

Sistemas compatibles determinados en R

Daniel Eduardo Macias Estrada

27/8/2020

Resolución de un sistema compatible determinado con la función solve

Dado el sistema de ecuaciones lineales

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{cases}$$

Al representarlo en su forma matricial $AX = b$, podemos darle una solución con el uso de la función **solve(A,b)**, la cual funciona si, y solo si, el sistema de ecuaciones es compatible determinado.

Ejemplo 1

Dado el siguiente sistema de ecuaciones lineales, calculemos su solución

$$\begin{cases} x_1 + x_2 + 2x_3 &= 9 \\ 2x_1 + 4x_2 - 3x_3 &= 1 \\ 3x_1 + 6x_2 - 5x_3 &= 0 \end{cases}$$

En este caso, tanto el número de ecuaciones, así como el número de incógnitas es 3

En su forma matricial, se expresaría en R como:

```
A = rbind(c(1,1,2), c(2,4,-3),c(3,6,-5))
b = c(9,1,0)
Ab = cbind(A,b)
```

Observamos en el anterior chunk que la función **cbind()** es usada para unir dos matrices en columnas, solo si la dimensión de ambas lo permite. En el documento de **Matrices con R** se menciona la misma función, pero con un uso distinto.

Mediante el teorema de Rouché-Frobenius, y con ayuda de R, verificamos si el rango de la matriz A es igual al rango de la matriz Ampliada, para saber si el sistema es compatible

```
qr(A)$rank == qr(Ab)$rank
```

```
## [1] TRUE
```

Ahora nos aseguramos si este es igual al número de incógnitas, el cual es 3, para saber si es un sistema compatible determinado.

```
qr(A)$rank == 3
```

```
## [1] TRUE
```

De esta manera, es posible aplicar la función **solve()** a este caso, por lo que procedemos a realizarla

```
solve(A,b)
```

```
## [1] 1 2 3
```

La solución que nos devuelve, se puede expresar como:

$$x_1 = 1, \quad x_2 = 2, \quad x_3 = 3$$

Una forma de comprobar este resultado es sustituyendo días variables en cada una de las ecuaciones del sistema y verificar la igualdad en todas.

Otra forma de comprobar el resultado, es realizando un producto de matrices con la matriz A y el vector de resultados

```
solution = c(1,2,3)
A%%solution
```

```
##      [,1]
## [1,]    9
## [2,]    1
## [3,]    0
```

o de la siguiente forma

```
A%%solution == b
```

```
##      [,1]
## [1,] TRUE
## [2,] TRUE
## [3,] TRUE
```

Así, nos damos cuenta que el vector (1,2,3) es correcto

Resolución de un sistema compatible determinado con la libreria matlab

Otra forma para dar solución a un sistema determinado, se da gracias a la libreria **Matlib**, la cual ofrece variedad de funciones útiles en resolver sistemas de ecuaciones lineales, sobre todos aquellos con 2 o 3 ecuaciones

Ejemplo 2

Dada el siguiente sistema de ecuaciones lineales

$$\begin{cases} 2x_1 + 2x_2 = 1 \\ -x_1 + x_2 = 2 \end{cases}$$

Al expresarlo en su forma matricial, obtenemos

```
A = rbind(c(2,2),c(-1,1))
b = c(1,2)
Ab = cbind(A,b)
```

Una vez tenemos la forma matricial, podemos mostrar el sistema con la función **showEqn()** de la libreria **matlib**

```
library(matlib)
showEqn(A,b)
```

```
## 2*x1 + 2*x2 = 1
## -1*x1 + 1*x2 = 2
```

Esta misma librería nos permite obtener el rango de una matriz con la función **R()**

```
R(A)
```

```
## [1] 2
```

```
R(Ab)
```

```
## [1] 2
```

Gracias a esto podemos determinar el tipo de sistema con el que se trabaja. En este caso, al ser igual los rangos de ambas matrices, del mismo modo que es igual al número de incógnitas, podemos decir que el sistema es compatible determinado.

Por otro lado, con la función **all.equal()** es posible comprobar que un sistema es compatible. Esta función compara los rangos de matrices, y determinar si son iguales

```
all.equal(R(A),R(Ab))
```

```
## [1] TRUE
```

Finalmente, con la función **Solve()** de **matlib** podemos resolver el sistema.

```
Solve(A, b, fractions = TRUE)
```

```
## x1 = -3/4
## x2 = 5/4
```

Cabe resaltar que el parámetro **fractions**, que recibe un valor booleano, permite mostrar las soluciones no enteras en forma de fracción, siempre que exista

Representación de sistemas con R

Otro punto importante de la librería **matlib** es el de representar de manera gráfica las ecuaciones de un sistema lineal. Con las funciones **plotEqn()** para dibujar un sistema lineal de 2 incógnitas, y **plotEqn3d()** para dibujar un sistema lineal de 3 incógnitas.

Ejemplo 2

En caso de tener un sistema de dos ecuaciones con dos incógnitas

```
showEqn(A, b)
```

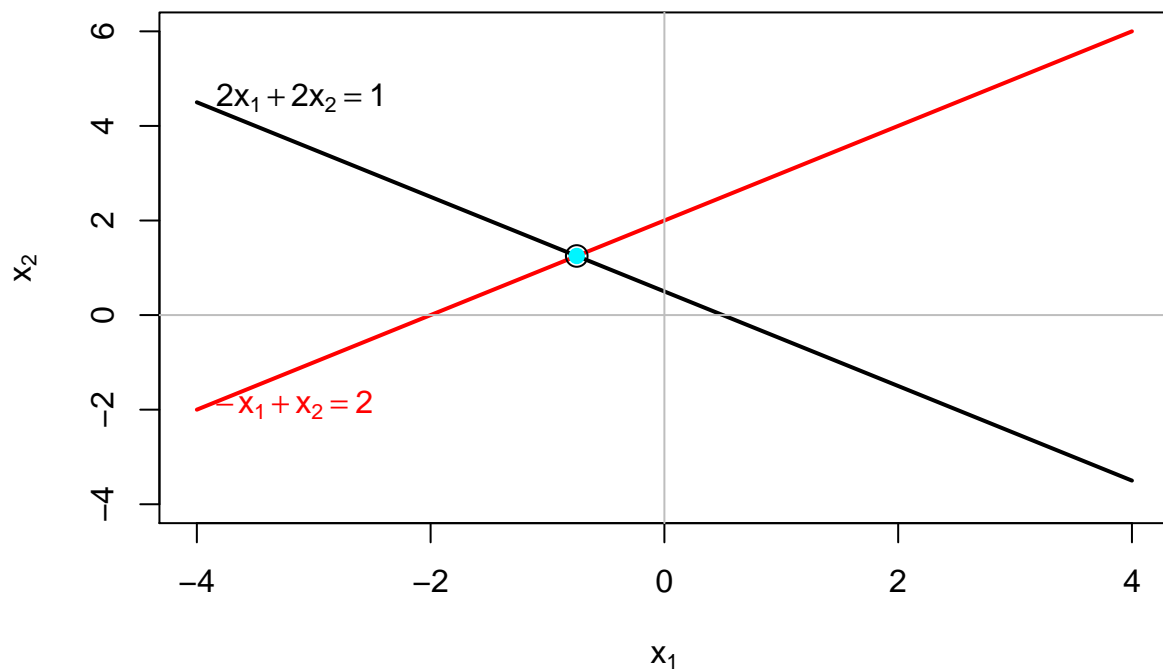
```
## 2*x1 + 2*x2 = 1
## -1*x1 + 1*x2 = 2
```

Su representación sería

```
plotEqn(A, b)
```

```
## 2*x[1] + 2*x[2] = 1
## -x[1] + x[2] = 2
```

```
points(-3/4, 5/4, col = "turquoise1", pch = 19)
```



La solución que anteriormente habíamos encontrado

$$\begin{cases} x_1 &= -\frac{3}{4} \\ x_2 &= \frac{5}{4} \end{cases}$$

es el punto donde ambas rectas intersectan