B --> 2  C --> 3  D --> 4      E --> 5      F --> 6      G --> 7
H --> 8  I --> 9      J --> 10     K --> 11     L --> 12     M --> 13
N --> 14
Ñ --> 15  O --> 16      P --> 17     Q --> 18      R --> 19     S --> 20
T --> 21
U --> 22  V --> 23     W --> 24      X --> 25      Y --> 26
Z --> 27
, --> 29      ; --> 30      : --> 31      % --> 32      ! --> 33      ¿ --> 34
? --> 35
@ --> 36      espacio --> 0

```

Supongamos que el mensaje que queremos cifrar, una vez identificado numéricamente mediante el alfabeto anterior, es el siguiente:

```

1  0  18  22  9  5  14  0  3  1
4  19  22  7  1  29  0  4  9  16
20 0  12  5  0  1  26  22  4  1

```

Agrupamos elementos de cinco en cinco de este vector de números, con el fin de aplicar posteriormente sobre estos vectores una matriz de transformación que nos permita la codificación. Agrupamos los números anteriores de la siguiente manera:

```

1  0  18  22  9
5  14  0  3  1
4  19  22  7  1
29 0  4  9  16
20 0  12  5  0
1  26  22  4  1

```

Consideremos la siguiente matriz de cifrado

$$A = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & -1 & 1 & -1 \\ -4 & 2 & -3 & 5 & 1 \\ 1 & 0 & 1 & -1 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Se pide:

1. Cifra el mensaje anterior utilizando la matriz A .
2. Calcula la matriz de descifrado.
3. Supongamos que recibes el mensaje siguiente:
S!W?,E: ,DRG,R XQGWDHMSZBOHCRC
¿Puedes descifrarlo?