

Matrius-guia18-mates1-21: Determinants

f.u., a.g.

1/28/2021

Definició i notacions

Calcular el determinant d'una matriu quadrada en R és ben fàcil, fem servir la funció `det()`.

```
A <- matrix(c(2, 0, 3, 4, 7, 5, 5, 4, 9), ncol = 3)
A
```

```
      [,1] [,2] [,3]
[1,]    2    4    5
[2,]    0    7    4
[3,]    3    5    9
```

```
det(A)
```

```
[1] 29
```

Propietats dels determinants

Fem servir aquesta matriu

```
A <- matrix(sample(-12:12), nrow = 5) # són els nombres des de -12 fins a 12 reordenats aleatòriament
A
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]   -8  -12  -5    0   10
[2,]   -7  -10    6    3   12
[3,]    8    9   -2  -11   -6
[4,]    2   -1   -4    5    1
[5,]   11   -3    4    7   -9
```

```
det(A)
```

```
[1] 108981
```

```
det(t(A)) # Propietat D2: el determinant de la transposada és igual al de la original
```

```
[1] 108981
```

Propietat D3, la comprovem en un cas: la matriu B és com la A però la fila 3 està multiplicada per -2.

```
B <- A
B[3,] <- -2 * B[3,]
B
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]   -8  -12  -5    0   10
[2,]   -7  -10    6    3   12
[3,]  -16  -18    4   22   12
[4,]    2   -1   -4    5    1
```

```
[5,] 11 -3 4 7 -9
```

```
det(B)
```

```
[1] -217962
```

```
# Propietat D4, el determinant de 2A és 2^5 vegades el de A
```

```
det(2*A)
```

```
[1] 3487392
```

```
(2^5 * det(A))
```

```
[1] 3487392
```

Propietat D5:

Per intercanviar dues files demanem les files 2,1,3,4,5 i totes les columnes:

```
A2 <- A[c(2,1,3,4,5),]
```

```
A2
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]   -7  -10    6    3   12
[2,]   -8  -12   -5    0   10
[3,]    8    9   -2  -11   -6
[4,]    2   -1   -4    5    1
[5,]   11   -3    4    7   -9
```

```
det(A2)
```

```
[1] -108981
```

Propietat D8:

Per substituir la columna 3 per la suma d'ella amb -2 vegades la columna 4:

```
A[,3] <- A[,3] - 2 * A[,4]
```

```
A
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]   -8  -12   -5    0   10
[2,]   -7  -10    0    3   12
[3,]    8    9   20  -11   -6
[4,]    2   -1  -14    5    1
[5,]   11   -3  -10    7   -9
```

```
det(A) # el determinant no ha canviat
```

```
[1] 108981
```

Propietat D9: El determinant del producte és el producte de determinants

```
A <- matrix(sample(-12:12), nrow = 5) # són els nombres des de -12 fins a 12 reordenats aleatòriament
```

```
B <- matrix(sample(-12:12), nrow = 5) # són els nombres des de -12 fins a 12 reordenats aleatòriament
```

```
det(A%*%B)
```

```
[1] -24316963476
```

```
det(A)*det(B)
```

```
[1] -24316963476
```

Determinants i dependència lineal

Si tenim dos vectors del pla $\mathcal{R}^2$ o tres vectors de $\mathcal{R}^3$ podem comprobar la seva independència lineal formant una matriu amb els vectors i calculant-ne el determinant. Aquí ho fem a vectors de $\mathcal{R}^4$ i fem servir la funció `cbind()` que serveix per juntar-los en columnes (la funció `rbind()` fa el mateix però en files).

```
v1 <- c(1, 0, -1, 1)
v2 <- c(-1, 1, 0, 1)
v3 <- c(0, -1, -1/2, 1/2)
v4 <- c(1/3, -3, 1/4, 0)
M <- cbind(v1,v2,v3,v4)
M
```

```
      v1 v2  v3      v4
[1,]  1 -1  0.0  0.3333333
[2,]  0  1 -1.0 -3.0000000
[3,] -1  0 -0.5  0.2500000
[4,]  1  1  0.5  0.0000000
```

```
round(M,2) # arrodonim a dos decimals
```

```
      v1 v2  v3      v4
[1,]  1 -1  0.0  0.33
[2,]  0  1 -1.0 -3.00
[3,] -1  0 -0.5  0.25
[4,]  1  1  0.5  0.00
```

```
det(M)
```

```
[1] -2.458333
```

Producte vectorial

Quan parlem de producte de vectors hem d'anar en compte. Fixa't en els diferents productes:

```
v1 <- c(1, 0, -1, 1)
v2 <- c(-1, 1, 1, 1)
v1 * v2 # és el producte terme a terme
```

```
[1] -1  0 -1  1
```

```
v1 %*% v2 # fem servir el producte matricial i obtenim el que en diem producte escalar
```

```
      [,1]
[1,]    -1
```

```
sum(v1*v2) # millor així, que ens ho torna com a un sol nombre
```

```
[1] -1
```

El producte vectorial que usem en el curs no està definit en R, però es pot definir (o utilitzar el package **pracma** que té moltes funcions útils per a l'àlgebra lineal i de matrius)

```
producteVectorial <- function (x, y){
  if (is.numeric(x) && is.numeric(y))
    if (is.vector(x) && is.vector(y)) {
      if (length(x) == length(y) && length(x) == 3) {
        xxy <- c(x[2] * y[3] - x[3] * y[2], x[3] * y[1] -
                  x[1] * y[3], x[1] * y[2] - x[2] * y[1])
        return(xxy)
      }
    }
}
```

```

    }
  }
  stop("x i y han de ser vector numèrics de llargada 3")
}

```

```

u1 <- c(1, 0, -1)
u2 <- c(-1, 1, 1)
producteVectorial(u1,u2)

```

```
[1] 1 0 1
```

I si volem normalitzar el resultat, caldrà dividir-lo per la seva norma:

```

v <- producteVectorial(u1,u2)
v / sqrt(sum(v * v))

```

```
[1] 0.7071068 0.0000000 0.7071068
```

Un altre exemple:

```

a <- c(1, 2, 3)
b <- c(1, -3, 3)
v <- producteVectorial(a,b)
nv <- sqrt(sum(v*v))
v/nv

```

```
[1] 0.9486833 0.0000000 -0.3162278
```

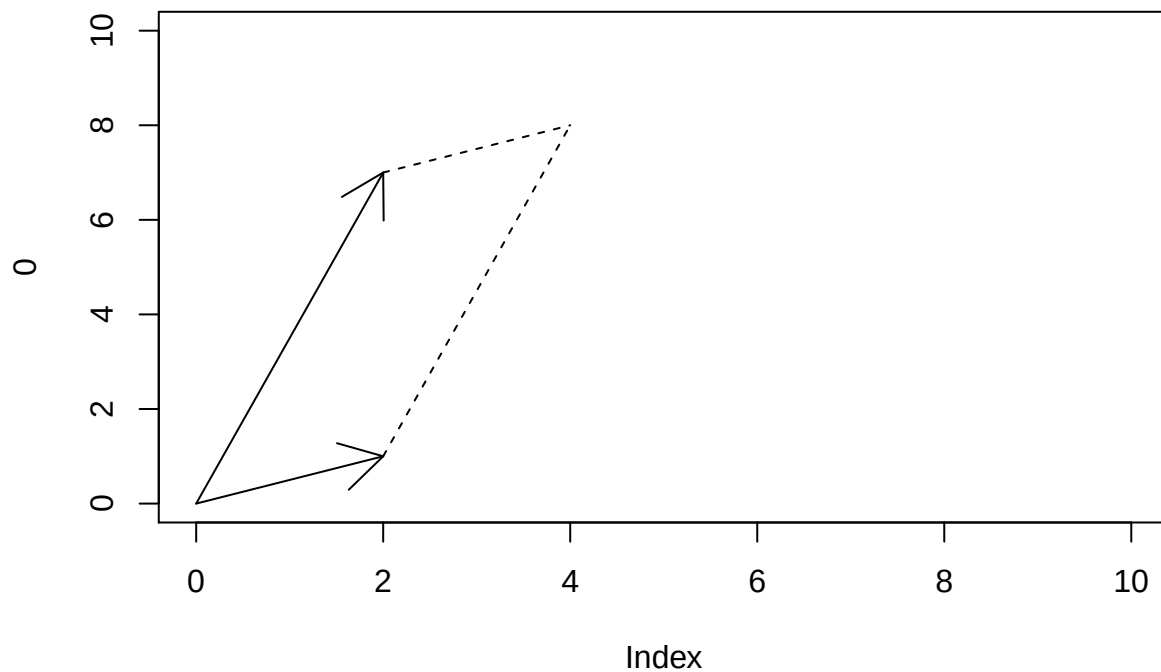
Exercicis

En el pla, l'àrea determinada per dos vectors coincideix amb el determinant.

```

plot(0, type="n", xlim = c(0,10), ylim = c(0,10))
u = c(2,1)
v = c(2,7)
arrows(0,0,u[1],u[2])
arrows(0,0,v[1],v[2])
lines(rbind(u, u+v), lty=2)
lines(rbind(v, u+v), lty=2)

```



```
mat <- rbind(u,v) # juntem per files els dos vectors
mat
```

```
  [,1] [,2]
u     2     1
v     2     7
```

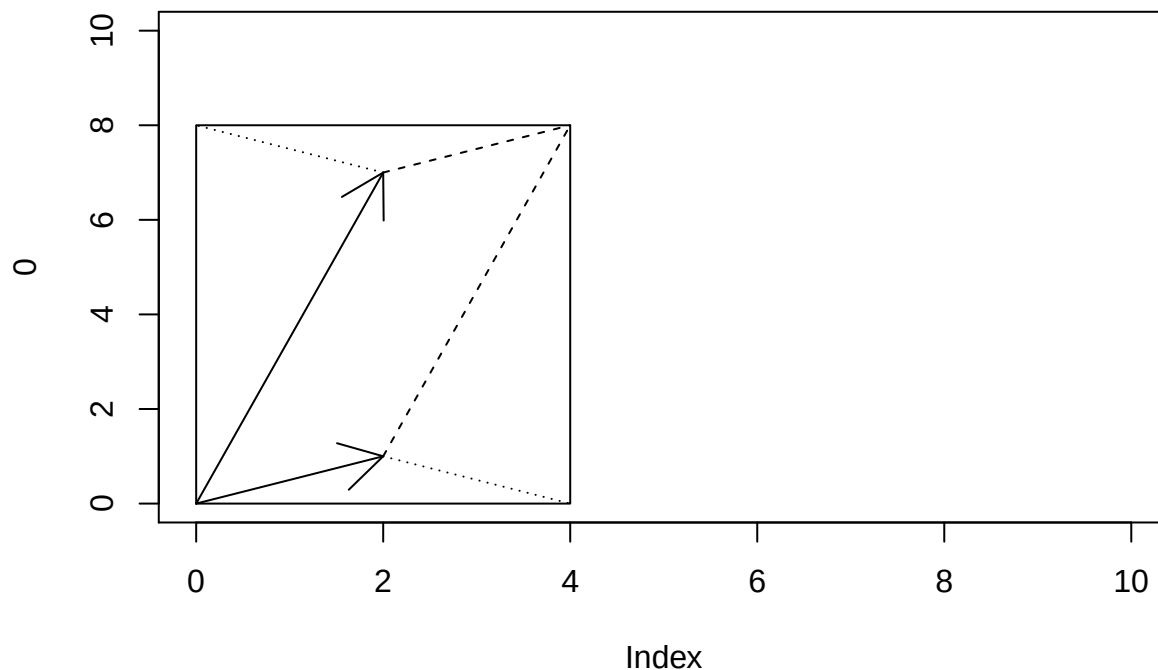
```
det(mat)
```

```
[1] 12
```

Per demostrar-ho, tanquem el paralelogram en un rectangle i restem les àrees dels triangles que sobren.

```
plot(0, type="n", xlim = c(0,10), ylim = c(0,10))
u = c(2,1)
v = c(2,7)
arrows(0,0,u[1],u[2])
arrows(0,0,v[1],v[2])
lines(rbind(u, u+v), lty="dashed")
lines(rbind(v, u+v), lty="dashed")

uv = u+v
polygon(x=c(0,uv[1],uv[1],0), y=c(0,0,uv[2], uv[2]))
lines(rbind(u, c(uv[1],0)), lty="dotted")
lines(rbind(v, c(0,uv[2])), lty="dotted")
```



Qüestions avançades de computació

Els programes de computació numèrica sovint treballen amb els nombres “en coma flotant” (*floating point*), és a dir, en la forma **mantisa** * **10^{exponent}**, on la mantisa és un nombre entre 0 i 1, i l'exponent pot ser molt gran en valor absolut, positiu o negatiu. (De fet això dins l'ordinador es fa en sistema binari, però no entrem per aquí ara).

Així, la representació de $1/3$, per exemple, serà 0.333333×10^0 , i l'import total dels ingressos de l'estat espanyol al 2021, segons els pressupostos generals de l'estat (157.114.530,88 euros) estaria representat per $0.15711453088 \times 10^9$.

Llavors, molts dels algorismes de càlcul, com per exemple el del determinant d'una matriu quadrada tenen versions més eficients que fan servir no els nombres enters sino nombres amb coma flotant, i per tant aproximats i amb un cert marge d'error, molt petit.

Això explica que passin coses com el que reflectim aquí:

A

```

      [,1] [,2] [,3] [,4] [,5]
[1,]   -7    3   -8  -10    7
[2,]   11   -9  -11    5   -5
[3,]   12   10    0   -6   -4
[4,]    4    9    2   -1    1
[5,]    8   -2  -12   -3    6

```

```
# Segons la propietat D4, el determinant de 2A és 2^5 vegades el de A
```

```
x1 <- det(2*A)
```

```
x1
```

```
[1] -3470784
```

```
x2 <- (2^5 * det(A))
```

```
x2
```

```
[1] -3470784
```

Per tant, x1 i x2 són iguals. Són iguals?

```
x1 == x2
```

```
[1] FALSE
```

```
x1 - x2 # quina és la diferència?
```

```
[1] -1.396984e-09
```

Si el càlcul del determinant es fes de manera exacta amb nombres enters, els dos nombres serien exactament iguals. Però R treballa amb coma flotant pràcticament sempre, i quan es tracta de matrius, sempre. Llavors fent els càlculs amb coma flotant apareixen petits errors. En aquest cas hi ha una diferència, molt petita, sí, però no és zero.

Si t'interessa el tema, potser (https://ca.wikipedia.org/wiki/Coma_flotant) et pot ajudar a entendre-ho millor.