

# Ejercicios sobre LaTeX, R y Markdown

Enrique Ortega

4/4/2019

## Instrucciones

...omitido... Este ejercicio está parcialmente resuelto repositorio de github por lo voy a hacer mi propio documento, contraviniendo las instrucciones del propio ejercicio.

## Preguntas

### Pregunta 1

Realizar los siguientes productos de matrices en `R` :

$$A \cdot B$$

$$B \cdot A$$

$$(A \cdot B)^t$$

$$B^t \cdot A$$

$$(A \cdot B)^{-1}$$

$$A^{-1} \cdot B^t$$

donde

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 0 & 3 & 2 & 1 \\ 3 & 0 & 4 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 0 & 3 & 0 & 4 \\ 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 2 \end{pmatrix}$$

Finalmente, escribir haciendo uso de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  el resultado de los dos primeros productos de forma adecuada.

### Respuesta 1

Empezamos por instanciar las variables. Utilizaremos la función `matrix` para declarar la matrix a partir de un vector plano e indicando el número de filas más relleno por filas.

```
A = matrix(c(1, 2, 3, 4, 4, 3, 2, 1, 0, 1, 0, 2, 3, 0, 4, 0), nrow = 4, byrow = TRUE)
B = matrix(c(4, 3, 2, 1, 0, 3, 0, 4, 1, 2, 3, 4, 0, 1, 0, 2), nrow = 4, byrow = TRUE)
```

Las matrices resultantes son:

A

	[,1]	[,2]	[,3]	[,4]
[1,]	1	2	3	4
[2,]	4	3	2	1
[3,]	0	1	0	2
[4,]	3	0	4	0

B

	[,1]	[,2]	[,3]	[,4]
[1,]	4	3	2	1
[2,]	0	3	0	4
[3,]	1	2	3	4
[4,]	0	1	0	2

Seguimos con las operaciones solicitadas

## Producto $A \cdot B$

A %\*% B

	[,1]	[,2]	[,3]	[,4]
[1,]	7	19	11	29
[2,]	18	26	14	26
[3,]	0	5	0	8
[4,]	16	17	18	19

## Producto (¿comple conmutativa?) $B \cdot A$

B %\*% A

	[,1]	[,2]	[,3]	[,4]
[1,]	19	19	22	23
[2,]	24	9	22	3
[3,]	21	11	23	12
[4,]	10	3	10	1

Evidentemente, no la cumple.

## Transpuesta $(A \cdot B)^t$

t(A %\*% B)

	[,1]	[,2]	[,3]	[,4]
[1,]	7	18	0	16
[2,]	19	26	5	17
[3,]	11	14	0	18
[4,]	29	26	8	19

## Transpuesta y producto $B^t \cdot A$

```
t(B) %*% A
```

```
      [,1] [,2] [,3] [,4]  
[1,]    4    9   12   18  
[2,]   18   17   19   19  
[3,]    2    7    6   14  
[4,]   23   18   19   16
```

## Matriz invertida del producto $(A \cdot B)^{-1}$

```
solve(A %*% B)
```

```
      [,1] [,2] [,3] [,4]  
[1,] -1.66 -0.65  4.52  1.52  
[2,]  1.60  0.80 -4.60 -1.60  
[3,]  1.02  0.35 -2.84 -0.84  
[4,] -1.00 -0.50  3.00  1.00
```

## ...e inversa de una por la transpuesta de la otra $A^{-1} \cdot B^t$

```
solve(A) %*% t(B)
```

```
      [,1] [,2] [,3] [,4]  
[1,]  6.000000e-01  2.4  6.4  1.2  
[2,] -3.330669e-16 -2.0 -7.0 -1.2  
[3,] -2.000000e-01 -0.8 -3.8 -0.4  
[4,]  1.000000e+00  1.0  5.0  0.6
```

## Finalmente el resultado de los dos primeros productos en $\text{\LaTeX}$

Ya he practicado la generación manual de matrices en  $\text{\LaTeX}$  representando las matrices  $A$  y  $B$ . Voy a probar a representar sus productos mediante `result=asis`. Esto es, que sea R quien construya el código  $\text{\LaTeX}$

He consultado varios ejemplos y finalmente me he decantado por esta solución que paso a describir.

El algoritmo utilizado segmenta el texto  $\text{\LaTeX}$  final en tres partes.

- **El inicio**, con la sintaxis de apertura de matrix con paréntesis
- **El cuerpo**, con la sintaxis iterativa para representar los elementos de la matriz y los saltos de línea
- **El final**, con la sintaxis de cierre de matrix con paréntesis

### El cuerpo

El cuerpo de la matriz se genera mediante el uso de estas instrucciones:

- **apply** para poder tratar los elementos de la matrix mediante una función. Haremos el tratamiento por filas.
- **paste**, para concatenar texto. En este caso anidamos dos paste consecutivos para:
  - Mediante el *paste exterior* concatenar los elementos de una fila con el salto de línea

- Mediante el *paste interior* agrupar los elementos de una fila utilizando como *separador* el &. Esta agrupación se realiza con el parámetro *collapse*. En otros lenguajes sería equivalente a hacer un *join* de los elementos de un vector donde especificamos el separador entre elementos.

**Importante** Hay indicar el parámetro *result='asis'* para que el resultado de `R` sea interpretado como parte del markdown. En mi caso también he tenido que forzar la salida por HTML ya que por PDF no llegaba a renderizarse.

```
porFilas=1
# Preparamos las variables de inicio y fin de matriz
# Calculamos las matrices mostrar
P1 = A%%B
P2 = B%%A

# Las dobles \\ son para escapar el carácter \.
inicio1 = "$$(A \\cdot B) = \\begin{pmatrix}"
inicio2 = "$$(B \\cdot A) = \\begin{pmatrix}"
fin = "\\end{pmatrix}$$"

# Creamos el cuerpo de la matriz. Dado que cada salto de línea son dos \ tenemos que
# escaparlos dos veces para lo que necesitaremos cuatro \.
cuerpo1 = apply(P1,porFilas, FUN=function(x){paste(paste(x, collapse = "&"),"\\\\\\")})
cuerpo2 = apply(P2,porFilas, FUN=function(x){paste(paste(x, collapse = "&"),"\\\\\\")})

# Extraemos el texto LaTeX
writeLines(c(inicio1,cuerpo1,fin))
```

$$(A \cdot B) = \begin{pmatrix} 7 & 19 & 11 & 29 \\ 18 & 26 & 14 & 26 \\ 0 & 5 & 0 & 8 \\ 16 & 17 & 18 & 19 \end{pmatrix}$$

```
writeLines(c(inicio2,cuerpo2,fin))
```

$$(B \cdot A) = \begin{pmatrix} 19 & 19 & 22 & 23 \\ 24 & 9 & 22 & 3 \\ 21 & 11 & 23 & 12 \\ 10 & 3 & 10 & 1 \end{pmatrix}$$

## Pregunta 2

Crear un vector llamado `dni` donde almacenar los números del DNI cualquiera. Ejemplo de un DNI

$$dni = (5, 4, 2, 0, 1, 5, 6, 7)$$

Definir el vector en `R`. Calcular con `R` el vector `dni` al cuadrado, la raíz cuadrada del vector `dni` y, por último, la suma de todas las cifras del vector `dni`.

Finalmente, escribir todos los estos vectores en  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

## Respuesta 2

No me parece adecuado utilizar el número de DNI en un entorno como este. Así pues utilizaré un vector de números al azar. Concretamente utilizaré el siguiente vector:

$$dni = (2, 5, 1, 2, 7, 8, 5, 3)$$

Entiendo que las operaciones que se piden es sobre cada uno de los elementos del vector. Esto es, cuadrado de cada uno de los elementos, raíz de cada uno y la suma de todos. Para ello:

- Inicializaremos el vector
- Realizaremos las operaciones y las almacenaremos en variables para su posterior uso
- Generaremos el texto *L<sup>A</sup>T<sub>E</sub>X*
- Mostraremos el resultado

En esta ocasión como sólo tenemos una fila no nos hará falta utilizar apply. La conversión la realizaremos de esta manera: - Mediante la función sprintf preparamos la cadena con un punto de sustitución de tipo string. - Mediante paste y collapse agrupamos los elementos de vector en una cadena separada por comas.

```
dni=c(2,5,1,2,7,8,5,3)

# Resultado de las operaciones
cuadrado = dni^2
raiz = round(sqrt(dni),2)
suma = sum(dni)

cuadrado_md = sprintf("$$cuadrado=(%s)$$",paste(cuadrado,collapse=","))
raiz_md = sprintf("$$raiz=(%s)$$",paste(raiz,collapse=","))
suma_md = sprintf("$$suma=%s$$",suma)

writeLines(c(cuadrado_md, raiz_md, suma_md))
```

$$cuadrado = (4, 25, 1, 4, 49, 64, 25, 9)$$

$$raiz = (1.41, 2.24, 1, 1.41, 2.65, 2.83, 2.24, 1.73)$$

$$suma = 33$$

## Pregunta 3

Definir el vector *name* de letras con nombre y apellido en R y:

- Calcular el subvector que sólo contenga el nombre
- Calcular también el subvector que contenga sólo el apellido
- Ordenar el vector alfabéticamente
- Crear una matriz a partir del vector

## Respuesta 3

Vamos a ello.

1. Definimos el vector
2. Calculamos los subvectores para nombre y apellidos
3. Calculamos el vector ordenado
4. Creamos la matriz
5. Mostramos los resultados

Notas: Para separar nombre de apellidos localizaremos la posición del espacio y extraeremos el lado izquierdo para el nombre y el lado derecho para el apellido. El resto es combinación de lo ya visto para convertir vectores y/o matrices a formato *LaTeX*

```
full_name = c("E","n","r","i","q","u","e"," ","O","r","t","e","g","a")
first_name = full_name[1:(which(full_name==" ")-1)]
last_name = full_name[(which(full_name==" ")+1):length(full_name)]
sorted = sort(full_name)
mtx = matrix(full_name,nrow = 2, byrow = TRUE)

full_name_md = sprintf("$$fullname=(%s)$$",paste(full_name,collapse = ","))
first_name_md = sprintf("$$firstname=(%s)$$",paste(first_name,collapse = ","))
last_name_md = sprintf("$$lastname=(%s)$$",paste(last_name,collapse = ","))
sorted_md = sprintf("$$sorted=(%s)$$",paste(sorted,collapse = ","))
mtx_inicio = "$$matrix = \\begin{pmatrix}"
mtx_fin = "\\end{pmatrix}$$"
porFilas=1

mtx_cuerpo = apply(mtx,porFilas, FUN=function(x){paste(paste(x, collapse = "&"),
"\\\\")})
mtx_md = sprintf("%s%s%s",mtx_inicio,mtx_cuerpo,mtx_fin)

writeLines(c(full_name_md,first_name_md,last_name_md,sorted_md,c(mtx_inicio,mtx_cuerp
o,mtx_fin)))
```

$fullname = (E, n, r, i, q, u, e, , O, r, t, e, g, a)$

$firstname = (E, n, r, i, q, u, e)$

$lastname = (O, r, t, e, g, a)$

$sorted = (, a, e, e, E, g, i, n, O, q, r, r, t, u)$

$matrix = \begin{pmatrix} E & n & r & i & q & u & e \\ & O & r & t & e & g & a \end{pmatrix}$